# Facets in Transfer Learning - Across Modalities and Across Tasks

*A Project Report*

*submitted by*

**JANARTHANAN R**

*in fulfilment of the requirements*
*for the award of the degree of*

**BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY MADRAS.**

**May 2016**

# THESIS CERTIFICATE

This is to certify that the thesis titled **Facets in Transfer Learning - Across Modalities and Across Tasks**, submitted by **Janarthanan R**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

**Dr. Balaraman Ravindran**
Research Guide
Associate Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

**Dr. Kaushik Mitra**
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: May 19, 2016

# ACKNOWLEDGEMENTS

Firstly, I would like to thank my guide, Dr. Balaraman Ravindran for his great support throughout my project. He has helped me take a lot of crucial technical and personal life decisions, which have had a great positive impact in my life. He had lot of trust in me and gave me lot of freedom in research and I am very thankful to him for that. His diversity in expertise and his constant enthusiasm, especially about tough and interesting research problems has inspired me a lot. During the 2015 summer, I had the opportunity to work with Dr. Mitesh M Khapra at IBM Research Lab. Since then he has been guiding me in technical and academic life in general. I would like to thank him very much for his constant support and guidance. His planning and hard work has inspired me a lot. I would like to thank my co-guide, Prof. Kaushik Mitra for his constant support and guidance. He also gave me lot of freedom in research. I would like to thank him for his support and guidance. He has been very understanding. His patience and hard work has inspired me a lot.

Sarath Chandar has been my friend, brother, senior, guide and family. Nearly, for the past 2 and half years, he has been guiding and supporting me in all forms possible, both in technical and personal life. Though, he would not expect any of the thanks and in no way I can thank him enough, just for the record, I would say that I thank him for all his help. He has and is my role model. He is hard working, consistent, technically very strong, friendly, simple and has many more great qualities. He has inspired many people including me and I am sure, will inspire more.

I am really lucky to have made a lot of great friends during my stay here. Another person, who has had a great impact on my life, other than Sarath is Prasanna. Being my brother and friend, he has helped me in both technical and personal life. He has cared for me a lot and has always tried to put me in the right path. I would like to thank him for that. I would like to thank my friend Venkatesh, who has supported and guided me in many of my endeavours. He has spent a lot of time and energy to support my progress. I especially thank him for his great support and help during my graduate admission process.

I would also like to thank all my other friends, especially Anil Kumar and Aravind Srinivas for their wonderful technical and non-technical discussions, which have helped me a lot. Apart from the great deal of help and support, Anil has also inspired me a lot through his hard work. I would like to thank my parents, sister and family for their full support, without which non of my progress would have been possible.

It was through all of your support that I have progressed. I am sure our friendship and collaborations will continue and will achieve great heights together. Thank you everyone.

# ABSTRACT

KEYWORDS: Transfer Learning, Deep Learning, Reinforcement Learning, Common Representation Learning, Multimodal Learning

Transfer learning plays a very crucial role in intelligence. There are two main parts to it. One, the ability to transfer knowledge between different modalities and the other, the ability to transfer knowledge from previously learnt tasks and use it in a new task.

One way to address the former, is to learn common representations for multiple views of data. Typically, such common representations are learned using a parallel corpus between the two views (say, 1M images and their English captions). In this work, we address a real-world scenario where no direct parallel data is available between two views of interest (say, $V_1$ and $V_2$) but parallel data is available between each of these views and a pivot view ($V_3$). We propose a model for learning a common representation for $V_1$, $V_2$ and $V_3$ using only the parallel data available between $V_1V_3$ and $V_2V_3$. The proposed model is generic and even works when there are $n$ views of interest and only one pivot view which acts as a bridge between them. There are two specific downstream applications that we focus on (i) transfer learning between languages $L_1$,$L_2$,...,$L_n$ using a pivot language $L$ and (ii) cross modal access between images and a language $L_1$ using a pivot language $L_2$. Our model achieves state-of-the-art performance in multilingual document classification on the publicly available multilingual TED corpus and promising results in multilingual multimodal retrieval on a new dataset created and released as a part of this work.

The latter, i.e, the ability to transfer knowledge from learnt source tasks to a new target task can be very useful in speeding up the learning process of an agent. This has been receiving a lot of attention, but the application of transfer poses two serious challenges which have not been adequately addressed in the past. First, the agent should be able to avoid negative transfer, which happens when the transfer hampers or slows down the learning instead of speeding it up. Secondly, the agent should be able to do selective transfer, which is the ability to select and transfer from different and multiple source tasks for different parts of the state space of the target task. We propose ADAAPT: A Deep Architecture for Adaptive Policy Transfer, which addresses these challenges. We test ADAAPT using two different instantiations: One as ADAAPTive REINFORCE algorithm for direct policy search and another as ADAAPTive Actor-Critic where the actor uses ADAAPT. Empirical evaluations on simulated domains show that ADAAPT can be effectively used for policy transfer from multiple source MDPs sharing the same state and action space.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Overview

Many tasks which humans perform easily and excellently, such as image recognition, speech recognition and language understanding, are found to be tough for machines. This is partly because, we do not fully understand how we perform these tasks and hence find it tough to make machines perform them. There is also an important difference in the way, we humans do these tasks and most present day machines do them. Even while performing any of the individual tasks, which might be a new task and involve a particular modality (such as, text), we humans use the valuable knowledge from many other modalities (such as audio, video and images) and many other previously performed or learnt tasks seamlessly. This makes our learning faster and better. This requires the following abilities. Firstly, the ability to combine the knowledge from different modalities. Then, we also need to transfer knowledge between these different modalities, as different modalities might have different information. One way of achieving this is to learn common representation for data from different modalities. While performing tasks which involve decision making, we also use the knowledge of other previously learnt tasks, instead of learning the new task from scratch.

Recently, there has been a lot of interest in learning common representations for different views of the data. These views could be different modalities, like images and their captions or different languages, like the documents in different languages. Typically these representations are learnt, when there is parallel data available between all the views. A more general and real world scenario is when there is no parallel data available between all the views, but there is parallel data between these views and a pivot view. For example, we could have "image and its English captions" and "English and its parallel French documents". We do not have any "image and French captions". Can we learn a common representation between image, English and French captions? We propose Bridge Correlational Neural Networks, that can learn common representations for the different views in such scenarios. Apart from the fact that transfer between the various views is possible, once we build the common representations, there is another type of transfer happening here. The pivot view (English in the above example) acts as a bridge and transfers knowledge between the non-pivot views (Image and French captions). We evaluate our model on two tasks, one on the cross language document classification across 11 languages with English as the pivot language and the other on the cross modal retrieval task, where we retrieve image given French (or German) captions and vice versa when the training data only has image-English and English-French (or German) parallel data.

Coming to the part of transferring between tasks, the agent can converge faster and to a better policy, if it can transfer its knowledge from the learnt source tasks to the new current, target task. This has a few serious challenges that have not been dealt with adequately. First is the problem of negative transfer, which happens when the transfer from the source tasks degrades the performance of the agent in the target task. This problem has restricted the application of many previously proposed transfer learning techniques to restricted scenarios. Second is the ability to perform selective transfer. This is the ability to transfer from multiple different source tasks for different parts of the state space of the target task. We propose ADAAPT: A Deep Architecture for Adaptive Policy Transfer from multiple sources. This architecture has a neural network based attention mechanism which, during the occurrence of negative transfer uses a modifiable randomly initialised policy, thus avoiding negative transfer. It also has the ability to use policies from different and multiple source tasks at the granularity of state, thus performing selective transfer. We evaluate this model on a well designed set of grid world problem and verified the different capabilities proposed.

The rest of the thesis is organised as follows. In chapter 2, we will look at Bridge Correlational Neural Networks for learning representation for multilingual and multi-modal data. In chapter 3, we look at ADAAPT architecture for transfer learning, avoiding negative transfer and performing selective transfer. Chapter 4 concludes the thesis by giving a summary of the work done and discussing possible future works.

# Chapter 2

# Bridge Correlational Neural Networks for Multilingual Multimodal Representation Learning

## 2.1 Introduction

The proliferation of multilingual and multimodal content online has ensured that multiple views of the same data exist. For example, it is common to find the same article published in multiple languages online in multilingual news articles, multilingual wikipedia articles, *etc*. Such multiple views can even belong to different modalities. For example, images and their textual descriptions are two views of the same entity. Similarly, audio, video and subtitles of a movie are multiple views of the same entity.

Learning common representations for such multiple views of data will help in several downstream applications. For example, learning a common representation for audio and subtitles could help in generating subtitles from a given audio. Similarly, learning a common representation for images and their textual descriptions could help in finding images which match a given textual description. Further, such common representations can also facilitate transfer learning between views. For example, a document classifier trained on one language (view) can be used to classify documents in another language by representing documents of both languages in a common subspace.

Existing approaches to common representation learning Ngiam *et al.* (2011); Klementiev *et al.* (2012); Chandar *et al.* (2013, 2014); Andrew *et al.* (2013); Wang *et al.* (2015) except Hermann and Blunsom (2014*b*) typically require parallel data between all views. However, in many real-world scenarios such parallel data may not be available. For example, while there are many publicly available datasets containing images and their corresponding English captions, it is very hard to find datasets containing images and their corresponding captions in Russian, Dutch, Hindi, Urdu, *etc.* In this work, we are interested in addressing such scenarios. More specifically, we consider scenarios where we have $n$ different views but parallel data is only available between each of these views, and a pivot view. In particular, there is no parallel data available between the non-pivot views. For example, consider the case where the two views of interest are images and French captions. Suppose, there is no direct parallel data between these two views but parallel data is available between (i) images and English captions and (ii) English and French texts. We propose to use English as a pivot view and learn a common representation for English text, French text and images.

To this end, we propose Bridge Correlational Neural Networks (Bridge CorrNets) which learn aligned representations across multiple views using a pivot view. We build on the work of Chandar *et al.* (2016) but unlike their model, which only addresses scenarios where direct parallel data is available between two views, our model can work for $n(\geq 2)$ views even when no parallel data is available between all of them. Our model only requires parallel data between each of these $n$ views and a pivot view. During training, our model maximizes the correlation between the representations of the pivot view and each of the $n$ views. Intuitively, the pivot view ensures that similar entities across different views get mapped close to each other since the model would learn to map each of them close to the corresponding entity in the pivot view. Pivot view in essence acts as a bridge, and transfers knowledge across the non-pivot views.

We evaluate our approach using two downstream applications. First, we employ our model to facilitate transfer learning between multiple languages using English as the pivot language. For this, we do an extensive evaluation using 110 source-target language pairs and clearly show that we outperform the current state-of-the art approach Hermann and Blunsom (2014*b*). Second, we employ our model to enable cross modal access between images and French/German captions using English as the pivot view. For this, we created a test dataset consisting of images and their captions in French and German in addition to the English captions which were publicly available. To the best of our knowledge, this task of retrieving images given French/German captions (and vice versa) without direct parallel training data between them has not been addressed in the past. Even on this task we report promising results. Code and data used can be downloaded from `http://sarathchandar.in/bridge-cornet`.

## 2.2 Related Work

Canonical Correlation Analysis (CCA) and its variants Hotelling (1936); Vinod (1976); Nielsen *et al.* (1998); Cruz-Cano and Lee (2014); Akaho (2001) are the most commonly used methods for learning a common representation for two views. However, most of these models generally work with two views only. Even though there are multi-view generalizations of CCA Tenenhaus and Tenenhaus (2011); Luo *et al.* (2015), their computational complexity makes them unsuitable for larger data sizes.

Another class of algorithms for multiview learning is based on Neural Networks. One of the earliest neural network based model for learning common representations was proposed in Hsieh (2000). Recently, there has been a renewed interest in this field and several neural network based models have been proposed. For example, Multimodal Autoencoder Ngiam *et al.* (2011), Deep Canonically Correlated Autoencoder Wang *et al.* (2015), Deep CCA Andrew *et al.* (2013) and Correlational Neural Networks (CorrNet) Chandar *et al.* (2016). CorrNet performs better than most of the above mentioned methods and we build on their work as discussed in the next section.

One of the tasks that we address in this work is multilingual representation learning where the aim is to learn aligned representations for words across languages. Some notable neural network based approaches here include the works of Klementiev *et al.* (2012); Zou *et al.* (2013); Mikolov *et al.* (2013); Hermann and Blunsom (2014*b*,*a*); Chandar *et al.* (2014); Soyer *et al.* (2015); Gouws *et al.* (2015). However, except for Hermann and Blunsom (2014*a*,*b*), none of these other works handle the case when parallel data is not available between all languages. Our model addresses this issue and outperforms the model of HermannK2014.

The task of cross modal access between images and text addressed in this work comes under MultiModal Representation Learning where each view belongs to a different modality. ngiam11 proposed an autoencoder based solution to learning common representation for audio and video. JMLRv15srivastava14b extended this idea to RBMs and learned common representations for image and text. Other solutions for image/text representation learning include Zheng *et al.* (2014*a*,*b*); Socher *et al.* (2014). All these approaches require parallel data between the two views and do not address multimodal, multilingual learning in situations where parallel data is available only between different views and a pivot view.

In the past, pivot/bridge languages have been used to facilitate MT (for example, Wu and Wang (2007); Cohn and Lapata (2007); Utiyama and Isahara (2007); Nakov and Ng (2009)), transitive CLIR Ballesteros (2000); Lehtokangas *et al.* (2008), transliteration and transliteration mining Khapra *et al.* (2010*a*); Kumaran *et al.* (2010); Khapra *et al.* (2010*b*); Zhang *et al.* (2011). None of these works use neural networks but it is important to mention them here because they use the concept of a pivot language (view) which is central to our work.

## 2.3 Bridge Correlational Neural Network

In this section, we describe Bridge CorrNet which is an extension of the CorrNet model proposed by Chandar *et al.* (2016). They address the problem of learning common representations between two views when parallel data is available between them. We propose an extension to their model which simultaneously learns a common representation for $M$ views when parallel data is available only between one pivot view and the remaining $M-1$ views.

Let these views be denoted by $V_1, V_2, ..., V_M$ and let $d_1, d_2, ..., d_M$ be their respective dimensionalities. Let the training data be $\mathcal{Z} = \{z^i\}_{i=1}^N$ where each training instance contains only two views, *i.e.*, $z^i = (v_j^i, v_M^i)$ where $j \in \{1, 2, .., M-1\}$ and $M$ is a pivot view. To be more clear, the training data contains $N_1$ instances for which $(v_1^i, v_M^i)$ are available, $N_2$ instances for which $(v_2^i, v_M^i)$ are available and so on till $N_{M-1}$ instances for which $(v_{M-1}^i, v_M^i)$ are available (such that $N_1 + N_2 + ... + N_{M-1} = N$). We denote each of these disjoint pairwise training sets by $\mathcal{Z}_1$, $\mathcal{Z}_2$ to $\mathcal{Z}_{M-1}$ such that $\mathcal{Z}$ is the union of all these sets.

As an illustration consider the case when English, French and German texts are the three views of interest with English as the pivot view. As training data, we have $N_1$ instances containing English and their corresponding French texts and $N_2$ instances containing English and their corresponding German texts. We are then interested in learning a common representation for English, French and German even though we do not have any training instance containing French and their corresponding German texts.

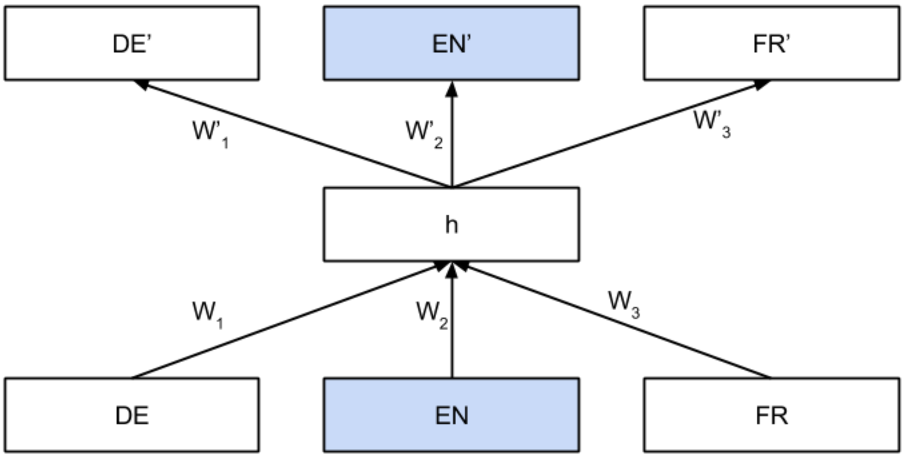



Figure 2.1: Bridge Correlational Neural Network. The views are English, French and German with English being the pivot view.

Bridge CorrNet uses an encoder-decoder architecture with a correlation based regularizer to achieve this. It contains one encoder-decoder pair for each of the $M$ views. For each view $V_j$, we have,

$$h_{V_j}(v_j) = f(W_j v_j + b) \tag{2.1}$$

where $f$ is any non-linear function such as sigmoid or tanh, $W_j \in \mathbb{R}^{k \times d_j}$ is the encoder matrix for view $V_j$, $b \in \mathbb{R}^k$ is the common bias shared by all the encoders. We also compute a hidden representation for the concatenated training instance $z = (v_j, v_M)$ using the following encoder function:

$$h_Z(z) = f(W_j v_j + W_{{}_M} v_{{}_M} + b) \tag{2.2}$$

In the remainder of this thesis, whenever we drop the subscript for the encoder, then the encoder is determined by its argument. For example $h(v_j)$ means $h_{V_j}(v_j)$, $h(z)$ means $h_Z(z)$ and so on.

Our model also has a decoder corresponding to each view as follows:

$$g_{V_j}(h) = p(W_j' h + c_j) \tag{2.3}$$

where $p$ can be any activation function, $W_j' \in \mathbb{R}^{d_j \times k}$ is the decoder matrix for view $V_j$, $c_j \in \mathbb{R}^{d_j}$ is the decoder bias for view $V_j$. We also define $g(h)$ as simply the concatenation of $[g_{V_j}(h), g_{V_M}(h)]$.

In effect, $h_{V_j}(.)$ encodes the input $v_j$ into a hidden representation $h$ and then $g_{V_j}(.)$ tries to decode/reconstruct $v_j$ from this hidden representation $h$. Note that $h$ can be computed using $h(v_j)$ or $h(v_M)$. The decoder can then be trained to decode/reconstruct both $v_j$ and $v_M$ given a hidden representation computed using any one of them. More formally, we train Bridge CorrNet by minimizing the following objective function:

$$\begin{aligned}\mathcal{J}_{\mathcal{Z}}(\theta) &= \sum_{i=1}^{N} L(z^i, g(h(z^i))) + \sum_{i=1}^{N} L(z^i, g(h(v^i_{l(i)}))) \\ &+ \sum_{i=1}^{N} L(z^i, g(h(v^i_{M}))) - \lambda \operatorname{corr}(h(V_{l(i)}), h(V_M)) \end{aligned} \tag{2.4}$$

where $l(i) = j$ if $z^i \in \mathcal{Z}_j$ and the correlation term corr is defined as follows:

$$\operatorname{corr} = \frac{\sum_{i=1}^{N}(h(x^i) - \overline{h(X)})(h(y^i) - \overline{h(Y)})}{\sqrt{\sum_{i=1}^{N}(h(x^i) - \overline{h(X)})^2 \sum_{i=1}^{N}(h(y^i) - \overline{h(Y)})^2}} \tag{2.5}$$

Note that $g(h(z^i))$ is the reconstruction of the input $z^i$ after passing through the encoder and decoder. $L$ is a loss function which captures the error in this reconstruction, $\lambda$ is the scaling parameter to scale the last term with respect to the remaining terms, $\overline{h(X)}$ is the mean vector for the hidden representations of the first view and $\overline{h(Y)}$ is the mean vector for the hidden representations of the second view.

We now explain the intuition behind each term in the objective function. The first term captures the error in reconstructing the concatenated input $z^i$ from itself. The second term captures the error in reconstructing both views given the non-pivot view, $v^i_{l(i)}$. The third term captures the error in reconstructing both views given the pivot view, $v^i_M$. Minimizing the second and third terms ensures that both the views can be predicted from any one view. Finally, the correlation term ensures that the network learns correlated common representations for all views.

Our model can be viewed as a generalization of the two-view CorrNet model proposed in Chandar *et al.* (2016). By learning joint representations for multiple views using disjoint training sets $\mathcal{Z}_1$, $\mathcal{Z}_2$ to $\mathcal{Z}_{M-1}$ it eliminates the need for ${}^nC_2$ pair-wise parallel datasets between all views of interest. The pivot view acts as a bridge and ensures that similar entities across different views get mapped close to each other since all of them would be close to the corresponding entity in the pivot view.

Note that unlike the objective function of CorrNet Chandar *et al.* (2016), the objective function of Equation 2.4, is a dynamic objective function which changes with each training instance. In other words, $l(i) \in \{1, 2, .., M-1\}$ varies for each $i \in \{1, 2, .., N\}$. For efficient implementation, we construct mini-batches where each mini-batch will come from only one of the sets $\mathcal{Z}_1$ to $\mathcal{Z}_{M-1}$. We randomly shuffle these mini-batches and use corresponding objective function for each mini-batch.

**Algorithm 1:** Train Bridge CorrNet

**Input:** Number of views $M$, training sets $\mathcal{Z}_1$, $\mathcal{Z}_2$ to $\mathcal{Z}_{M-1}$, $\lambda$, mini-batch size $k$.
**Output:** Learned parameters

Initialize the encoder/decoder for each view.
**for** *each epoch* **do**
  **while** *not seen all examples* **do**
    Randomly sample $i \in \{1, 2, ..., M-1\}$.
    Sample random mini-batch from $\mathcal{Z}_i$.
    Use the mini-batch and compute gradient for equation (4).
    Update the parameters.
  **end**
**end**

As a side note, we would like to mention that in addition to $\mathcal{Z}_1$, $\mathcal{Z}_2$ to $\mathcal{Z}_{M-1}$ as defined earlier, if additional parallel data is available between some of the non-pivot views then the objective function can be suitably modified to use this parallel data to further improve the learning. However, this is not the focus of this work and we leave this as a possible future work.

## 2.4 Datasets

In this section, we describe the two datasets that we used for our experiments.

### 2.4.1 Multlingual TED corpus

HermannK2014 provide a multilingual corpus based on the TED corpus for IWSLT 2013 Cettolo *et al.* (2012). It contains English transcriptions of several talks from the TED conference and their translations in multiple languages. We use the parallel data between English and other languages for training Bridge Corrnet (English, thus, acts as the pivot langauge). HermannK2014 also propose a multlingual document classification task using this corpus. The idea is to use the keywords associated with each talk (document) as class labels and then train a classifier to predict these classes. There are one or more such keywords associated with each talk but only the 15 most frequent keywords across all documents are considered as class labels. We used the same pre-processed splits[1] as provided by Hermann and Blunsom (2014*b*). The training corpus consists of a total of 12,078 parallel documents distributed across 12 language pairs.

### 2.4.2 Multilingual Image Caption dataset

The MSCOCO dataset[2] contains images and their English captions. On an average there are 5 captions per image. The standard train/valid/test splits for this dataset are also available online. However, the reference captions for the images in the test split are not provided. Since we need such reference captions for evaluations, we create a new train/valid/test of this dataset. Specifically, we take 80K images from the standard train split and 40K images from the standard valid split. We then randomly split the merged 120K images into train(118K), validation (1K) and test set (1K).

We then create a multilingual version of the test data by collecting French and German translations for all the 5 captions for each image in the test set. We use crowd-sourcing to do this. We used the CrowdFlower platform and solicited one French and one German translation for each of the 5000 captions using native speakers. We got

[1] `http://www.clg.ox.ac.uk/tedcorpus`
[2] `http://mscoco.org/dataset/`

each translation verified by 3 annotators. We restricted the geographical location of annotators based on the target language. We found that roughly 70% of the French translations and 60% of the German translations were marked as correct by a majority of the verifiers. On further inspection with the help of in-house annotators, we found that the errors were mainly syntactic and the content words are translated correctly in most of the cases. Since none of the approaches described in this work rely on syntax, we decided to use all the 5000 translations as test data. This multilingual image caption test data (MIC test data) will be made publicly available[3] and will hopefully assist further research in this area.

## 2.5 Experiment 1: Transfer learning using a pivot language

From the TED corpus described earlier, we consider English transcriptions and their translations in 11 languages, *viz.*, Arabic, German, Spanish, French, Italian, Dutch, Polish, Portuguese (Brazilian), Roman, Russian and Turkish. Following the setup of HermannK2014, we consider the task of cross language learning between each of the $^{11}C_2$ non-English language pairs. The task is to classify documents in a language when no labeled training data is available in this language but training data is available in another language. This involves the following steps:

**1. Train classifier:** Consider one language as the source language and the remaining 10 languages as target languages. Train a document classifier using the labeled data of the source language, where each training document is represented using the hidden representation computed using a trained Bridge Cornet model. As in Hermann and Blunsom (2014*b*) we used an averaged perceptron trained for 10 epochs as the classifier for all our experiments. The train split provided by Hermann and Blunsom (2014*b*) is used for training.

**2. Cross language classification:** For every target language, compute a hidden representation for every document in its test set using Bridge CorrNet. Now use the classifier trained in the previous step to classify this document. The test split provided by Her-

[3] `http://sarathchandar.in/bridge-cornet`

| **Language** | $\|V\|$ | **Langauge** | $\|V\|$ |
|---|---|---|---|
| Arabic | 60326 | Dutch | 31213 |
| English | 42897 | Polish | 55983 |
| German | 37096 | Pt-Br | 33548 |
| Spanish | 35345 | Rom'n | 43968 |
| French | 34146 | Russian | 50734 |
| Italian | 37961 | Turkish | 58697 |

Table 2.1: The size of the vocabulary used for each language.

| Training | Test Language | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Language | Arabic | German | Spanish | French | Italian | Dutch | Polish | Pt-Br | Rom'n | Russian | Turkish |
| Arabic | | **0.662** | **0.654** | **0.645** | **0.663** | **0.654** | **0.626** | **0.628** | **0.630** | **0.607** | **0.644** |
| German | **0.920** | | **0.544** | **0.505** | **0.654** | **0.672** | **0.631** | **0.507** | **0.583** | **0.537** | **0.597** |
| Spanish | **0.666** | **0.465** | | **0.547** | **0.512** | **0.501** | **0.537** | **0.518** | **0.573** | **0.463** | **0.434** |
| French | **0.761** | **0.585** | **0.679** | | **0.681** | **0.646** | **0.671** | **0.650** | **0.675** | **0.613** | **0.578** |
| Italian | **0.701** | **0.421** | 0.456 | 0.457 | | **0.530** | **0.442** | **0.491** | **0.390** | **0.402** | **0.499** |
| Dutch | **0.847** | **0.370** | **0.511** | **0.472** | **0.600** | | **0.536** | **0.489** | **0.458** | **0.470** | **0.516** |
| Polish | **0.533** | **0.387** | **0.556** | **0.535** | **0.536** | **0.454** | | **0.446** | **0.521** | **0.473** | **0.413** |
| Pt-Br | **0.609** | **0.502** | **0.572** | **0.553** | **0.548** | **0.535** | **0.545** | | **0.557** | **0.451** | **0.463** |
| Rom'n | **0.573** | **0.460** | **0.559** | **0.530** | **0.521** | **0.484** | **0.475** | **0.485** | | **0.486** | **0.458** |
| Russian | **0.755** | **0.460** | **0.537** | **0.437** | **0.567** | **0.499** | **0.550** | **0.478** | **0.475** | | **0.484** |
| Turkish | **0.950** | **0.373** | **0.480** | **0.452** | **0.542** | **0.544** | **0.585** | 0.297 | **0.512** | **0.412** | |

Table 2.2: F1-scores for TED corpus document classification results when training and testing on two languages that do not share any parallel data. We train a Bridge CorrNet model on all en-L2 language pairs together, and then use the resulting embeddings to train document classifiers in each language. These classifiers are subsequently used to classify data from all other languages.

mann and Blunsom (2014*b*) is used for testing.

### 2.5.1 Training and tuning Bridge Cornet

For the above process to work, we first need to train Bridge Cornet so that it can then be used for computing a common hidden representation for documents in different languages. For training Bridge CorrNet, we treat English as the pivot language (view) and construct parallel training sets $\mathcal{Z}_1$ to $\mathcal{Z}_{11}$. Every instance in $\mathcal{Z}_1$ contains the English and Arabic view of the same talk (document). Similarly, every instance in $\mathcal{Z}_2$ contains the English and German view of the same talk (document) and so on. For every language, we first construct a vocabulary containing all words appearing more than 5 times in the corpus (all talks) of that language. We then use this vocabulary to construct a bag-of-words representation for each document. The size of the vocabulary ($|V|$) for different languages varied from 31213 to 60326 words. To be more clear, $v_1 = v_{arabic} \in \mathbb{R}^{|V|_{arabic}}$, $v_2 = v_{german} \in \mathbb{R}^{|V|_{german}}$ and so on.

We train our model for 10 epochs using the above training data $\mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_{11}\}$. We use hidden representations of size $D = 128$, as in Hermann and Blunsom (2014*b*).

| Training Language | Test Language | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Arabic | German | Spanish | French | Italian | Dutch | Polish | Pt-Br | Rom'n | Russian | Turkish |
| Arabic | | 0.378 | 0.436 | 0.432 | 0.444 | 0.438 | 0.389 | 0.425 | 0.42 | 0.446 | 0.397 |
| German | 0.368 | | 0.474 | 0.46 | 0.464 | 0.44 | 0.375 | 0.417 | 0.447 | 0.458 | 0.443 |
| Spanish | 0.353 | 0.355 | | 0.42 | 0.439 | 0.435 | 0.415 | 0.39 | 0.424 | 0.427 | 0.382 |
| French | 0.383 | 0.366 | 0.487 | | 0.474 | 0.429 | 0.403 | 0.418 | 0.458 | 0.415 | 0.398 |
| Italian | 0.398 | 0.405 | **0.461** | **0.466** | | 0.393 | 0.339 | 0.347 | 0.376 | 0.382 | 0.352 |
| Dutch | 0.377 | 0.354 | 0.463 | 0.464 | 0.46 | | 0.405 | 0.386 | 0.415 | 0.407 | 0.395 |
| Polish | 0.359 | 0.386 | 0.449 | 0.444 | 0.43 | 0.441 | | 0.401 | 0.434 | 0.398 | 0.408 |
| Pt-Br | 0.391 | 0.392 | 0.476 | 0.447 | 0.486 | 0.458 | 0.403 | | 0.457 | 0.431 | 0.431 |
| Rom'n | 0.416 | 0.32 | 0.473 | 0.476 | 0.46 | 0.434 | 0.416 | 0.433 | | 0.444 | 0.402 |
| Russian | 0.372 | 0.352 | 0.492 | 0.427 | 0.438 | 0.452 | 0.43 | 0.419 | 0.441 | | 0.447 |
| Turkish | 0.376 | 0.352 | 0.479 | 0.433 | 0.427 | 0.423 | 0.439 | **0.367** | 0.434 | 0.411 | |

Table 2.3: F1-scores for TED corpus document classification results when training and testing on two languages that do not share any parallel data. Same procedure as Table 2.2, but with DOC/ADD model in Hermann and Blunsom (2014*b*).

| Setting | Languages | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Arabic | German | Spanish | French | Italian | Dutch | Polish | Pt-Br | Rom'n | Russian | Turkish |
| Raw Data NB | **0.469** | **0.471** | 0.526 | **0.532** | 0.524 | 0.522 | 0.415 | 0.465 | **0.509** | 0.465 | **0.513** |
| DOC/ADD (Single) | 0.422 | 0.429 | 0.394 | 0.481 | 0.458 | 0.252 | 0.385 | 0.363 | 0.431 | **0.471** | 0.435 |
| DOC/BI (Single) | 0.432 | 0.362 | 0.336 | 0.444 | 0.469 | 0.197 | 0.414 | 0.395 | 0.445 | 0.436 | 0.428 |
| DOC/ADD (Joint) | 0.371 | 0.386 | 0.472 | 0.451 | 0.398 | 0.439 | 0.304 | 0.394 | 0.453 | 0.402 | 0.441 |
| DOC/BI (Joint) | 0.329 | 0.358 | 0.472 | 0.454 | 0.399 | 0.409 | 0.340 | 0.431 | 0.379 | 0.395 | 0.435 |
| Bridge CorrNet | 0.266 | 0.456 | **0.535** | 0.529 | **0.551** | **0.565** | **0.478** | **0.535** | 0.490 | 0.447 | 0.477 |

Table 2.4: : F1-scores on the TED corpus document classification task when training and evaluating on the same language. Results other than Bridge CorrNet are taken from Hermann and Blunsom (2014*b*).

Further, we used stochastic gradient descent with mini-batches of size 20. Each mini-batch contains data from only one of the $\mathcal{Z}_i s$. We get a stochastic estimate for the correlation term in the objective function using this mini-batch. The hyperparameter $\lambda$ was tuned to each task using a training/validation split for the source language and using the performance on the validation set of an averaged perceptron trained on the training set (notice that this corresponds to a monolingual classification experiment, since the general assumption is that no labeled data is available in the target language).

### 2.5.2 Results

Before presenting the results for our cross language classification experiment, we would first like to give a qualitative feel for the representations learned using Bridge CorrNet. For this, we randomly select a few English words and find their nearest neighbors in different languages based on the representations learned using Bridge CorrNet. These English words and their neighbors are shown in Table 2.5 2.6. In almost all the cases the nearest neighbors of the English words turn out to be their exact translations or highly semantically related words. Also, we observed that the representations of translation

| English word | Languages | | | | |
|---|---|---|---|---|---|
| | Spanish | French | Italian | Dutch | Polish |
| market | mercado | marché | mercato | markt | mercado |
| | market | market | market | arbeidsmarkt | lançadas |
| | place | boursier | vendita | marktonderzoek | mercados |
| | comercializan | marketer | azionario | marktaandeel | timbuktu |
| oil | petróleo | pétrole | petrolio | olie | petróleo |
| | aceite | l'huile | olio | olieprijs | óleo |
| | petroleros | d'huile | l'olio | olieprijzen | azeite |
| | crudo | pétrolières | dell'olio | olieramp | derramamento |
| home | casa | maison | casa | thuis | casa |
| | hogar | foyer | tramandare | huis | lar |
| | casas | domicile | dimora | woning | casas |
| | hogares | rentre | casetta | thuisblijven | lares |
| history | historia | l'histoire | storia | geschiedenis | história |
| | historial | d'histoire | qualcuno | wereldgeschiedenis | histórico |
| | contado | histoire | dell'umanità | history | historia |
| | participó | historique | popolo | historie | britannica |

Table 2.5: English words and their nearest neighbours in different languages (based on Euclidean distance).

| English word | Languages | | | |
|---|---|---|---|---|
| | Pt-Br | Rom'n | Russian | Turkish |
| market | rynku | piaţă | рынок | pazar |
| | rynek | piaţă | рынка | piyasa |
| | giełdzie | piaţa | рынке | pazara |
| | targ | piata | рыночные | pazarın |
| oil | ropy | petrol | нефти | petrol |
| | ropa | petrolul | нефть | petrolün |
| | ropę | petrolului | нефтью | petrolü |
| | oleju | ulei | масло | petrolden |
| home | domu | acasă | домой | eve |
| | dom | acasa | дома | evde |
| | domem | casă | дом | ev |
| | domach | casa | доме | evine |
| history | historii | istoria | истории | tarih |
| | historię | istorie | историю | tarihin |
| | historia | istoriei | история | tarihi |
| | dziejach | poveste | историей | tarihinde |

Table 2.6: English words and their nearest neighbours in different languages (based on Euclidean distance).

pairs in non-English languages (say, French and German) are also transitively close to each other due to the pivot language.

We now present the results of our cross language classification task in Table 2.2. Each row corresponds to a source language and each column corresponds to a target language. We report the average F1-scores over all the 15 classes. We compare our results with the best results reported in Hermann and Blunsom (2014*b*) (see Table 2.3). Out of the 110 experiments, our model outperforms the model of Hermann and Blunsom (2014*b*) in 107 experiments. This suggests that our model efficiently exploits the pivot language to facilitate cross language learning between other languages.

Finally, we present the results for a monolingual classification task in Table 2.4. The idea here is to see if learning common representations for multiple views can also help in improving the performance of a task involving only one view. HermannK2014 argue that a Naive Bayes (NB) classifier trained using a bag-of-words representation of the documents is a very strong baseline. In fact, a classifier trained on document representations learned using their model does not beat a NB classifier for the task of monolingual classification. Rows 2 to 5 in Table 2.4 show the different settings tried by them (we refer the reader to Hermann and Blunsom (2014*b*) for a detailed description of these settings). On the other hand our model is able to beat NB for 5/11 languages. Further, for 4 other languages (German, French, Romanian, Russian) its performance is only marginally poor than that of NB.

## 2.6 Experiment 2: Cross modal access using a pivot language

In this experiment, we are interested in retrieving images given their captions in French (or German) and vice versa. However, for training we do not have any parallel data containing images and their French (or German) captions. Instead, we have the following datasets: (i) a dataset $\mathcal{Z}_1$ containing images and their English captions and (ii) a dataset $\mathcal{Z}_2$ containing English and their parallel French (or German) documents. For $\mathcal{Z}_1$, we use the training split of MSCOCO dataset which contains 118K images and their English captions (see Section 2.4.2). For $\mathcal{Z}_2$, we use the English-French (or German) parallel documents from the train split of the TED corpus (see Section 2.4.1). We use English as the pivot language and train Bridge Corrnet using $\mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2\}$ to learn common representations for images, English text and French (or German) text. For text, we use bag-of-words representation and for image, we use the 4096 (fc6) representation got from a pretrained ConvNet (BVLC Reference CaffeNet Jia *et al.* (2014)). We learn hidden representations of size $D = 200$ by training Bridge Cornet for 20 epochs using stochastic gradient descent with mini-batches of size $20$. Each mini-batch contains data from only one of the $\mathcal{Z}_i s$.

For the task of retrieving captions given an image, we consider the 1000 images in our test set (see section 2.4.2) as queries. The 5000 French (or German) captions corresponding to these images (5 per image) are considered as documents. The task is then to retrieve the relevant captions for each image. We represent all the captions and images in the common space as computed using Bridge Cornet. For a given query, we rank all the captions based on the Euclidean distance between the representation of the image and the caption. For the task of retrieving images given a caption, we simply reverse the role of the captions and images. In other words, each of the 5000 captions is treated as a query and the 1000 images are treated as documents. $\lambda$ was tuned to each task using a training/validation split. For the task of retrieving French/German captions given an image, $\lambda$ was tuned using the performance on the validation set for retrieving French (or German) sentences for a given English sentence. For the other task, $\lambda$ was tuned using the performance on the validation set for retrieving images, given English captions. We do not use any image-French/German parallel data for tuning the hyperparameters.

We use recall@k as the performance metric and compare the following methods in Table 2.7:

**1. En-Image CorrNet**: This is the CorrNet model trained using only $\mathcal{Z}_1$ as defined earlier in this section. The task is to retrieve English captions for a given image (or vice versa). This gives us an idea about the performance we could expect if direct parallel data is available between images and their captions in some language. We used the publicly available implementation of CorrNet provided by Chandar *et al.* (2016).

**2. Bridge CorrNet**: This is the Bridge CorrNet model trained using $\mathcal{Z}_1$ and $\mathcal{Z}_2$ as defined earlier in this section. The task is to retrieve French (or German) captions for a given image (or vice versa).

| Model | Captions | I To C Recall@5 | I To C Recall@10 | I To C Recall@50 | C To I Recall@5 | C To I Recall@10 | C To I Recall@50 |
|---|---|---|---|---|---|---|---|
| En-Image CorrNet | English | 0.118 | 0.190 | 0.456 | 0.091 | 0.168 | 0.532 |
| Bridge MAE | French | 0.008 | 0.017 | 0.069 | 0.007 | 0.013 | 0.063 |
| 2-CorrNet | French | 0.018 | 0.024 | 0.085 | 0.027 | 0.055 | 0.205 |
| Bridge CorrNet | French | 0.072 | 0.135 | 0.335 | 0.032 | 0.060 | 0.235 |
| CorrNet+MT | French | 0.101 | 0.163 | 0.414 | 0.069 | 0.127 | 0.416 |
| Bridge MAE | German | 0.005 | 0.009 | 0.053 | 0.006 | 0.013 | 0.058 |
| 2-CorrNet | German | 0.009 | 0.013 | 0.071 | 0.012 | 0.023 | 0.098 |
| Bridge CorrNet | German | 0.063 | 0.105 | 0.298 | 0.027 | 0.049 | 0.183 |
| CorrNet+MT | German | 0.084 | 0.163 | 0.420 | 0.061 | 0.107 | 0.343 |
| Random | | 0.006 | 0.009 | 0.044 | 0.005 | 0.009 | 0.050 |

Table 2.7: Performance of different models for image to caption (I to C) and caption to image (C to I) retrieval

**3. Bridge MAE**: The Multimodal Autoencoder (MAE) proposed by Ngiam *et al.* (2011) was the only competing model which was easily extendable to the bridge case. We train their model using $\mathcal{Z}_1$ and $\mathcal{Z}_2$ to minimize a suitably modified objective function. We then use the representations learned to retrieve French (or German) captions for a given image (or vice versa).

**4. 2-CorrNet**: Here, we train two individual CorrNets using $\mathcal{Z}_1$ and $\mathcal{Z}_2$ respectively. For the task of retrieving images given a French (or German) caption we first find its nearest English caption using the Fr-En (or De-En) CorrNet. We then use this English caption to retrieve images using the En-Image CorrNet. Similarly, for retrieving captions given an image we use the En-Image CorrNet followed by the En-Fr (or En-De) CorrNet.

**5. CorrNet + MT**: Here, we train an En-Image CorrNet using $\mathcal{Z}_1$ and an Fr/De-En MT system[4] using $\mathcal{Z}_2$. For the task of retrieving images given a French (or German) caption we translate the caption to English using the MT system. We then use this English caption to retrieve images using the En-Image CorrNet. For retrieving captions given images, we first translate all the 5000 French (or Germam) captions to English. We then embed these English translations (documents) and images (queries) in the common space computed using Image-En CorrNet and do a retrieval as explained earlier.

**6. Random**: A random image is returned for the given caption (and vice versa).

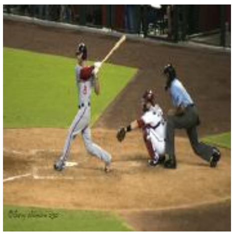
From Table 2.7, we observe that `CorrNet + MT` is a very strong competitor and gives the best results. The main reason for this is that over the years MT has matured enough for language pairs such as Fr-En and De-En and it can generate almost perfect

[4] `http://www.statmt.org/moses/`

1. Zwei Pferde stehen auf einem sandigen Strand nahe dem Ocean.
(Two horses standing on a sandy beach near the ocean.)
2. grasende Pferde auf einer trockenen Weide bei einem Flughafen.
(Horses grazing in a dry pasture by an airport.)
3. ein Elefant , Wasser aufseinen Rückend sprühend , in einem staubigen Bereich neben einem Baum.
(A elephant spraying water on its back in a dirt area next to tree .)
4. ein braunes pferd ißt hohes gras neben einem behälter mit wasser.
(Brown horses eating tall grass beside a body of water .)
5. vier Pferde grasen auf ein Feld mit braunem gras.
(Four horses are grazing through a field of brown grass.)

1. Ein Teller mit Essen wie Sandwich , Chips , Suppe und einer Gurke.
(Plate of food including a sandwich , chips , soup and a pickle.)
2. Teller , gefüllt mit sortierten Früchten und Gemüse und einigem Fleisch.
(Plates filled with assorted fruits and veggies and some meat.)
3. Ein Tisch mit einer Schüssel Salat und einem Teller Pizza.
(a Table with a bowl of salad and plate with a cooked pizza .)
4. Ein Teller mit Essen besteht aus Brokkoli und Rindfleisch.
(A plate of food consists of broccoli and beef.)
5. Eine Platte mit Fleisch und grünem Gemüse gemixt mit Sauce.
(A plate with meat and green veggies mixed with sauce.)

1. un bus de la conduite en ville dans une rue entourée par de grands immeubles.
(A city bus driving down a street surrounded by tall buildings.)
2. un bus de conduire dans une rue dans une ville avec des bâtiments de grande hauteur.
(A bus driving down a street in a city with very tall buildings.)
3. bus de conduire dans une rue de ville surpeuplée.
(Double - decker bus driving down a crowded city street.)
4. le bus conduit à travers la ville sur une rue animée.
(The bus drives through the city on a busy street.)
5. un grand bus coloré est arrêté dans une rue de la ville.
(A big , colorful bus is stopped on a city street.)



1. Un homme portant une batte de baseball à deux mains lors d'un jeu de balle professionnel.
(A man holding a baseball bat with two hands at a professional ball game.)
2. un joueur de tennis balance une raquette à une balle.
(A tennis player swinging a racket t a ball.)
3. un garçon qui est de frapper une balle avec une batte de baseball.
(A boy that is hitting a ball with a baseball bat.)
4. une équipe de joueurs de baseball jouant un jeu de base-ball.
(A team of baseball players playing a game of baseball.)
5. un garçon se prépare à frapper une balle de tennis avec une raquette.
(A boy prepares to hit a tennis ball with a racquet.)

Table 2.8: Images and their top-5 nearest captions based on representations learned using Bridge CorrNet. First two examples show German captions and the last two examples show French captions. English translations are given in parenthesis.

translations for short sentences (such as captions). In fact, the results for this method are almost comparable to what we could have hoped for if we had direct parallel data between Fr-Images and De-Images (as approximated by the first row in the table which reports cross-modal retrieval results between En-Images using direct parallel data between them for training). However, we would like to argue that learning a joint embedding for multiple views instead of having multiple pairwise systems is a more elegant solution and definitely merits further attention. Further, a "translation system" may not be available when we are dealing with modalities other than text (for example, there are no audio-to-video translation systems). In such cases, BridgeCorrNet could still be employed. In this context, the performance of BridgeCorrNet is definitely promising and

| | |
|---|---|
| Speisen und Getränke auf einem Tisch mit einer Frau essen im Hintergrund. (Food and beverages set on a table with a woman eating in the background .) |  |
| ein Foto von einem Laptop auf einem Bett mit einem Fernseher im Hintergrund. (A photo of a laptop on a bed with a tv in the background .) |  |
| un homme debout à côté de aa groupe de vaches. (A man standing next to a group of cows.) |  |
| personnes portant du matériel de ski en se tenant debout dans la neige. (People wearing ski equipment while standing in snow.) |  |

Table 2.9: French and German queries and their top-5 nearest images based on representations learned using Bridge CorrNet. First two queries are in German and the last two queries are French. English translations are given in parenthesis.

shows that a model which jointly learns representations for multiple views can perform better than methods which learn pair-wise common representations (2-CorrNet).

### 2.6.1 Qualitative Analysis

To get a qualitative feel for our model's performance, refer Table 2.8 and 2.9. The first row in Table 2.8 shows an image and its top-5 nearest German captions (based on Euclidean distance between their common representations). As per our parallel image caption test set, only the second and fourth caption actually correspond to this image. However, we observe that the first and fifth caption are also semantically very related to the image. Both these captions talk about horses, grass or water body (ocean), *etc.* Similarly the last row in Table 2.8 shows an image and its top-5 nearest French captions. None of these captions actually correspond to the image as per our parallel image caption test set. However, clearly the first, third and fourth caption are semantically very relevant to this image. Even the remaining two captions capture related concepts. We can make a similar observation from Table 2.9 where most of the top-5 retrieved images do not correspond to the French/German caption but they are semantically very similar. It is indeed impressive that the model is able to capture such cross modal semantics between images and French/German even without any direct parallel data between them.

# Chapter 3

# ADAAPT: A Deep Architecture for Adaptive Policy Transfer from Multiple Sources

## 3.1 Introduction

One of the goals of Artificial Intelligence (AI) is to build autonomous agents that can learn and adapt to new environments. Reinforcement Learning (RL) is a key technique for achieving such adaptability. RL looks at the problem of intelligent decision making as one of stochastic sequential control Sutton and Barto (1998). The goal of RL algorithms is to learn an optimal policy for choosing actions that maximises some notion of long term performance. One of the chief drawbacks of RL is that learning on new tasks from the scratch takes a long time since the agent initially performs random exploration to discover details of the task. Much of the research in RL has been focused on cutting down this initial exploration. One of the key idea used is that of transfer learning.

Transfer is by no means limited to RL. The notion of transfer is to use knowledge gained from solving related instances of a problem (source tasks) to solve a new instance (target task) better - either in terms of speeding up the learning process or in terms of achieving a better solution, among other performance measures. When applied to RL, transfer could be accomplished in many ways (see Taylor and Stone (2009, 2011) for a very good survey of the field). One could think of transferring the value function from a source task and use that as the initial estimate of the value function in the target task to cut down on the initial exploration Sorg and Singh (2009). Another method to achieve transfer is to reuse policies derived in the source task(s) in the target task. This can take one of two forms - (i) the derived policies can be used as initial explorative trajectories Atkeson and Schaal (1997); Niekum *et al.* (2013) in the target task, thereby cutting down on random exploration; and (ii) the derived policy could be used to define *macro* actions which may then be used by the agent in solving the target task Mannor *et al.* (2004); Brunskill and Li (2014). Also, the knowledge of the domain's model parameters from the source tasks to the target task can be transferred. Such an approach assumes that the model of the target task is sufficiently close to the source task and prior knowledge of the model allows the agent to eliminate frivolous exploration.

While transfer in RL has been much explored, there are two crucial issues that have not received much attention. The first is *negative transfer* - when transfer from a source task degrades the performance of the agent on the target task. This is widely recognised as a serious problem in the transfer learning literature. In the context of RL too this severely limits the applicability of transfer to cases when some measure of relatedness between source and target tasks can be guaranteed. One work that explicitly addresses the question of negative transfer is that of Brunskill and Li (2014) where they assume that they have access to source tasks to sufficiently cover the space of problems from which the target task is drawn. Further the safe exploration that they use as part of the learning process ensures no negative transfer happens. In our work, we take a more general approach. We maintain a copy of the policy that is learned from scratch on the target task. If there is evidence of negative transfer happening, we will fall back to this base policy.

The second problem with transfer is that of identifying an appropriate source task to transfer from. This is especially problematic, if we are trying to transfer whole solutions or value functions to the target task. One way of mitigating this is to learn *macro* actions and transfer policy fragments - the learning agent decides if policy fragments are appropriate in the target task. Another way of approaching this, is to select different and multiple source tasks to transfer from at different points in the target task. We call this *selective transfer*. In fact ours is the first work that explicitly looks at blending policies from different source tasks for transfer to a single target task and for different parts of its state space. Earlier, multiple task transfer settings have either formulated it as a multi-task learning problem, or one of selecting a specific source task for a given target task. In our framework the agent can pick and choose portions of policies from different and multiple source tasks while solving a single target task. This allows us to treat all the prior policies as a partial basis from which the target policy is created.

In this work we propose ADAAPT, A Deep Architecture for Adaptive Policy Transfer, a transfer learning framework that avoids negative transfer while performing selective transfer from multiple source tasks. One key difficulty in selective transfer is that learning a selection function that blends the different policies together is a very challenging problem. One of the distinguishing features of our approach is the use of a deep neural network that leverages ideas from recent work on learning attention Bahdanau *et al.* (2014) to learn complex selection functions without worrying about representation issues a priori. Deep neural architecture also enables ADAAPT's deployment in large domains. Since our approach needs an explicit representation of the policies of the source tasks, we present the approach using specific choices for the reinforcement learning architecture (REINFORCE, Actor-Critic Williams (1992); Konda and Tsitsiklis (2000)), for ease of exposition. The ideas presented here extend to other architectures as well.

The main features of ADAAPT are:

1. It avoids negative transfer as is empirically demonstrated by transferring from carefully constructed *bad* initial policies.
2. It achieves selective policy transfer from multiple source tasks to a target task.
3. It uses a deep neural network architecture that enables the learning of the selective transfer in a natural way, and enables the deployment of the architecture in large domains.

## 3.2 Related Work

As mentioned earlier, transfer learning approaches could deal with transferring representations, policies or value functions. For example, Banerjee and Stone (2007) describe a method for transferring value functions by constructing a *Game tree*. Similarly, Sorg and Singh (2009) explores the idea of transferring the value function from a source task and use that as the initial estimate of the value function in the target task to cut down on the initial exploration. Another method to achieve transfer is to reuse policies derived in the source task(s) in the target task. Probabilistic Policy Reuse as discussed in Fernández and Veloso (2006) provides a useful way for transferring policies. This method maintains a library of policies and selects a policy based on a similarity metric, or a random policy, or a max-policy from the knowledge obtained. The policy selection happens a priori to each episode.

Atkeson and Schaal (1997); Niekum *et al.* (2013) propose a method to use the learned source policies as initial explorative trajectories in the target task instead of relying solely on random exploration. Lazaric and Restelli (2011) addresses the issue of negative transfer in transferring samples for a related task in a multi-task setting. Representation transfer is done using Proto Value Functions as discussed in Ferguson and Mahadevan (2006). Konidaris *et al.* (2012) discusses the idea of exploiting shared common features across related tasks. They learn a *shaping function* that can be used in later tasks. Also, Cesa-Bianchi *et al.* (2006) proposes a bandit algorithm that minimizes the regret, in a partial monitoring setting. This setting though is remotely connected to ours, the feedback that the network receives is similar to the expression the bandit setting receives in Cesa-Bianchi *et al.* (2006). The work Talvitie and Singh (2007) tries to find the promising policy from a set of candidate policies that are generated using different action mapping to a single solved task. On the other hand we make use of more than one source tasks to selectively transfer policies at the granularity of state.

In contrast to previous work, our work explicitly focuses on the ability to *selectively transfer*, using multiple source tasks while avoiding *negative transfer*. We define these two challenges in the next section and then propose a model to address them.

## 3.3 Challenges of Transfer Learning addressed in this work

#### Negative Transfer

Consider a performance measure $\rho$, as discussed in Taylor and Stone (2009), where $\rho$ could be, for example, jump start (the initial performance of the agent in the target task) or time to threshold (time to reach a predefined performance level). Let $\pi_{WT}$ be a policy learnt from scratch in the target without any transfer and $\pi_T$ be a policy learnt in the target using transfer from source tasks. If following $\pi_T$ gives a performance, measured by $\rho$, worse than that got through following $\pi_{WT}$, then we say it is a *negative transfer*.

#### Selective Transfer

Let there be $N$ policies, $\pi_1, \pi_2, \ldots, \pi_N$. When the agent learns to solve a new target task, it should be able to learn policies of the form, $\pi(s) = f(\pi_1(s), \ldots, \pi_N(s)) \; \forall \; s \in S$, the set of all states in an MDP.

## 3.4 Proposed Model

We propose a model for policy transfer from multiple source MDPs with the same structure, and different model parameters. Let there be N policies, $\pi_1, \pi_2, \ldots, \pi_N$ derived from solving $N$ prior tasks. When the agent learns to solve a new target task, the agent learns policies of the form $\pi(s) = f(\pi_1(s), \ldots, \pi_N(s), \pi_R(s))$ where $s \in S$ represents the state and $\pi_R$ is a policy learnt from scratch on the target task. The policies used are stochastic. In this work, we assume that $f$ is implemented as a convex combination of the policies (action probabilities) and is given by $f(\pi_1(s), \ldots, \pi_N(s), \pi_R(s)) = \sum_{i=1}^{N} w_{i,s}\pi_i(s) + w_{N+1,s}\pi_R(s)$, where $\sum_{i=1}^{N+1} w_{i,s} = 1$ and $w_{i,s} \in [0, 1]$. $\pi$ is the policy that the agent follows. Figure 3.1 shows the architecture diagram of the proposed model. The key component of the model is the central network which learns the weights ($w_{i,s}$, $i \in 1, 2, \ldots, N+1$) to be assigned to the different policies. We refer to this network as the attention network. The weights allow the network to selectively accept or reject the policies of other source tasks depending on the input state. This ability allows the model to achieve both its stated goals, *viz.*, (i) avoid negative transfer from policy $\pi_i(s)$ by setting $w_{i,s}$ to a very low value and (ii) selectively transfer the knowledge from the source tasks for certain states by setting weights of those tasks to high values for those states.

Depending on the feedback obtained from the environment upon following $\pi$, the attention network's parameters are updated to improve performance. Even though the agent follows the policy $\pi$, we update the parameters of the network that produces $\pi_R$, the randomly initialised policy network, as if the action taken by the agent was based only on $\pi_R$. The networks which produce the polices of the source tasks, $\pi_1, \ldots, \pi_N$ remain fixed.

Alternately, we could also update the parameters of the networks that produce $\pi_1, \ldots, \pi_N$. However, doing so has two major drawbacks. First, if we update the parameters of all the source networks, there could be a significant amount of unlearning in the source networks before the attention network identifies the utility of the source tasks for the target task. This could result in a weaker transfer than actually possible. Secondly, since the number of parameters of the model would increase linearly with the number of source tasks, it could lead to problems when we have a large number of source tasks. We verified this empirically and hence for all experiments reported we

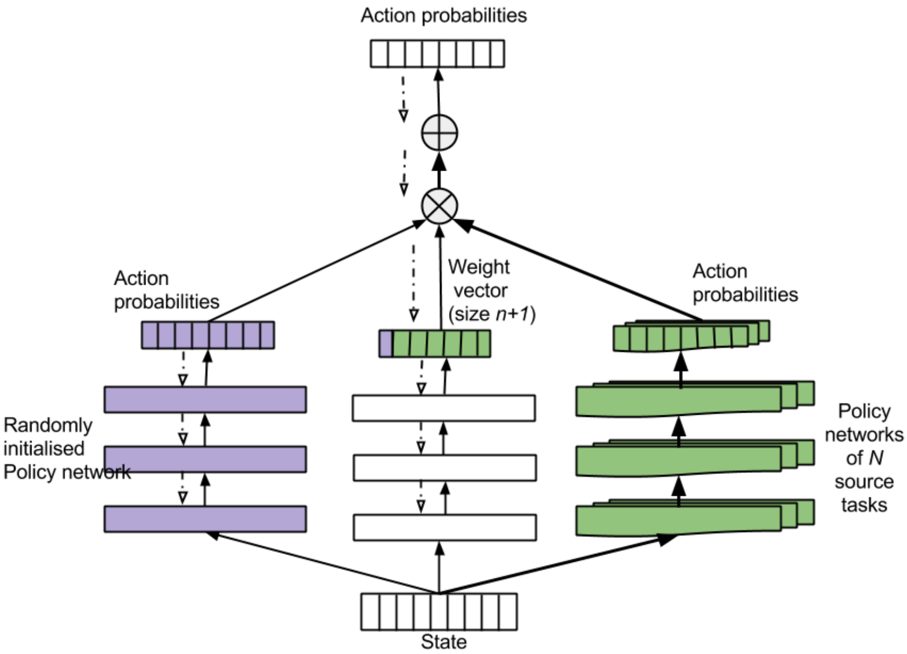



Figure 3.1: ADAAPT. The doted arrows represent the path of back propagation.

only update the parameters of the network that produces $\pi_R$ and the attention network which produces $w_i$.

If there is a source task whose policy $\pi_j$ is useful for the target task in some parts of its state space, then over time, $\pi_R$ would start replicating that source task policy $\pi_j$ in those parts of the state spaces. Note that the agent could follow $\pi_j$ even before $\pi_R$ attains its replication in the corresponding parts of the state space. Since the attention is soft, our model has the flexibility to combine multiple source task policies. After the learning is done, $\pi_R$ alone can be used as the policy of the target task for future endeavors. When the attention network's weight for the policy $\pi_R$ is high, the mixture policy is dominated by $\pi_R$, and the behavior is nearly on-policy. In the other case, $\pi_R$ undergoes off-policy learning. Empirically, we observe that $\pi_R$ converges.

Following the recent success of Deep Neural Networks in a variety of Machine Learning tasks we made a design choice to use deep neural networks in our model (and hence the name ADAAPT). This should potentially allow the model to work even for large, complex Reinforcement Learning problems. Using deep neural networks that leverages ideas from recent work on learning attention allows the agent to learn complex selection functions, without worrying about representation issues a priori.

### 3.4.1 Instantiations of ADAAPT

ADAAPT is a generic framework that can be used alongside any algorithm that has an explicit representation of the policy. Here we describe two instantiations of ADAAPT, one for direct policy search using REINFORCE algorithm and another in the Actor-Critic setup.

#### ADAAPTive REINFORCE

REINFORCE algorithms Williams (1992) can be used for direct policy search by making weight adjustments in a direction that lies along the gradient of the expected reinforcement. In ADAAPTive REINFORCE, ADAAPT is used directly to do policy search, and its parameters are updated using REINFORCE. Let $\psi$ represent the attention network. $\psi$ outputs $w$ and is parameterised by $u$. The update of $u$ is given by,

$$\Delta u = \alpha_u (r - b) \frac{\partial \sum_{t=1}^{L} \log(\pi(s_t, a_t))}{\partial u} \tag{3.1}$$

$$u \leftarrow u + \Delta u \tag{3.2}$$

where, $\alpha_u$ is a non-negative factor, $r$ is the current reinforcement, which is the return at the end of an episode in our case, $b$ is the reinforcement baseline, $L$ is the total number of steps in the episode, $a_t$ is the action taken at step $t$ from state $s$ following $\pi$ and $\pi(s, a_t)$ is the probability of taking action $a_t$ in state $s$ as given by $\pi$. Note that $\pi$ is the policy the agent follows.

Let $\phi$ represent the randomly initialised policy network, which is learnt from scratch for the target task. $\phi$ outputs $\pi_R$, and is parameterised by $v$. The update for $v$ is given by,

$$\Delta v = \alpha_v (r - b) \frac{\partial \sum_{t=1}^{L} \log(\pi_R(s_t, a_t))}{\partial v} \tag{3.3}$$

$$v \leftarrow v + \Delta v \tag{3.4}$$

where, $\alpha_v$ is a non-negative factor, $a_t$ is the action taken by the agent at step $t$ from state $s$ following $\pi$, and $\pi_R(s, a_t)$ is the probability of action $a_t$ given by $\pi_R$. As mentioned earlier, though the agent follows $\pi$, the parameters of the network representing $\pi_R$ are updated as if the action was taken by following $\pi_R$.

### ADAAPTive Actor-Critic

Actor-Critic methods Konda and Tsitsiklis (2000) are Temporal Difference (TD) methods that have two separate components, *viz.*, an *actor* and a *critic*. The actor proposes a policy whereas the critic estimates the value function of the policy and criticizes the policy of the actor. The updates to the actor happens through *TD-error* which is the one step estimation error that helps in reinforcing an agent's behaviour. This *TD-error* is a scalar value and is referred to as the critic.

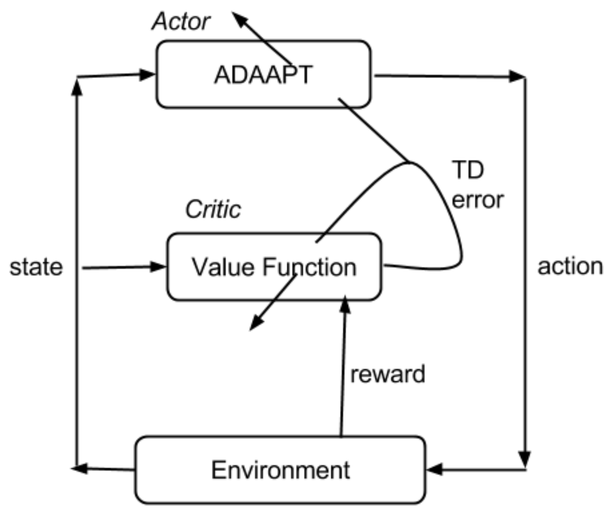



Figure 3.2: ADAAPTive Actor-Critic

ADAAPTive Actor-Critic is an Actor-Critic model, where the actor uses ADAAPT. We have a deep adaptive actor which utilises the knowledge of the learned source task while avoiding negative transfer and performing selective transfer wherever appropriate. The critic here learns the state values from scratch. In other words, the actor is aware of all the previous learnt tasks and tries to use that knowledge for its benefit. The critic evaluates this selection as well as the action solely depending on the target task (*i.e.*, it does not care about the source tasks).

Let $s_t$ be the state the agent is at time step $t$. $s_{t+1}$ is the state the agent reaches with a reward of $r_{t+1}$ upon taking action $a_t$ at time step $t$ from state $s$ following the policy $\pi$. Let $V(s)$ represent the value of state $s$. Then, the update equations for parameters $u$ of the attention network $\psi$ and the parameters $v$ of the randomly initialised policy's ($\pi_R$'s) network $\phi$ are as follows.

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{3.5}$$

where, $\gamma$ is the discount factor.

$$\Delta u = \alpha_u \delta_t \frac{\frac{\partial \log \pi(s_t, a_t)}{\partial u}}{\left|\frac{\partial \log \pi(s_t, a_t)}{\partial u}\right|} \tag{3.6}$$

$$u \leftarrow u + \Delta u \tag{3.7}$$

where, $\alpha_u$ is a non-negative factor and $\pi(s_t, a_t)$ is the probability of taking action $a_t$ from state $s_t$ given by the policy $\pi$.

$$\Delta v = \alpha_v \delta_t \frac{\frac{\partial \log \pi_R(s_t, a_t)}{\partial v}}{\left|\frac{\partial \log \pi_R(s_t, a_t)}{\partial v}\right|} \tag{3.8}$$

$$v \leftarrow v + \Delta v \tag{3.9}$$

where, $\alpha_v$ is a non-negative factor, $\pi_R(s_t, a_t)$ is the probability of taking action $a_t$ from state $s_t$ given by the policy $\pi_R$.

## 3.5 Experiments and Discussion

We evaluate the performance of ADAAPT using two simulated worlds, *viz.*, chain world and puddle world as described below. One main motivation of these experiments is to test the consistency of results with the algorithm motivation. While the presence of a deep architecture, allows for the usage of the algorithm for high dimensional, complex domains and large number of domains, the goals of these experiments are not to evaluate them.

**Chain world:** Figure 3.3 shows the chain world where the goal of the agent is to go from one point in the chain (starting state) to another point (goal state) in the least number of steps. At each state the agent can choose to either move one position to the left or to the right. After reaching the goal state the agent gets a reward that is inversely proportional to the number of steps taken to reach the goal.

**Puddle worlds:** Figure 3.4 shows the discrete version of the standard puddle world that is widely used in Reinforcement Learning literature. In this world, the goal of the agent is to go from a specified start position to the goal position, maximising its return. At each state the agent can choose one of these four actions: move one position to the north, south, east or west. With $0.9$ probability the agent moves in the chosen direction and with $0.1$ probability it moves in a random direction irrespective of its choice of action. On reaching the goal state, the agent gets a reward of $+10$. On reaching other parts of the grid the agent gets different penalties as mentioned in the legend of Figure 3.4. Figures 3.5 to 3.8 show different variants of the puddle world which we constructed to evaluate different features of ADAAPT as described below.

### 3.5.1 Experiment 1: Ability to avoid negative transfer

We first consider the case when only one learned source task is available such that it can hamper the learning process of the new target task. We refer to such a source task as an unfavorable source task. In such a scenario, the attention network shown in Figure 3.1 should learn to assign a very low weight to the action probabilities output by the policy network of this unfavorable source task. We now define an experiment using the puddle world from Figure 3.4 to show that ADAAPT indeed does so. The target task in our experiment is to maximize the return in reaching the goal state $G1$ starting from any one of the states $S1, S2, S3, S4$. We artificially construct an unfavorable source task by first learning to solve the above task and then negating the weights of the topmost layer of the actor network. Given such an unfavorable task, Figure 3.9 compares the performance of the following methods:

- `R`: In this case, there is no learned source task and the new task simply starts with a randomly initialized actor network and learns the weights of this network over time/episodes from scratch.
- `B`: In this case, the new task simply starts with the actor network learned for the unfavorable task and adjusts the weights of this network over time/episodes.
- `ADAAPT_RB`: In this case, the actor uses ADAAPT. Specifically, it is provided a randomly initialized policy network as well as the policy network of the unfavorable task.

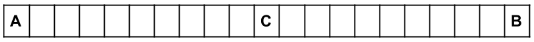



Figure 3.3: Chain World

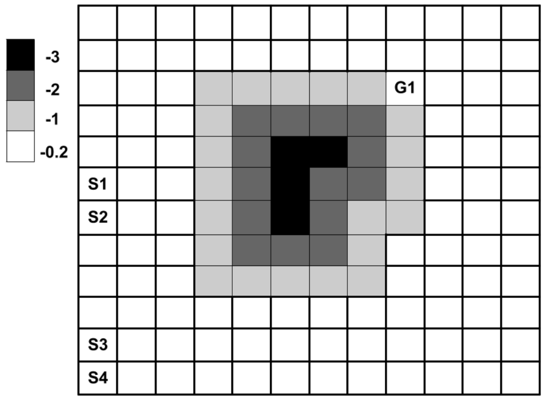



Figure 3.4: Puddle World 1

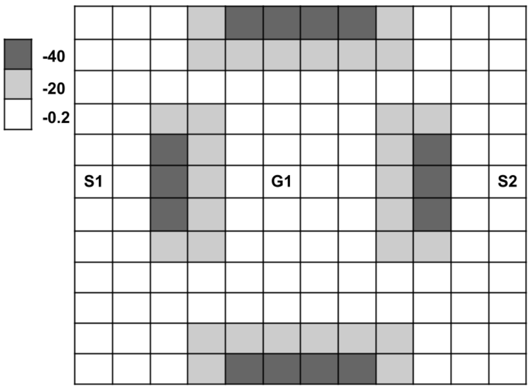



Figure 3.5: Puddle World 2

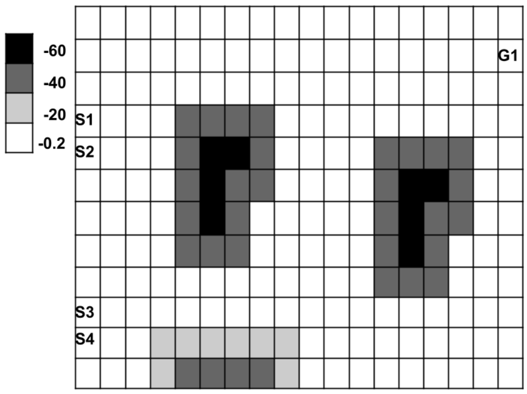



Figure 3.6: Puddle World 3

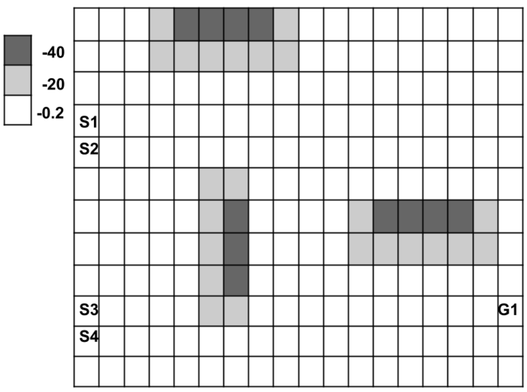



Figure 3.7: Puddle World 4

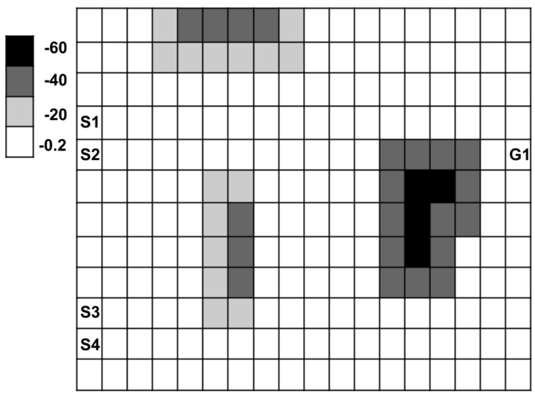



Figure 3.8: Puddle World 5

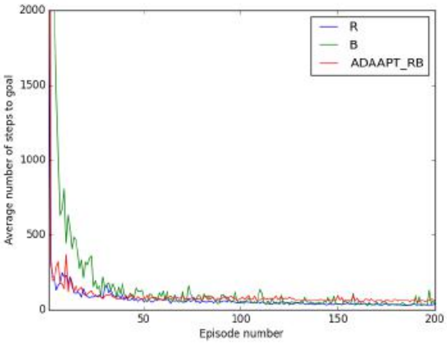



Figure 3.9: Avoiding negative transfer

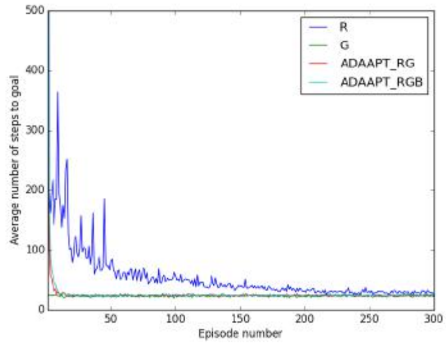



Figure 3.10: Transferring from a favorable task

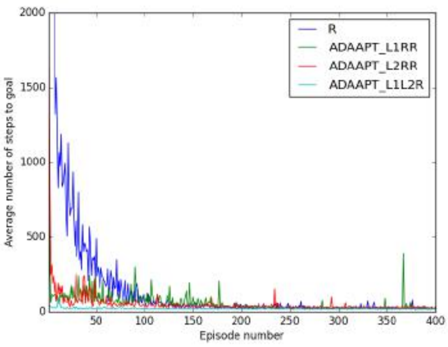



Figure 3.11: Transferring from multiple similar source tasks

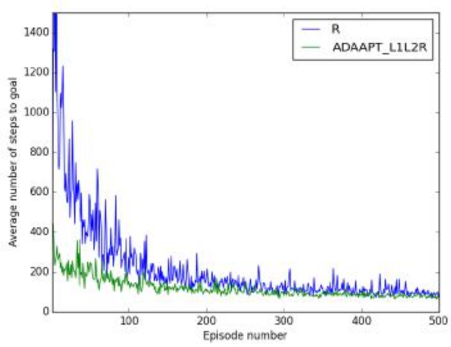



Figure 3.12: Transferring from multiple different source tasks

As is evident from Figure 3.9, ADAAPT does not get hampered by the unfavorable source task. It learns to ignore the unfavorable task and does as good as the case when such an unfavorable source task is not available and only a randomly initialized network is available (R).

### 3.5.2 Experiment 2: Ability to transfer from a favorable source task

Now, consider the case when a favorable source task is available that can help the learning process of the target task. In such a scenario, the attention network shown in Figure 3.1 should learn to assign high weights to the action probabilities output by the policy network of this favorable source task. To show that ADAAPT indeed does so, we use the same target task as used in Experiment 1. We artificially construct a favorable source task simply by learning to solve the target task and using the learned actor network. Figure 3.10 compares the following methods:

- `R`: This is same as described in Experiment 1.
- `G`: Here, the target task simply starts with the actor network learned for the favorable task and adjusts the weights of this network over time/episodes if needed.
- `ADAAPT_RG`: Here, the actor uses ADAAPT. Specifically, it is provided a randomly initialized policy network as well as the policy network of the favorable task.
- `ADAAPT_RGB`: Here again, ADAAPT is used but in addition to the randomly initialized policy network and the trained policy network of a favorable task, the trained policy network of an unfavorable source task is also available.

As is evident from Figure 3.10, when a favorable source task is available, ADAAPT is able to exploit it and improve the learning speed of the new task. Further, ADAAPT is not affected by the presence of an unfavorable task (as the performance of `ADAAPT_RGB` and `ADAAPT_RG` are almost the same).

### 3.5.3 Experiment 3: Ability to selectively transfer from multiple source tasks

In this section, we consider the case when multiple partially favorable source tasks are available such that each of them can assist the learning process for different parts of the state space of the target task. We illustrate this first using the simple chain world shown in (Fig. 3.3). Consider that the target task $LT$ is to start in $A$ or $B$ with uniform probability and reach $C$ in the least number of steps. Now, consider that two learned source tasks, *viz.*, $L1$ and $L2$, are available. $L1$ is the source task where the agent has learned to reach the left end ($A$) starting from the right end ($B$). In contrast, $L2$ is the source task where the agent has learned to reach the right end ($B$) starting from the left end ($A$). Intuitively, it should be clear that the target task should benefit from the policies learnt for tasks $L1$ and $L2$. We learn the task $LT$ using ADAAPTive REINFORCE with the following policies (i) policies learned for $L1$ (i) policies learned for $L2$, and (iii) a randomly initialized policy network. Figure 3.13 shows the weights given by the attention network to the different source policies for different parts of the state space at the end of learning. We observe that the attention network has learned to ignore $L1$, and $L2$ for the left, and right half of the state space of the target task, respectively. As the randomly initialised actor network becomes the good policy over time, it has a high weight throughout the state space of the target task.

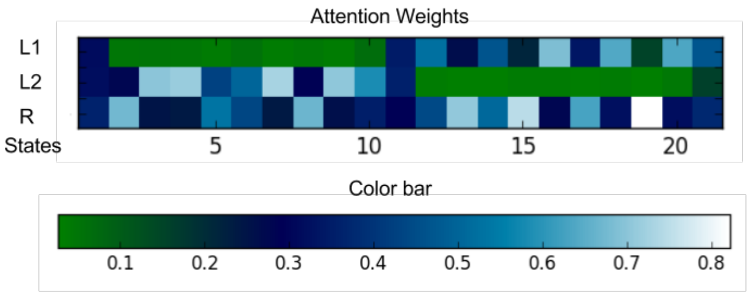



Figure 3.13: The weights (in the range of $[0, 1]$) given by the attention network to different policy networks for different parts of the state space at the end of learning.

We repeat the same experiment in a relatively more complex puddle world shown in Figure 3.5. In this case, $L1$ is the task of moving from $S1$ to $G1$, and $L2$ is the task of moving from $S2$ to $G1$. In the target task $LT$, the agent has to learn to move to $G1$ starting from either $S1$ or $S2$ chosen with uniform probability. We learn the task $LT$ using ADAAPTive Actor-Critic method, where the following are available (i) learned policy networks for $L1$ (ii) learned policy network for $L2$ and (iii) a randomly initialized policy network. Figure 3.11 compares the performance of the following methods.

- `R`: This is same as described in Experiment 1.
- `ADAAPT_L1RR`: In this case, ADAAPT is provided two randomly initialized policy networks as well as the learned actor network of $L1$.
- `ADAAPT_L2RR`: In this case, ADAAPT is provided two randomly initialized policy networks as well as the learned actor network of $L2$.
- `ADAAPT_L1L2R`: In this case, ADAAPT is provided one randomly initialized policy network as well as the learned actor networks of both $L1$ and $L2$.

We use two random networks in `ADAAPT_L1RR` `ADAAPT_L2RR` so that the number of parameters in this setup are comparable to `ADAAPT_L1L2R`. We observe that `ADAAPT_L1L2R` is able to perform better than the other configurations. It is able to exploit the policies learned for $L1$, and $L2$ and performs better than $R$.

Finally, we move to an even more challenging task involving three variants of the puddle world. Specifically, $L1$ is the task shown in Figure 3.6, $L2$ is the task shown in Figure 3.7 and $LT$ is the task shown in Figure 3.8. In all these worlds, the agent can start from either $S1$, $S2$, $S3$ or $S4$ with uniform probability and has to reach the goal state $G1$. The position and shape of the puddles as well as the position of the goal state $G1$ are different in each of the three worlds. Figure 3.12 compares the following methods:

- `R`: This is same as described in Experiment 1.
- `ADAAPT_L1L2R`: In this case, ADAAPT is provided one randomly initialized actor network as well as the learned actor networks of both $L1$ and $L2$.

Despite clear differences between the source tasks and the target task, ADAAPT does meaningful transfer.

# Chapter 4

# CONCLUSION AND FUTURE WORK

We have proposed Bridge Correlational Neural Networks which can learn common representations for multiple views even when parallel data is available only between these views and a pivot view. Our method performs better than the existing state of the art approaches on the cross language classification task and gives very promising results on the cross modal access task. We also release a new multilingual image caption benchmark (MIC benchmark) which will help in further research in this field[1].

We then presented a deep neural network architecture for transfer learning that avoids negative transfer while enabling selective transfer from multiple source tasks. We empirically evaluate the performance of the proposed model using a variety of simulated worlds and show that it indeed achieves its stated goals. While in this work we focused on transfer between tasks that share the same state and action spaces, the use of deep networks opens up the possibility of going beyond this setting. For example, a deep neural network can be used to learn common representations Wernsdorfer and Schmid (2014) for multiple tasks thereby enabling transfer between related tasks that could possibly have different state and action parameterisation. Further, the use of deep networks provide a straightforward way of applying these ideas in a continuous domain. Over all we believe that ADAAPT is a novel way to approach transfer learning that opens up many new avenues of research in this area.

In future, we would like to work on the problem of video completion using the textual description of image, where the goal is to complete a video, given the textual description of the image that the video should reach to. The model would combine recurrent network based video completion architectures and Bridge Correlational Neural Network based common representation learning architectures.

Transfer Learning and Multimodal learning, we feel, when explored, could improve our understanding of intelligence and hence intelligence itself to new levels. We want to

[1]Details about the MIC benchmark and performance of various state-of-the-art models will be maintained at `http://sarathchandar.in/bridge-corrnet`

come up with a unified model which could do transfer learning and multimodal learning. As an outline, we feel, the model should be such that the encoding part of the model has different paths for different modalities, which allows different types of processing on different modalities or views. Once we get the higher level abstract knowledge from the various modalities, they should be effectively combined. During the process of using this knowledge and performing various tasks, the model should be able to use this combined knowledge as well as the knowledge of the previous learnt tasks from memory seamlessly. Various tasks, involving different types of outputs and signals should have different paths branching from the common path. Building such a model, which is effective, could be challenging, and involve solving several sub problems, like the ones that we have tried to solve in this work. It could have a lot of new applications and could also improve the performance of systems in many applications, for example in the areas of vision, NLP and robotics.

# Bibliography


1. **Akaho, S.** (2001). A kernel method for canonical correlation analysis. *In Proc. Int'l Meeting on Psychometric Society*.

2. **Andrew, G.**, **R. Arora**, **J. Bilmes**, and **K. Livescu** (2013). Deep canonical correlation analysis. *ICML*.

3. **Atkeson, C. G.** and **S. Schaal**, Robot learning from demonstration. *In In Proceedings of International Conference on Machine Learning*, volume 97. 1997.

4. **Bahdanau, D.**, **K. Cho**, and **Y. Bengio** (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

5. **Ballesteros, L.**, Cross language retrieval via transitive translation. *In W.B. Croft (Ed.), Advances in information retrieval: Recent research from the CIIR*. Boston: Kluwer Academic Publishers., 2000.

6. **Banerjee, B.** and **P. Stone**, General game learning using knowledge transfer. *In In The 20th International Joint Conference on Artificial Intelligence*. 2007.

7. **Brunskill, E.** and **L. Li**, Pac-inspired option discovery in lifelong reinforcement learning. *In Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014.

8. **Cesa-Bianchi, N.**, **G. Lugosi**, and **G. Stoltz** (2006). Regret minimization under partial monitoring. *Mathematics of Operations Research*, **31**(3), 562–580.

9. **Cettolo, M.**, **C. Girardi**, and **M. Federico**, Wit$^3$: Web inventory of transcribed and translated talks. *In Proceedings of the 16$^{th}$ Conference of the European Association for Machine Translation (EAMT)*. Trento, Italy, 2012.

10. **Chandar, S.**, **M. M. Khapra**, **H. Larochelle**, and **B. Ravindran** (2016). Correlational neural networks. *Neural Computation*, **28**(2), 257 – 285.

11. **Chandar, S.**, **M. M. Khapra**, **B. Ravindran**, **V. C. Raykar**, and **A. Saha** (2013). Multilingual deep learning. *NIPS Deep Learning Workshop*.

12. **Chandar, S.**, **S. Lauly**, **H. Larochelle**, **M. M. Khapra**, **B. Ravindran**, **V. C. Raykar**, and **A. Saha**, An autoencoder approach to learning bilingual word representations. *In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2014. URL `http://papers.nips.cc/paper/5270-an-autoencoder-approach-to-learning-bilingual-word-representat`

13. **Cohn, T.** and **M. Lapata**, Machine translation by triangulation: Making effective use of multi-parallel corpora. *In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic, 2007.

14. **Cruz-Cano, R.** and **M.-L. T. Lee** (2014). Fast regularized canonical correlation analysis. *Computational Statistics & Data Analysis*, **70**, 88 – 100.

15. **Ferguson, K.** and **S. Mahadevan** (2006). Proto-transfer learning in markov decision processes using spectral methods. *Computer Science Department Faculty Publication Series*, 151.

16. **Fernández, F.** and **M. Veloso**, Probabilistic policy reuse in a reinforcement learning agent. *In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 2006.

17. **Gouws, S.**, **Y. Bengio**, and **G. Corrado**, Bilbowa: Fast bilingual distributed representations without word alignments. *In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 2015. URL `http://jmlr.org/proceedings/papers/v37/gouws15.html`.

18. **Hermann, K. M.** and **P. Blunsom**, Multilingual Distributed Representations without Word Alignment. *In Proceedings of International Conference on Learning Representations (ICLR)*. 2014*a*.

19. **Hermann, K. M.** and **P. Blunsom**, Multilingual models for compositional distributed semantics. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. 2014*b*.

20. **Hotelling, H.** (1936). Relations between two sets of variates. *Biometrika*, **28**, 321 – 377.

21. **Hsieh, W.** (2000). Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, **13**(10), 1095 – 1105.

22. **Jia, Y.**, **E. Shelhamer**, **J. Donahue**, **S. Karayev**, **J. Long**, **R. Girshick**, **S. Guadarrama**, and **T. Darrell** (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

23. **Khapra, M. M.**, **A. Kumaran**, and **P. Bhattacharyya**, Everybody loves a rich cousin: An empirical study of transliteration through bridge languages. *In Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*. 2010*a*. URL `http://www.aclweb.org/anthology/N10-1065`.

24. **Khapra, M. M.**, **R. Udupa**, **A. Kumaran**, and **P. Bhattacharyya**, PR + RQ ALMOST EQUAL TO PQ: transliteration mining using bridge language. *In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. 2010*b*. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1937`.

25. **Klementiev, A.**, **I. Titov**, and **B. Bhattarai**, Inducing Crosslingual Distributed Representations of Words. *In Proceedings of the International Conference on Computational Linguistics (COLING)*. 2012.

26. **Konda, V.** and **J. Tsitsiklis**, Actor-critic algorithms. *In SIAM Journal on Control and Optimization*. MIT Press, 2000.

27. **Konidaris, G.**, **I. Scheidwasser**, and **A. G. Barto** (2012). Transfer in reinforcement learning via shared features. *The Journal of Machine Learning Research*, **13**(1), 1333–1371.

28. **Kumaran, A.**, **M. M. Khapra**, and **P. Bhattacharyya** (2010). Compositional machine transliteration. *ACM Trans. Asian Lang. Inf. Process.*, **9**(4), 13. URL `http://dx.doi.org/10.1145/1838751.1838752`.

29. **Lazaric, A.** and **M. Restelli**, Transfer from multiple mdps. *In Advances in Neural Information Processing Systems*. 2011.

30. **Lehtokangas, R.**, **H. Keskustalo**, and **K. Järvelin** (2008). Experiments with transitive dictionary translation and pseudo-relevance feedback using graded relevance assessments. *Journal of the American Society for Information Science and Technology*, **59**(3), 476–488.

31. **Luo, Y.**, **D. Tao**, **Y. Wen**, **K. Ramamohanarao**, and **C. Xu** (2015). Tensor canonical correlation analysis for multi-view dimension reduction. *In Arxiv*.

32. **Mannor, S.**, **I. Menache**, **A. Hoze**, and **U. Klein**, Dynamic abstraction in reinforcement learning via clustering. *In Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.

33. **Mikolov, T.**, **Q. Le**, and **I. Sutskever** (2013). Exploiting Similarities among Languages for Machine Translation. Technical report, arXiv.

34. **Nakov, P.** and **H. T. Ng**, Improved statistical machine translation for resource-poor languages using related resource-rich languages. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore, 2009.

35. **Ngiam, J.**, **A. Khosla**, **M. Kim**, **J. Nam**, **H. Lee**, and **N. Andrew** (2011). Multimodal deep learning. *ICML*.

36. **Niekum, S.**, **S. Chitta**, **A. G. Barto**, **B. Marthi**, and **S. Osentoski**, Incremental semantically grounded learning from demonstration. *In Robotics: Science and Systems*, volume 9. 2013.

37. **Nielsen, F. Å.**, **L. K. Hansen**, and **S. C. Strother** (1998). Canonical ridge analysis with ridge parameter optimization.

38. **Socher, R.**, **A. Karpathy**, **Q. V. Le**, **C. D. Manning**, and **A. Y. Ng** (2014). Grounded compositional semantics for finding and describing images with sentences. *TACL*, **2**, 207–218. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/325`.

39. **Sorg, J.** and **S. Singh**, Transfer via soft homomorphisms. *In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

40. **Soyer, H.**, **P. Stenetorp**, and **A. Aizawa**, Leveraging monolingual data for crosslingual compositional word representations. *In Proceedings of the 3rd International Conference on Learning Representations*. San Diego, California, USA, 2015.

41. **Sutton, R. S.** and **A. G. Barto**, *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998, 1st edition. ISBN 0262193981.

42. **Talvitie, E.** and **S. Singh**, An experts algorithm for transfer learning. *In Proceedings of the 20th international joint conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2007.

43. **Taylor, M. E.** and **P. Stone** (2009). Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, **10**, 1633–1685.

44. **Taylor, M. E.** and **P. Stone** (2011). An introduction to intertask transfer for reinforcement learning. *AI Magazine*, **32**(1), 15.

45. **Tenenhaus, A.** and **M. Tenenhaus** (2011). Regularized generalized canonical correlation analysis. *Psychometrika*, **76**(2), 257–284.

46. **Utiyama, M.** and **H. Isahara**, A comparison of pivot methods for phrase-based statistical machine translation. *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Rochester, New York, 2007.

47. **Vinod, H.** (1976). Canonical ridge and econometrics of joint production. *Journal of Econometrics*, **4**(2), 147 – 166.

48. **Wang, W.**, **R. Arora**, **K. Livescu**, and **J. Bilmes**, On deep multi-view representation learning. *In ICML*. 2015.

49. **Wernsdorfer, M.** and **U. Schmid** (2014). Grounding hierarchical reinforcement learning models for knowledge transfer. *CoRR*, **abs/1412.6451**. URL `http://arxiv.org/abs/1412.6451`.

50. **Williams, R. J.** (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, **8**(3-4), 229–256.

51. **Wu, H.** and **H. Wang** (2007). Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, **21**(3), 165–181. ISSN 0922-6567.

52. **Zhang, M.**, **X. Duan**, **M. Liu**, **Y. Xia**, and **H. Li**, Joint alignment and artificial data generation: An empirical study of pivot-based machine transliteration. *In Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*. 2011. URL `http://aclweb.org/anthology/I/I11/I11-1135.pdf`.

53. **Zheng, Y.**, **Y. Zhang**, and **H. Larochelle** (2014*a*). A deep and autoregressive approach for topic modeling of multimodal data. *CoRR*, **abs/1409.3970**. URL `http://arxiv.org/abs/1409.3970`.

54. **Zheng, Y.**, **Y. Zhang**, and **H. Larochelle**, Topic modeling of multimodal data: An autoregressive approach. *In 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. 2014*b*. URL `http://dx.doi.org/10.1109/CVPR.2014.178`.

55. **Zou, W. Y.**, **R. Socher**, **D. Cer**, and **C. D. Manning**, Bilingual Word Embeddings for Phrase-Based Machine Translation. *In Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*. 2013.

# LIST OF PAPERS BASED ON THESIS

1. Janarthanan Rajendran, Mitesh M. Khapra, Sarath Chandar, Balaraman Ravindran, "**Bridge Correlational Neural Networks for multilingual multimodal representation learning**", Proceedings of NAACL, 2016. ArXiv preprint, abs/1510.03519.

2. Janarthanan Rajendran, Prasanna P, Balaraman Ravindran, Mitesh M. Khapra, "**ADAAPT: A Deep Architecture for Adaptive Policy Transfer from multiple sources**", Deep Reinforcement Learning Workshop, NIPS 2015. ArXiv preprint, abs/1510.02879.