Instance-based Inductive Deep Transfer Learning by Cross-Dataset Querying with Locality Sensitive Hashing

Somnath Basu Roy Chowdhury
IIT Kharagpur
brcsomnath@gmail.com

K M Annervaz
Indian Institute of Science
annervaz@iisc.ac.in

Ambedkar Dukkipati
Indian Institute of Science
ambedkar@iisc.ac.in

Abstract

Supervised learning models are typically trained on a single dataset and the performance of these models rely heavily on the size of the dataset i.e., the amount of data available with ground truth. Learning algorithms try to generalize solely based on the data that it is presented with during the training. In this work, we propose an inductive transfer learning method that can augment learning models by infusing similar instances from different learning tasks in Natural Language Processing (NLP) domain. We propose to use instance representations from a source dataset, *without inheriting anything* else from the source learning model. Representations of the instances of *source* and *target* datasets are learned, retrieval of relevant source instances is performed using soft-attention mechanism and *locality sensitive hashing* and then augmented into the model during training on the target dataset. Therefore, while learning from a training data, we also simultaneously exploit and infuse relevant local *instance-level information* from an external data. Using this approach we have shown significant improvements over the baseline for three major news classification datasets. Experimental evaluations also show that the proposed approach reduces dependency on labeled data by a significant margin for comparable performance. With our proposed cross dataset learning procedure we show that one can achieve competitive/better performance than learning from a single dataset.

1 Introduction

A fundamental issue with performance of supervised learning techniques (like classification) is the requirement of enormous amount of labeled data, which in some scenarios maybe expensive or impossible to acquire. Every supervised task requires a dedicated labeled dataset and training state-of-the-art deep learning model requires extensive computational power. In this paper, we propose a deep transfer learning method that can enhance the performance of learning models by incorporating information from a secondary dataset belonging to a similar domain.

We present our approach in an *inductive transfer learning* (Pan and Yang, 2010) framework, with a labeled *source* ($\mathcal{D}_S$ domain and task $\mathcal{T}_S$) and *target* ($\mathcal{D}_T$ domain and task $\mathcal{T}_T$) dataset, the aim is to boost the performance of target predictive function $f_T(\cdot)$ using available knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, given $\mathcal{T}_S \neq \mathcal{T}_T$. Knowledge transfer in our approach takes place in four ways (a) instance-transfer (b) feature-representation-transfer (c) parameter-transfer and (d) relational-knowledge-transfer. Parameter and relational knowledge transfer are studied exhaustively in inductive transfer literature. Our work is based on a simple inductive bias (also used in (Snell et al., 2017)), that there exists an embedding space where instances belonging to the same class cluster around a central point. We utilize the instance-level information in the source dataset, and also make the newly learnt target instance representation similar to the retrieved source instances. This allows the learning algorithm to improve generalization across the source and target datasets. We use *instance-based learning* that actively looks for similar instances in the source dataset given a target instance. The intuition behind retrieving similar instances comes from instance-based learning perspective, where simplification of the class distribution takes place within the locality of a test instance. As a result, modeling of similar instances become easier (Aggarwal, 2014). Similar instances have the maximum amount of information necessary to classify an unseen instance, as exploited by techniques like k-nearest neighbours.

We derived inspiration to propose this method

Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo), pages 183–191
Hong Kong, China, November 3, 2019.
https://doi.org/10.18653/v1/P17

from the working of the human brain, where *memory consolidation* (McGaugh, 2000) occurs, in which new memory representations are consolidated slowly over time for efficient retrieval in future. According to (McGaugh, 2000), newly learnt memory representation remain in a fragile state and are affected as further learning takes place. In our approach, we make use of encodings of instances precipitated while training for the source task using an independent model. This model being independently used for an source task and can be adapted as required, is in alignment with memory consolidation in human brain.

One of the attractive features of the proposed method is that the search mechanism allows us to use more than one source dataset during training the joint model to achieve inductive transfer learning. Our approach differs from the standard instance-based learning in two major aspects. First, the instances retrieved are not necessarily from the same dataset, but can be from various secondary datasets. Secondly, our model simultaneously makes use of local instance level information as well as the macro-statistical view point of the dataset, where typical instance-based learning like k-nearest neighbour search make use of only the local instance level information.

2 Background

Locality Sensitive Hashing (LSH): Locality Sensitive Hashing (Gao et al., 2014; Gionis et al., 1999) is an algorithm which performs approximate nearest neighbor similarity search for high-dimensional data in sub-linear time. LSH is a data independent hashing technique as the hash functions are selected at random, which makes LSH perfectly suited for our purpose. Latent vectors encountered during training cannot be accessed, which is required for constructing data-driven hash functions.

The locality sensitive hash family, $\mathcal{H}$ has to satisfy certain constraints mentioned in (Indyk and Motwani, 1998) for nearest neighbor retrieval. The LSH Index maps each point p into a bucket in a hash table with a label $g(p) = (h_1(p), h_2(p), \ldots, h_k(p))$, where $h_1, h_2, \ldots, h_k$ are chosen independently with replacement from $\mathcal{H}$. We generate l different hash functions of length k given by $G_j(p) = (h_{1j}(p), h_{2j}(p), \cdots, h_{kj}(p))$ where $j \in 1, 2, \ldots, l$ denotes the index of the hash table. Given a collection of data points $\mathcal{C}$, we hash them into l hash tables by concatenating randomly sampled k hash functions from $\mathcal{H}$ for each hash table. While returning the nearest neighbors of a query Q, it is mapped into a bucket in each of the l hash tables. The union of all points in the buckets $G_j(Q), j = 1, 2, \ldots, l$ is returned. Therefore, all points in the collection $\mathcal{C}$ is not scanned and the query is executed in sub-linear time. The storage overhead for LSH is sub-quadratic in n, the number of points in the collection $\mathcal{C}$.

LSH Forests (Bawa et al., 2005) are an improvement over LSH Index which relaxes the constraints on hash family $\mathcal{H}$ with better practical performance guarantees. LSH Forests utilizes l prefix trees (LSH trees) instead of having hash tables, each constructed from independently drawn hash functions from $\mathcal{H}$. The hash function of each prefix tree is of variable length (k) with an upper bound k_m. The length of the hash label of a point is increased whenever a collision occurs to form leaf nodes from the parent node in the LSH tree. For m nearest neighbour query of a point p, the l prefix trees are traversed in a top-down manner to find the leaf node with highest similarity with point p. From the leaf node, we traverse in a bottom-up fashion to collect M points from the forest, where $M = cl$, c being a small constant. It has been shown in (Bawa et al., 2005), that for practical cases the LSH Forests execute each query in constant time with storage cost linear in n, the number of points in the collection $\mathcal{C}$.

Instance-based transfer learning: Instance-based transfer learning has been extensively studied in literature (Zadrozny, 2004) (Gretton et al., 2009) (Huang et al., 2007) (Sugiyama et al., 2008) (Dai et al., 2007). These methods primarily focus on the problem of distribution mismatch between data from two different sources. They also assume that the training instances are sampled from a homogenous distribution and have the same target label space. In our approach, we are not assuming any constraints on the distribution of data or label space, our only assumption is that the datasets should have certain feature overlap in some embedding space. The feature overlap may not necessarily be substantial, as we also enforce the instance representations to be similar using a penalty function. The penalty function performs structural transformation of the feature space, which is usually an attribute of feature-

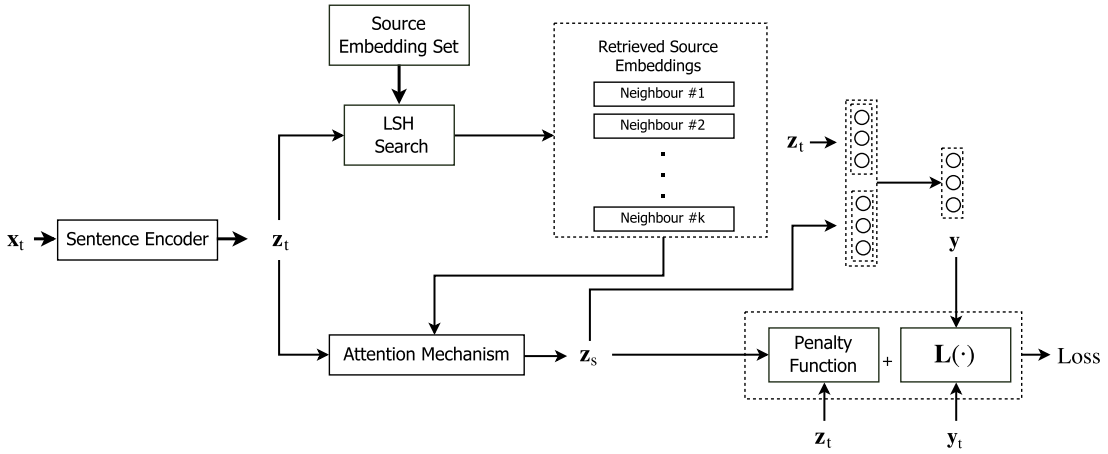

Figure 1: Proposed Model Architecture

based transfer learning methods (Pan et al., 2011).

3 Proposed Model

Given the data x with the ground truth y, supervised learning models aim to find the parameters Θ that maximizes the log-likelihood as

$$\Theta = \underset{\Theta}{\operatorname{argmax}} \log P(y|\mathrm{x}, \Theta). \quad (1)$$

To augment the learning by infusing similar source instances latent representations, a latent vector from source dataset z_s is retrieved using the data sample x_t (target dataset instance). Thus, our modified objective function can be expressed as

$$\max_{\Theta} P(y|\mathrm{x}_t, \mathrm{z}_s, \Theta). \quad (2)$$

To enforce latent representations of the instances to be similar, for better generalization across the tasks, we add a suitable penalty to the objective. The modified objective then becomes,

$$\Theta = \underset{\Theta}{\operatorname{argmax}} \log P(y|\mathrm{x}_t, \mathrm{x}_s, \Theta) - \lambda \mathcal{L}(\mathrm{z}_s, \mathrm{z}_t) \quad (3)$$

where $\mathcal{L}$ is the penalty function and λ (scale-factor) is a hyperparameter.

The subsequent sections focus on the methods to retrieve instance latent vector z_s using the data sample x_t. It is important to note that, we do not assume any structural form for P. Hence the proposed method is applicable to augment any supervised learning setting with any form for P. In the experiments we have used softmax using the bi-LSTM (Greff et al., 2015) encodings of the input as the form for P. Any state of the art text encoding scheme (Le and Mikolov, 2014) can be used here instead. The schematic representation of the model is shown in Figure 1. In the following section, we discuss the in-detail working of individual modules in Figure 1 and formulation of the penalty function $\mathcal{L}$.

Sentence Encoder: The purpose of this module is to create a vector in some latent space, encoding the semantic context of a sentence from the input sequence of words. The context vector c is obtained from an input sentence which is a sequence of word vectors $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, using a bi-LSTM (Sentence Encoder shown in Figure 1) as

$$h_t = f(x_t, h_{t-1}), \quad (4)$$

where $h_t \in \mathbb{R}^n$ is the hidden state of the bi-LSTM at time t and n is the embedding size. We combine the states at multiple time steps using a linear function g. We have,

$$o = g(\{h_1, \ldots, h_T\}),\ c = \mathrm{ReLU}(o^T W), \quad (5)$$

where $W \in \mathbb{R}^{n \times m}$ and m is a hyper parameter representing the dimension of the context vector. g in our experiments is set as

$$g(\{h_1, h_2, \ldots, h_T\}) = \frac{1}{T} \sum_{t=1}^{T} h_t. \quad (6)$$

The bi-LSTM module is responsible for generating the context vector c is pre-trained on the target classification task. A separate bi-LSTM module (sentence encoder for the source dataset) is trained on the source classification task. In our experiments we used similar modules for creating the instance embeddings of the source and target dataset, this is not constrained by the method and different modules can be used here.

Instance Retrieval: Using the obtained context vector c_t (c in Equation 5) corresponding to a target instance as a query, k-nearest neighbours are searched from the source dataset $(z_1^s, z_2^s, \ldots, z_k^s)$ using Locality Sensitive Hashing (LSH). The search mechanism using LSH takes constant time in practical scenario (Bawa et al., 2005) and therefore does not affect the training duration by large margins. Although LSH returns approximate nearest neighbours it doesn't introduce any extra loss (compared to exact nearest neighbour retrieval) in our model, as our objective is to retrieve similar instances in order to determine the class label. Even if the ranking of the instances retrieved are not accurate, retrieving multiple instances (k) reduces the chance of missing out very similar instances. The retrieved source dataset instance embeddings receive attention α_i^z, using soft-attention mechanism based on inner product similarity given as,

$$\alpha_i^z = \frac{\exp(c_t^T z_i^s)}{\sum_{j=1}^{k} \exp(c_t^T z_j^s)}, \tag{7}$$

where $c_t \in \mathbb{R}^m$ and $z_i^s, z_j^s \in \mathbb{R}^m$.

The fused instance embedding vector z_s formed after soft attention mechanism is given by,

$$z_s = \sum_{i=1}^{k} \alpha_i^z z_i^s, \tag{8}$$

where $z_s \in \mathbb{R}^m$. The retrieved instance is concatenated with the context vector c (in Equation 5) as

$$\mathrm{s} = [c_t, z_s] \text{ and } \mathbf{y} = \mathrm{softmax}(\mathrm{s}^T W^{(1)}), \tag{9}$$

where $W^{(1)} \in \mathbb{R}^{2m \times u}$, $\mathbf{y}$ is the output of the final target classification task. This model is then trained jointly with the initial parameters from the pre-trained classification module. The pre-training of the classification module is necessary because if we start from a randomly initialized context vector c_t, the LSH Forest retrieves arbitrary vectors and the model as a whole fails to converge. As the error only propagates through the attention values and penalty function it is impossible to simultaneously rectify the query and search results of the hashing mechanism.

It is important to note that the proposed model adds only a limited number of parameters over the baseline model. The extra trainable weight matrix in the model is $W^{(1)} \in \mathbb{R}^{2m \times u}$, adding only $2m \times u$, where m is the size of the context vector c and u is the number of classes.

Penalty Function: In instance-based learning, a test instance is assigned the label of the majority of its nearest-neighbour instances. This follows from the fact that similar instances belong to the same class distribution. Following the retrieval of latent vector embeddings from the source dataset, the target latent embedding is constrained to be similar to the retrieved source instances. In order to enforce this, we introduce an additional penalty along with the loss function (shown in Figure 1). The modified objective function is given as

$$\min_{\theta} L(\mathbf{y}, y_t) + \lambda ||z_s - z_t||_F^2 \ , \tag{10}$$

where $||\bullet||_F$ stands for Frobenius norm of a matrix, $\mathbf{y}$ and z_s are the outputs of the model and retrieved latent embedding respectively, y_t is the label, λ is the scale factor and z_t is the latent vector embedding of the target instance. $L(\cdot)$ in the above equation denotes the loss function used to train the model (depicted as $\mathbf{L}(\cdot)$ in Figure 1) and θ denotes the model parameters. The additional penalty term enables the latent vectors to be similar across multiple datasets, which aids performance in the subsequent stages.

4 Experiments & Results

The experiments are designed in a manner to compare the performance of the baseline model with that of external dataset augmented model. A simple *bi-LSTM (target-only)* model is trained without consideration for source-domain instances (no source-instance retrieval branch included into the network), which acts as the baseline. The embeddings of the source instances are also trained using bi-LSTM classifier. The only constraint on the embeddings is that their shape should be same across multiple domain for LSH search to take place. Our experiments shows performance enhancement across several datasets by incorporating relevant instance information from a source dataset in varying setups. Our experiments also illustrate that our proposed model continues to perform better even when the size of training set is reduced, thereby reducing the dependence on labeled data. We also demonstrate the efficacy of our model through latent vector visualizations.

Datasets & Setup: For our experiments, we have chosen three popular publicly-available news classification datasets (a) 20 Newsgroups

METHOD	TARGET	NEWS20		BBC		BBC SPORTS	
	SOURCE	BBC		NEWS20		BBC	
		Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
Bi-LSTM (*target only*)		65.17	0.6328	91.33	0.9122	84.22	0.8395
Instance-Infused Bi-LSTM		76.44	0.7586	95.35	0.9531	88.78	0.8855
Instance-Infused Bi-LSTM (*with penalty*)		**78.29**	**0.7773**	**96.09**	**0.9619**	**91.56**	**0.9100**

Table 1: Classification accuracies and F1-Scores for news arcticle classifications for different source and target domains. The first row corresponds to the baseline performance trained on the target dataset. The next two rows shows the performance of instance-infusion method with and without the penalty function.

Dataset	Train Size	Test Size	#Classes
News20	18000	2000	20
BBC	2000	225	5
BBC Sports	660	77	5

Table 2: Dataset Specifications

(News20)[1] (Lichman, 2013) (b) BBC[2] (Greene and Cunningham, 2006), (c) BBC Sports[2] (Greene and Cunningham, 2006). The datasets are chosen in such a way that all of them share common domain knowledge and have small number of training examples so that the improvement observed using instance-infusion is significant. The statistics of the three real-world datasets are mentioned in Table 2.

The mentioned datasets do not have a dedicated test set, so the evaluations were performed using *k-fold cross validation* scheme. All performance scores that are reported in this paper are the mean performance over all the folds.

Parameter	News20	BBC	BBC-Sports
Batch size	256	32	16
Learning rate	0.01	0.01	0.01
Word vector dim	300	300	300
Latent dim (m)	50	50	50
#Neighbours (k)	5	5	5
Scale factor (λ)	10^{-4}	10^{-4}	10^{-4}
# Epochs	30	20	20

Table 3: Hyper-parameters which were used in experiments for News20, BBC & BBC-Sports

The word embeddings were randomly initialized and trained along with the model. The learning rate is regulated over the training epochs, it is decreased to 0.3 times its previous value every 10 epochs. The relevant hyper-parameters are listed in Table 3.

Results: Table 1 shows the details results of our approach for all the datasets. The source and target are chosen in such a manner so that the source dataset is able to provide relevant information. In Table 1, we have shown improvements by a high margin for all datasets. For 20Newsgroups the improvement over baseline model is 12%, BBC and BBC Sports datasets show an improvement of 5%. As the proposed approach is independent of the source encoding procedure, the source instance embeddings are kept constant during training, source instances from multiple datasets can be incorporated. In the subsequent sections, we describe various setups to prove the efficacy of our model.

Instance Infusion from Same Dataset: We study the results of using the target dataset as the source for instance retrieval. This setting is same as the conventional instance-based learning setup. However, our approach not only uses the instance based information, but also leverage the macro statistics of the target dataset. The intuition behind this experimental setup is that instances from the same dataset is also useful in modeling other instances especially when a class imbalance exists in the target dataset. In this experimental setup, the *nearest neighbour retrieved is ignored* as it would be same as the instance sample being modeled during training. The performance of this setup is shown in Table 4.

Dataset Reduction with Single Source: We will discuss a set of experiments performed to support our hypothesis that the proposed model is capable of reducing the dependency on labeled instances. In these set of experiments, we show that the cross-dataset augmented models perform sig-

[1]http://qwone.com/ jason/20Newsgroups/
[2]http://mlg.ucd.ie/datasets/bbc.html

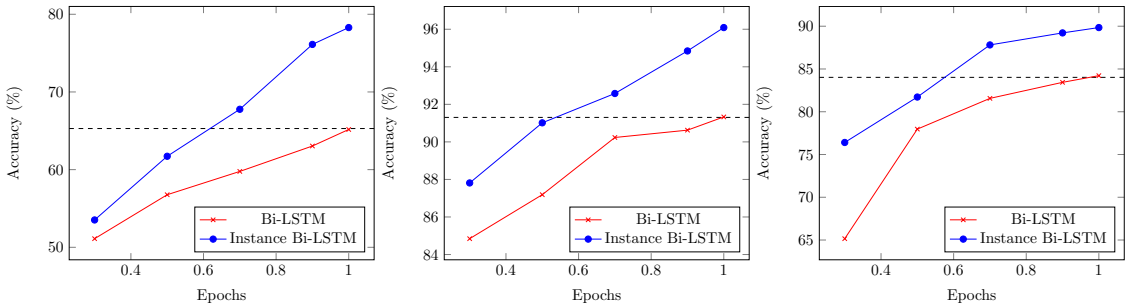

Figure 2: Accuracy Plot over dataset fractions for baseline and proposed model for (a) News20 (b) BBC (c) BBC Sports datasets. The proposed approach (in blue) beats the baseline (in red) performance by a significant margin across varying dataset fractions for all datasets.

Dataset	Acc	F1	Source
News20	77.51	0.7707	News20
BBC	96.17	0.9606	BBC
BBC Sports	90.63	0.8931	BBC Sports

Table 4: Test Accuracy for proposed model using instances from the same target dataset

nificantly better than baseline models when varying fractions of the training data is used. Figure 2 shows the variation of *instance-infused bi-LSTM* and *bi-LSTM (target-only)* performance for 20Newsgroups, BBC and BBC Sports datasets. In these set of experiments 20Newsgroups had BBC, BBC had 20Newsgroup and BBC Sports had BBC as source dataset. As shown in the plot, 0.3, 0.5, 0.7, 0.9 and 1.0 fraction of the dataset are used for performance analysis. The dashed line in the plots indicates the baseline model performance with 100% dataset support. It is observed that the performance of instance-infused bi-LSTM with 70% dataset, is better than the baseline model trained on the entire dataset. This observation shows that our proposed approach is successful in reducing the dependency on the training examples by at least 30% across all datasets.

Dataset Reduction with Multiple Source:. We design an experimental setup in which only 0.5 fraction of the target dataset is utilized and study the influence of multiple source dataset infusion. Table 6 compares the results, when single source and multiple source datasets are used for 50% dataset fraction. The results improves as and when more source datasets are used in the infusion process. This can be effectively leveraged for improving the performance of very lean datasets, by heavily deploying large datasets as source. For the single source setup, the same source datasets are used as mentioned in results section. In multiple source experiment setup, for a given target dataset the other two datasets are used as source.

Comparative Study: Table 5 gives the experimental results for our proposed approach, baselines and other conventional learning techniques on the 20 Newsgroups, BBC and BBC Sports datasets. Literature involving these datasets mostly focus on non-deep learning based approaches, we compare our results with some popular conventional learning techniques. The experiments involving conventional learning were performed using *scikit-learn* (Pedregosa et al., 2011) library in Python[3]. For the k-NN-ngram experiments, the number of nearest neighbours k was set to 5. In Table 5, the models studied are Multinomial Naive Bayes, k-nearest neighbour classifier, Support Vector Machine (SVM) (Bishop, 2006) and Random Forests Classifier. The input vectors were initialized using n-grams, bi-gram or term frequency-inverse document frequency (tf-idf). For the mentioned datasets, conventional models outperform our baseline Bi-LSTM model, however upon *instance infusion* the deep learning based model is able to achieve competitive performance across all datasets. Moreover by instance infusion the simple bi-LSTM model approaches the classical models in performance on News20 and BBC Sports dataset, whereas on BBC Dataset the proposed instance infused bi-LSTM model beats all the mentioned models. The improvement by instance infusion is 13% for News20, 5% for BBC and 8% for BBC Sports datasets. The

[3]https://www.python.org/

Model	News20		BBC		BBC Sports	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
k-NN-ngrams	35.25	0.3566	74.61	0.7376	94.59	0.9487
Multinomial NB-bigram	**79.21**	**0.7841**	95.96	0.9575	95.95	0.9560
SVM-bigram	75.04	0.7474	94.83	0.9456	93.92	0.9393
SVM-ngrams	78.60	0.7789	95.06	0.9484	95.95	0.9594
Random Forests-bigram	69.01	0.6906	87.19	0.8652	85.81	0.8604
Random Forests-ngrams	78.36	0.7697	94.83	0.9478	94.59	0.9487
Random Forests- tf-idf	78.6	0.7709	95.51	0.9547	**96.62**	**0.9660**
Bi-LSTM	65.17	0.6328	91.33	0.9122	84.22	0.8395
Instance-Infused Bi-LSTM	78.29	0.7773	**96.09**	**0.9619**	91.56	0.9100

Table 5: Comparison of results using other learning schemes on News20, BBC and BBC Sports datasets. Our approach achieves competitive performance compared to other methods across all datasets.

Dataset	Single Source		Multiple Source	
	Acc	**F1**	**Acc**	**F1**
News20	61.72	0.6133	67.32	0.6650
BBC	91.01	0.9108	91.41	0.9120
BBC Sports	81.72	0.7990	82.81	0.8027

Table 6: Test Accuracy using instances from multiple source datasets with 50% target dataset

important point to note here is that although for News20 dataset we are not able to beat the state of the art(by less than 1%), by instance infusion we are able to improve the performance of the deep learning model by a significant margin of 13%.

Visualization: We show visualizations of latent space embeddings formed using *bi-LSTM (target only)* and with *instance infusion*. In Figure 3, the latent vector embeddings of BBC Sports dataset with News20 support is shown for 0.3 in (a) & (b), 0.5 in (c) & (d) and 0.7 in (e) & (f), fraction of the target training dataset (BBC Sports). Figure 3 (f) is the embeddings representation with 70% data for which best performance (among the 6 visualizations) is observed.

It is evident from the figure that even with 30% and 50% of the data *instance infusion* tries to make the embedding distribution similar to Figure 3 (f) as seen in Figure 3 (b) and (d), when the *bi-LSTM (target-only)* instances representations in Figure 3 (a) and (c) are quite different. This illustrates that by instance infusion the latent space evolves faster to the better performing shape compared to the scenario where no instance infusion is done.

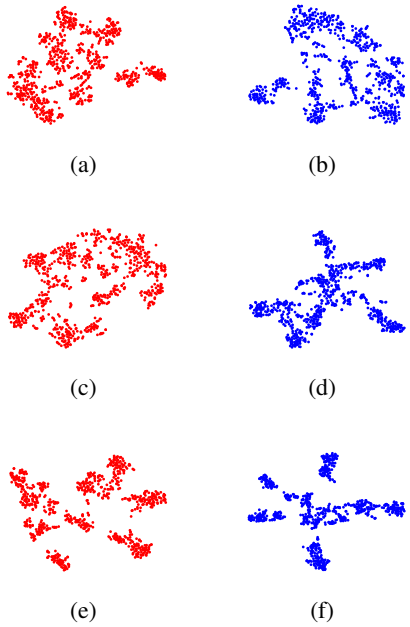

Figure 3: t-SNE visualization of LSTM latent space vectors (in red) and instance-infused embeddings (in blue) of BBC Sports with News20 as source dataset for varying dataset fractions. (a) & (b) show embeddings for 30% data fraction, (c) & (d) for 50% data, and (e) & (f) for 70% data. This figures shows the efficacy of our approach in shaping embedding space which leads to enhanced performance.

5 Related Work

The motivation behind our model comes from memory networks (Graves et al., 2014) that have an augmented long-term memory component and our model follows the general workflow in (Weston et al., 2014; Sukhbaatar et al., 2015). In our work we have incorporated instance level information using content-based attention from support dataset memory. Attention based approaches are widely used in text analysis (Bahdanau et al., 2014; Lin et al., 2017) . This approach has gained popularity in works with limited sample space. (Vinyals et al., 2016) uses a similar approach for one-shot learning however they form inference based on only support instance labels. (Snell et al., 2017) extends the idea to few shot learning in a discriminative manner by measuring distance from a class representative from a support set. (Triantafillou et al., 2017) introduced a scoring function to rank instances in a batch and optimize mean Average Precision (mAP) for few-shot learning. (Edwards and Storkey, 2016) used a generative approach for selecting representative samples for inference.

In our work, like memory network we maintain a fixed long term memory from source dataset but do not perform any modifications to it during training. We sample instances from the memory using content-based similarity but our model does not access labels like few-shot learning techniques. We present our work as a generalized approach for transfer learning across datasets sharing a common domain.

6 Conclusion & Future Work

In this work, we posit that while learning from a training data, infusion of instance level local information from an external data will improve the performance of learning algorithm, which we show through extensive experimentation on our proposed model. Although instance based learning is extensively studied in AI literature, this has rarely been used in a deep learning setup for transfer learning. An aspect of work which can be pursued to improve our setup is to incorporate a sophisticated search paradigm for instance retrieval in order to reduce latency. In this work, we have shown that our method is able to reduce the dependency on labeled data, which can also be extended to analyse performance in an unsupervised setup. Improved feature modification techniques can be augmented along with the search module in order to enhance the query formulation. We also assumed that the datasets share a common domain, in future work means to tackle domain discrepancy needs to be formulated to incorporate instances from a range of datasets.

References

Charu C Aggarwal. 2014. Instance-based learning: A survey.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Mayank Bawa, Tyson Condie, and Prasanna Ganesan. 2005. Lsh forest: self-tuning indexes for similarity search. In *Proceedings of the 14th international conference on World Wide Web*, pages 651–660. ACM.

Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM.

Harrison Edwards and Amos Storkey. 2016. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*.

Jinyang Gao, Hosagrahar Visvesvaraya Jagadish, Wei Lu, and Beng Chin Ooi. 2014. Dsh: data sensitive hashing for high-dimensional k-nnsearch. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1127–1138. ACM.

Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2015. Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*.

Arthur Gretton, Alexander J Smola, Jiayuan Huang, Marcel Schmittfull, Karsten M Borgwardt, and Bernhard Schölkopf. 2009. Covariate shift by kernel mean matching.

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

M. Lichman. 2013. UCI machine learning repository.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

James L McGaugh. 2000. Memory–a century of consolidation. *Science*, 287(5451):248–251.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. 2008. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. 2017. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pages 2252–2262.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

Bianca Zadrozny Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *In International Conference on Machine Learning ICML04*, pages 903–910.