Multilingual Deep Learning

Sarath Chandar A P
Department of Computer Science
Indian Institute of Technology Madras, India
apsarathchandar@gmail.com

Mitesh M. Khapra
IBM Research India
mikhapra@in.ibm.com

Balaraman Ravindran
Department of Computer Science
Indian Institute of Technology Madras, India
ravi@cse.iitm.ac.in

Vikas Raykar
IBM Research India
viraykar@in.ibm.com

Amrita Saha
IBM Research India
amrsaha4@in.ibm.com

Abstract

Resource fortunate languages such as English, French, Chinese, etc. clearly overshadow many *not-so-fortunate* languages in terms of both (i) the number of NLP tools available and (ii) the quality of these tools. One solution to alleviate this problem is to collect more annotated resources for these languages, but this is often not feasible due to the cost, time, and effort involved. A more feasible option is to use cross language learning which aims to use annotated resources available in some resource fortunate language to bring NLP capability to a less fortunate language. In this work, we propose such a cross language learning framework which has its origins in deep learning. The idea is to learn a shared deep representation for two languages (say, $L1$ and $L2$) to represent data from the two languages in a common space. Once such a shared representation is learned training data available in $L1$ can be projected to this space and the resulting representation can be used for training a model. Similarly, test data from $L2$ can be projected to this space and the resulting representation can be fed to the trained model for inference. We evaluate the proposed framework on two tasks, *viz.*, cross language sentiment analysis and cross language transliteration equivalence. The experimental results show that the performance of the proposed framework is comparable to state-of-the-art approaches for these tasks.

1 Introduction

Languages show different levels of maturity with respect to their Natural Language Processing (NLP) capabilities. This maturity in terms of the quality and number of NLP tools available for a given language is directly proportional to the amount of annotated resources available for that language. As a result, languages such as English which have plenty of annotated resources at their disposal are better equipped than other languages which are not so fortunate in terms of annotated resources. For example, high quality pos taggers, parsers, sentiment analyzers are already available for English but this is not the case for many other languages such as Hindi, Marathi, Bodo, Farsi, Urdu, *etc*. This situation was acceptable in the past when only a few languages dominated the digital content available online and elsewhere. However, the ever increasing number of languages on the web today has made it important to accurately process natural language data in such less fortunate languages also. An obvious solution to this problem is to improve the annotated inventory of these languages but the involved cost, time, and effort act as a natural deterrent to this.

To overcome this problem of resource scarcity, recently there as been a lot of interest in reusing resources from a resource fortunate language to develop NLP capabilities in a resource deprived language [25, 6, 14, 23, 17, 15, 13, 10]. One way of achieving this is to project parameters learned from the annotated data of one language to another language. These projections are enabled by a bilingual resource such as a Machine Translation tool, a parallel corpus[1] or a bilingual dictionary. Alternatively, one can exploit such bilingual resources to learn a shared representation for two languages. For example, [21] uses Canonical Correlation Analysis [9] to learn a common representation for names in two languages using a bilingual parallel list of names.

To further illustrate the above idea of reusing resources, we consider the task of Cross Language Sentiment Analysis [4, 24] which uses training data available in one language to develop a sentiment classifier for another language. The idea is to first build a model M for predicting sentiment polarity using the training data available in language L_1. Such a model can obviously not be used for predicting the sentiment polarity of a document belonging to another language (say L_2) due to the difference in the vocabulary (and hence the representation of documents) in the two languages. To circumvent this problem, the test document in L_2 is translated to L_1 using a machine translation system and then the model M is applied to this translated document. The Machine Translation system here enables the two documents to be represented in the same space (*i.e.*, the space comprising of the vocabulary of L_1). In this work, we propose a framework for representing entities (words, sentences, documents, etc.) from two languages in a common space using concepts from deep learning. This can be looked upon as an alternative to using Machine Translation for enabling a shared representation for two languages.

Our work uses deep learning (specifically, auto-encoders) to learn a shared representation for two languages. The model learns from a list of parallel entities in the two languages. These entities can either be words, sentences or documents depending on the task at hand. The objective function is designed to minimize the distance between the projections of parallel entities in this common space. For example, if the model is trained using a list of parallel sentences in the two languages, then the objective is to ensures that if sentence $S_1^{L_1}$ is a translation of $S_1^{L_2}$ then the distance between their projections in the common space is minimum. Once such a common representation is learned, entities from the two languages can be projected to this common space and model training and inference can then happen in this common space. A salient feature of the proposed model when compared to CCA is that in addition to learning a shared representation, it also has the ability to predict the representation of an entity in the target language given the representation of its parallel entity in the source language (as explained later in section 3).

We evaluate the proposed approach on the task of Cross Language Sentiment Analysis and show that its performance is comparable to state-of-the-art approaches. Further, to evaluate whether the proposed model indeed ensures that the projections of parallel entities have a high similarity we use it for the task of transliteration mining which aims at finding parallel transliteration pairs across two languages. Note that there is no cross language learning in this task but the aim is to just show that a pair of words in L_1 and L_2 which are transliterations of each other get projected close to each other. The main contributions of our work can be summarized as follows:

- We propose a novel variant of auto-encoder called predictive auto-encoder, that learns the shared representation for two different languages.
- The model can also predict the features in one language, given the features in another language. In some sense, the model does contextual translation without using any machine translation tool.
- The proposed framework is language-independent and task-independent. It can be applied to any task such as cross language sentiment analysis, cross language document classification, and so on.
- Apart from these contributions, we are also introducing a new benchmark dataset for cross language sentiment analysis between English and French.

[1] For example, a set of English documents with their corresponding French translations form a English-French parallel corpus

The remainder of this paper is organized as follows. In Section 2 we briefly discuss some related work . In section 3 we introduce the objective function used for our predictive encoder and compare it with the objective function of an auto-encoder. In section 4, we describe our overall approach for training and testing. In sections 5 and 6, we discuss the empirical performance of our approach on the task of Cross Language Sentiment Analysis and Transliteration Mining respectively. Finally, in section 7, we present concluding remarks and suggest possible future work.

2 Related Work

We borrow ideas from the vast literature on auto-encoders and hence its necessary to briefly discuss auto-encoders and their variants. An auto-encoder [20] is a three layer neural network containing an input layer, a hidden layer and an output layer which reconstructs the input. Typically, the aim is to learn a compact representation in the hidden layer such that the reconstruction error is minimum. This idea is further extended in [2] to design a deep neural network by stacking multiple auto-encoders and training them greedily. Similar greedy layer-wise training to design a deep neural network using Restricted Boltzman Machines was proposed in [8]. Both the methods consist of an unsupervised pre-training phase followed by supervised fine-tuning phase.

Several variants to the basic auto-encoder have been proposed. Denoising auto-encoder [22] is one such variant where the input is corrupted before feeding it to the auto-encoder and the goal is to reconstruct the clean input. This acts like a regularization for the auto-encoder. Following this, multiple regularization techniques for auto-encoders were proposed. [7] proposed saturating auto-encoder that explicitly limits the auto-encoder's ability to reconstruct the inputs which are not near the data manifold. [18] propose a different regularization criteria which favours mappings that are more strongly contracting at the training samples. [19] proposes a novel variant of auto-encoder called Discriminative Recurrent Sparse Auto-encoders, which allows sharing parameters between successive layers of a deep network.

Most of the work mentioned above focuses on single view input whereas we are interested in multi-view input where two different views of the data are available (for example a sentence and its translation in another language). In this context, the work of [16] is very closely related to our work and is in fact the inspiration for our work. They propose a methodology to train a network where two views of the data (audio and video in their case) are available. The goal is two fold: (i) to learn a common representation for audio and video data and (ii) to reproduce audio (or video) data given the corresponding video (or audio) data. Although our ideas are very similar, we use a different objective function than the one used in their work (as explained in section 3). Given that our aim is to learn a shared representation for two views of the data, it is also very closely related to Canonical Correlation Analysis (CCA) [9]. However, one difference is that CCA learns a common representation only, whereas, our model (in addition) can also predict the target view given the source view.

3 Predictive Auto-encoder

At the heart of our cross language learning framework, lies a novel Predictive Auto-encoder (PAE) which learns a shared representation for entities in two languages. As input it takes a parallel list of entities in L_1 and L_2. For the purpose of illustration, we will consider a list of parallel documents in the two languages. A document i in language L_1 can be represented by a feature vector $p_i \in \mathbb{R}^{d_1}$. In the simplest case each feature could be a binary feature indicating the absence or presence of a word in the document (in this case d_1 would simply be the size of the vocabulary of L_1). Similarly the corresponding parallel document in L_2 can be represented as a feature vector $q_i \in \mathbb{R}^{d_2}$. Now consider that we are given a sample $Z = \{(p_i, q_i)\}_{i=1}^{n}$ containing n such parallel documents. For a given pair (p_i, q_i) we construct two vectors, $z_i^1, z_i^2 \in \mathbb{R}^{d_1+d_2}$ such that $z_i^1 = (p_i \in \mathbb{R}^{d_1}, 0 \in R^{d_2})$ and $z_i^2 = (0 \in \mathbb{R}^{d_1}, q_i \in R^{d_2})$. z_i^1 is thus an embedding of p_i in a $d_1 + d_2$ dimensional space such that the values of the last d_2 dimensions are set to 0. Similarly, z_i^2 is an embedding of q_i in a $d_1 + d_2$ dimensional space such that the values of the first d_1 dimensions are set to 0. The aim is to learn a mapping function $f : \mathbb{R}^{d_1+d_2} \to \mathbb{R}^d$ such that $f(z_i^1) \in R^d$ is highly correlated with $f(z_i^2) \in R^d$. In other words the aim is to maximize the correlation between $f((p_i, \mathbf{0}))$ and $f((\mathbf{0}, q_i))$ which ensures that embeddings of p_i and q_i in this d dimensional space are close to each other.

To achieve this goal, we propose a variant of auto-encoders called Predictive Auto-encoder (PAE). Akin to an auto-encoder, our PAE also consists of an encoder followed by a decoder. The encoder is a function f that maps an input $z \in \mathbb{R}^{d_1+d_2}$ to a hidden representation $f(z) \in \mathbb{R}^d$. It can be defined as

$$h = f(z) = s_f(W \cdot z + b_h) \tag{1}$$

where s_f is a nonlinear activation function such as sigmoid function.

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

The parameters of the encoder are a weight matrix $W \in \mathbb{R}^{d \text{ x } (d_1+d_2)}$ and a bias vector $b_h \in \mathbb{R}^d$.

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

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

where s_g is the decoder's activation function (typically the identity function or a sigmoid function). The decoder's parameters are the matrix $W' \in \mathbb{R}^{(d_1+d_2) \text{ x } d}$ and a bias vector $b_y \; in \; \mathbb{R}^{d_x}$. In general, W' = W^T.

We now describe the process of training a PAE given a sample $Z = \{(p_i, q_i)\}_{i=1}^n$. From each input pair we construct three vectors: $z_i = (p_i, q_i)$ and z_i^1 and z_i^2 as defined earlier. z_i is simply a concatenation of the two views of the data and acts as a composite view of the data. Given n such triplets (z_i, z_i^1, z_i^2), the PAE is trained to learn parameters θ = $\{W, b_h, b_y\}$ which minimize the following objective function:

$$\begin{aligned} \mathcal{J}_{PAE}(\theta) = \sum_{i=1}^{n} L(z_i, g(f(z_i^1)) + \sum_{i=1}^{n} L(z_i, g(f(z_i^2)) \\ + \sum_{i=1}^{n} L(z_i, g(f(z_i)) - \alpha \sum_{i=1}^{n} cor(f(z_i^1), f(z_i^2)) \end{aligned} \tag{4}$$

where L is the reconstruction error and α is the scaling parameter used to scale the correlation term to the range of squared error terms. Lets understand the motivation behind each term in the objective function. The first term ensures that the the error of reconstructing the composite view $z_i = (p_i, q_i)$ given only one view $z_i^1 = (p_i, 0)$ is minimum. In other words, this ensures that the model has a predictive power and can predict the second view (q_i) given only the first view(p_i). The motivation behind the second term is similar except that the roles of p_i and q_i are reversed. The third term is the conventional auto-encoder error and helps to learn a compact representation of the composite view. Finally, the fourth term (with the negative sign) ensures that the hidden representations of the two parallel views are highly correlated. We contrast our approach with the approach described in [16] which uses the following objective function to train an auto-encoder using multiview data :

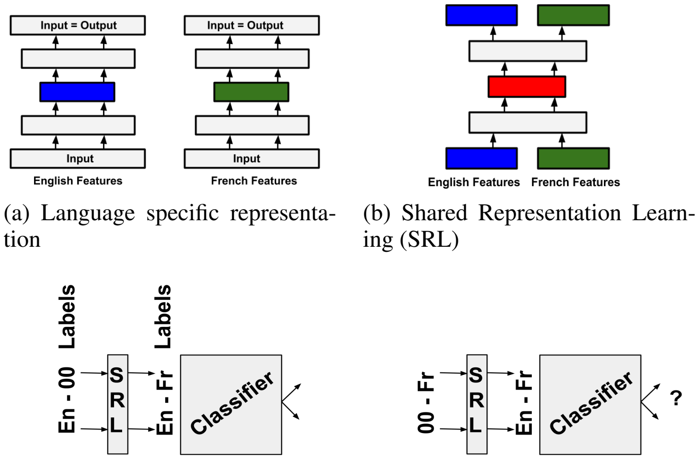
$$\mathcal{J}_{AE}(\theta) = \sum_{i=1}^{n} L(z_i, g(f(z_i)) \tag{5}$$

where L is the reconstruction error. Similar to our approach, they also construct the three vectors (z_i, z_i^1, z_i^2) from each input pair (p_i, q_i). However, unlike our approach these 3 vectors are fed independently to the network during training. In contrast, in our approach these vectors are not treated independently as the objective function tries to maximize the correlation between $f(z_i^1)$ and $f(z_i^2)$. Note that (5) lacks an explicit term that links the three inputs together. In our approach the parameter updates happen only after all the 3 versions of the input are passed through the network. While this introduces a tighter coupling between the 3 versions of the input, adding the explicit correlation term directly targets the hidden representation learnt. This also leads to better empirical performance as described in Section 6. Even though the work in [16] could conceptually be used for prediction of one language given the other, the authors report that the performance of the system was poor for audio-video shared representation learning. Adding the correlation term to the error and the joint training enables our system to achieve very good performance.

4 Overall process for Cross Language Learning

In this section, we describe the overall process for Cross Language Learning (CLL) framework which uses the predictive auto-encoder described above. The proposed approach has four phases as described below.

1. Language specific representation: In this phase, we are interested in obtaining a language specific representation (p_i or q_i) for an entity in L_1 or L_2 respectively. As mentioned earlier, if the entity is a document this representation can be as simple as a set of binary features indicating the presence or absence of a n-gram in the vocabulary of the language. Alternatively, if the entity is a word then each feature could indicate the presence or absence of any n-gram character in the language or some such suitable representation. Not surprisingly, if we use the raw representation in phase 2 directly, we obtain very poor performance. This was confirmed in preliminary experiments and hence we adopted this additional phase. We train a k-layered stacked auto-encoder to learn an abstract representation (Figure1(a)) for a given entity using its raw representation (n-gram words, n-gram characters, co-occurrence vectors, etc.).



(a) Language specific representation

(b) Shared Representation Learning (SRL)

(c) Source Langauge Training

(d) Target Langauge Testing

Figure 1: Proposed framework

2. Shared Representation Learning(SRL): For the next phase we need a pair of parallel entities in L_1 and L_2 wherein the representation p_i (or q_i) of an entity in L_1 (or L_2) is obtained using unsupervised feature learning as described above. A sample $Z = \{(p_i, q_i)\}_{i=1}^{n}$ of such parallel entities is then passed to the PAE to learn a shared representation. This process is illustrated in Figure1(b).

3. Source Language Training: Now we come to the crux of cross language training where the aim is to train a model using the data available in L_1 and apply this model to data from L_2. Lets assume we have a sample $D = \{x_i, y_i\}_{i=1}^{k}$ of training data available in L_1 where x_i is the input and y_i is the label. For each x_i we first learn the abstract representation p_i in L_1 using the auto-encoder in phase 1. Next for each p_i we obtain the compact representation $f((p_i, \mathbf{0})) = f(z_i^1)$ using the PAE trained in phase 2. Effectively, we have projected the original input x_i to a space in which entities from L_2 can also be represented. Thus, a model trained using this projected data $\{f(z_i^1)\}_{i=1}^{n}$ can be applied to entities belonging to L_2 after projecting them to this space. This process of training is illustrated in Figure1(c).

4. Target Language Testing: Finally, the model trained above is applied to test data from L_2 by first projecting it to the common space as illustrated in Figure1(d). For each x_i, we first learn the abstract representation q_i in L_2 using the auto-encoder in phase 1. Next for each q_i we obtain the compact representation $f((\mathbf{0}, q_i)) = f(z_i^2)$ using the PAE trained in phase 2. Now use the classifier to classify the test instance.

5 Empirical Evaluation: Cross Language Sentiment Analysis

We evaluate the performance of the proposed framework on the task of Cross Language Sentiment Analysis where the goal is to detect the sentiment polarity (positive or negative) of a document in language L_2 using training data available in language L_1. For the purpose of this evaluation, we created a Multilingual Dataset for Sentiment Analysis similar to the Multi-domain dataset used in [3]. Specifically, we collected reviews for English and French DVDs from amazon [2]. These reviews are accompanied with a reviewer rating on a scale of 1 to 5 (5 indicating excellent and 1 indicating poor). We considered reviews with ratings 4 and 5 to be positive and reviews with ratings 1 and 2 to be negative. The details of the dataset are provided in Table 1.

Language	Training instances			Test instances		
	Positive	Negative	Neutral	Positive	Negative	Neutral
English	20000	20000	10000	2000	2000	-
French	20000	20000	10000	2000	2000	-

Table 1: Multilingual Sentiment Dataset Description

The dataset will be made publicly available and will hopefully help in furthering the research on multilingual sentiment analysis.

5.1 Experimental Setup

As mentioned in section 4, our approach has four phases (i) language specific representation (ii) shared representation learning (iii) task specific supervised training and (iv) cross language testing. We describe the procedure followed for executing each of these phases.

For monolingual deep learning, we used 50,000 training documents for each language from the Multilingual Dataset. Note that we do not use parallel corpora in this phase. We chose an arbitrary set of documents in each language drawn from the same domain as the test documents. This is to ensure that there is a strong overlap in the vocabulary of the corpus used in phases 1 and 2. We used a five layered stacked auto-encoder for this phase. To feed the first layer of the auto-encoder we converted the documents to a feature vector comprising of the top 40,000 unigrams in the vocabulary. In subsequent layers of the stacked auto-encoder, we reduced the number of hidden neurons from 10,000 to 5000 to 2500 to 500. The representation from the last layer consisting of 500 hidden neurons is used as the deep representation of a given document.

Next, for the second phase, we need English-French parallel documents. For this, we translated the 50,000 English documents used above to French using a state of the art Machine Translation Tool [1] thus creating a English French parallel corpus. We then obtain the language specific deep representations for each of these documents and then use these parallel deep representations to train the predictive auto-encoder. In the third phase, we need to train a sentiment classifier using 40K English training data. Instead of using a simple unigram based feature representation for the documents we first obtain the language specific deep representation of these documents and then project this representation into the common space using the predictive auto-encoder. We then train a classifier using this shared representation of the English documents as the feature vector. Finally, we take the test documents from French, repeat the above process of projecting them to the common space and then feed them to the trained model for inference.

We compared the above approach with some standard approaches for CLSA. The empirical upper bound for the performance is obtained by using a classifier trained on French training data. In addition, we consider two baselines: (i) a classifier trained using English data and tested on French data after translating it to English using a MT system and (ii) a classifier trained after translating the training instances into French and then tested on French instances. For the two baseline approaches we use a ungiram feature representation. For all the methods we use SVM [5] as the classifier. The accuracy of the different approaches are reported in Table 2.

[2] www.amazon.com and www.amazon.fr

S.No	Approach	Train data	Test data	Accuracy
1	Self Training	Fr	Fr	86.1%
2	Translate and Train	En - translated to Fr	Fr	63.4%
3	Translate and Test	En	Fr - translated to En	65.15%
4	Common Representation Learning	En	Fr	72%

Table 2: Accuracy of different approaches for Cross Language Sentiment Analysis

6 Empirical Evaluation: Transliteration Equivalence

In the previous section, we showed the application of our model in a cross language learning setup. In addition to cross language learning, our model can also be used for the task of determining bilingual equivalence. As a case study, we consider the task of determining transliteration equivalence of named entities wherein given a word u from language L_1 and a word v from language L_2 the goal is to determine whether u and v are transliterations of each other. Several approaches have been proposed for this task and the one most related to our work is an approach which uses CCA for determining transliteration equivalence. We compare our results with this approach. Through this case study, we aim to answer the following questions:

1. Given source language view can the target language view be reconstructed?
2. Do equivalent entities have similar common representations ?
3. What is the effect of number of features in the shared representation ?

6.1 Experimental Setup

Once again we describe the procedure used for different phase of our approach. For obtaining language specific representation, we collected 50,000 words from English and Hindi Wikipedia titles. We used a k layered stacked auto-encoder for learning a language specific representation. To feed the bottom most layer in the auto-encoder we converted the words to a character-bigram based feature vector. There were 651 character-bigram features in English and 2860 character-bigram features in Hindi. In the final layer we retained 200 hidden neurons and used this as the language specific deep representation for the two languages. Next, for shared representation learning, we used 15,000 transliteration pairs from NEWS 2009 training set [12] to train the network. As before, we first obtain a language specific representation for each word in the pair and then use these parallel deep representations for training a predictive auto-encoder. Testing was done on the standard NEWS10 transliteration equivalence testset [11]. We report the findings of our experiments and in the process answer the questions raised above.

6.2 Reconstructing target language view

A useful functionality of the predictive auto-encoder is that given only source language features it can predict target language features. In this subsection, we provide a quantitative evaluation of how good these predictions are and also highlight the contribution of each term in the objective function. For this we trained the auto-encoder using different combinations of the terms in the objective function and noted the average squared error in the reconstruction of the target features, given the source features (see Table 4). For the remainder of this section, we use the naming convention given in Table 3 for referring to different terms in the objective functions.

Name	Term in Objective Function
$f1$	$L(z_i, g(f(z_i^1))$
$f2$	$L(z_i, g(f(z_i^2))$
$f3$	$L(z_i, g(f(z_i))$
$f4$	$cor(f(z_i^1), f(z_i^2))$

Table 3: Naming Convention for terms in Objective Function

S.No	Objective Function	Average Reconstruction Error
1	$f1 + f2$	3.439
2	$f1 + f2 + f3$	3.457
3	$f1 + f2 + f3 + f4$	3.396

Table 4: Reconstruction error for various error functions

6.3 Identifying equivalent entities

Next, we are interested in determining whether equivalent entities (transliteration pairs in this case) have very similar representations in the common space. For this, we use NEWS10 English-Hindi transliteration equivalence test set which contains 5468 word pairs out of which 982 are transliteration pairs and the remaining are not. For every word pair (u, v) we obtain a representations for u and v by passing them through the models trained in phase 1 and 2. We then calculate the cosine similarity between these representations of u and v. If the cosine similarity is above a threshold we mark the word pair as equivalent. We compare our approach with CCA which also learns a shared representation using the same training data that we used. The results of this experiment are reported in Table 5. In this task, CCA performs better than PAE. We suspect that CCA performs better since (a) we also try to ensure that the reconstruction error is minimized and (b) more importantly CCA minimizes correlation between the abstract features. But further study is needed to verify these hypothesis.

S.No	Model	Precision	Recall	F1-Measure
1	CCA	0.863	0.917	0.889
2	$f1 + f2$	0.702	0.848	0.768
3	$f1 + f2 + f3$	0.649	0.831	0.729
4	PAE (Our Model)	0.79	0.844	0.815

Table 5: Performance on NEWS10 En-Hi Transliteration Mining Dataset

6.4 Effect of number of features in the common representation

To answer this question we vary the number of neurons in the hidden layer while learning the shared representation. The results of this experiment are reported in Table 6. It was observed that the performance decreases as we increase the number of features. Also there is a decrease in performance when there are very few features. This requires further analysis of the functioning of PAE.

No. of nodes	Precision	Recall	F1-measure
10	0.664	0.873	0.754
20	0.79	0.844	0.815
30	0.739	0.861	0.795
40	0.801	0.779	0.790
50	0.708	0.869	0.780

Table 6: Effect of number of features

7 Conclusion

We proposed a Predictive auto-encoder for learning a shared representation for cross-language tasks. In addition to learning a shared representation, the proposed model is also capable of predicting one view given the other. We evaluated the approach on two NLP tasks *viz.*, Cross Language Sentiment Analysis and Transliteration Equivalence. While the initial results are encouraging further investigation is needed to beat state-of-the-art approaches. As future work, it would be interesting to see if the above model can be extended to more than two languages. We would also like to apply our model to other tasks such as cross language information retrieval, cross language subjectivity analysis, etc.

References

[1] Y. Al-Onaizan and K. Papineni. Distortion models for statistical machine translation. In *Proceedings of ACL*, ACL-44, pages 529–536, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Proceedings of NIPS*, pages 271–278, 2006.

[3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, 2007.

[4] J. Brooke, M. Tofiloski, and M. Taboada. Cross-linguistic sentiment analysis: From english to spanish. In *International Conference on Recent Advances in NLP*, 2009.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] D. Das and S. Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609, Portland, Oregon, USA, June 2011.

[7] R. Goroshin and Y. LeCun. Saturating auto-encoders. In *International Conference on Learning Representations*, 2013.

[8] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[9] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[10] M. M. Khapra, S. Shah, P. Kedia, and P. Bhattacharyya. Projecting parameters for multilingual word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 459–467, Singapore, August 2009.

[11] A. Kumaran, M. M. Khapra, and H. Li. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28, Uppsala, Sweden, July 2010.

[12] H. Li, A. Kumaran, M. Zhang, and V. Pervouvhine. Whitepaper of news 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 19–26, Suntec, Singapore, August 2009.

[13] P. Mannem and A. Dara. Partial parsing from bitext projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1597–1606, Portland, Oregon, USA, June 2011.

[14] R. Mihalcea, C. Banea, and J. Wiebe. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 976–983, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[15] S. Mukund, D. Ghosh, and R. K. Srihari. Using cross-lingual projections to generate semantic role labeled corpus for urdu: a resource poor language. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 797–805, Beijing, China, 2010.

[16] 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.

[17] S. Padó and M. Lapata. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research (JAIR)*, 36:307–340, 2009.

[18] S. Rifai, Y. Dauphin, P. Vincent, and Y. Bengio. A generative process for contractive auto-encoders. In *Proceedings of ICML*, 2012.

[19] J. T. Rolfe and Y. LeCun. Discriminative recurrent sparse auto-encoders. In *International Conference on Learning Representations*, 2013.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by backpropagating errors. *Nature*, 323:533–536, 1986.

[21] R. Udupa and M. M. Khapra. Transliteration equivalence using canonical correlation analysis. In *Proceedings of the 32nd European Conference on IR Research*, pages 75–86, 2010.

[22] P. Vincent, H. L. Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML08)*, pages 1096–1103, 2008.

[23] X. Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243, Suntec, Singapore, August 2009.

[24] X. Wan. Co-training for cross-lingual sentiment classification. In *ACL/AFNLP*, 2009.

[25] D. Yarowsky and G. Ngai. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8, Pittsburgh, Pennsylvania, 2001.