Learning and Knowledge Transfer with Memory Networks for Machine Comprehension

Mohit Yadav
TCS Research New-Delhi
y.mohit@tcs.com

Lovekesh Vig
TCS Research New-Delhi
lovekesh.vig@tcs.com

Gautam Shroff
TCS Research New-Delhi
gautam.shroff@tcs.com

Abstract

Enabling machines to read and comprehend unstructured text remains an unfulfilled goal for NLP research. Recent research efforts on the "machine comprehension" task have managed to achieve close to ideal performance on simulated data. However, achieving similar levels of performance on small real world datasets has proved difficult; major challenges stem from the large vocabulary size, complex grammar, and the frequent ambiguities in linguistic structure. On the other hand, the requirement of human generated annotations for training, in order to ensure a sufficiently diverse set of questions is prohibitively expensive. Motivated by these practical issues, we propose a novel curriculum inspired training procedure for Memory Networks to improve the performance for machine comprehension with relatively small volumes of training data. Additionally, we explore various training regimes for Memory Networks to allow knowledge transfer from a closely related domain having larger volumes of labelled data. We also suggest the use of a loss function to incorporate the asymmetric nature of knowledge transfer. Our experiments demonstrate improvements on Dailymail, CNN, and MCTest datasets.

1 Introduction

A long-standing goal of NLP is to imbue machines with the ability to comprehend text and answer natural language questions. The goal is still distant and yet generates tremendous amount of interest due to the large number of potential NLP applications that are currently stymied because of their inability to deal with unstructured text. Also, the next generation of search engines are aiming to provide precise and semantically relevant answers in response to questions-as-queries; similar to the functionality of digital assistants like *Cortana* and *Siri*. This will require text understanding at a non-superficial level, in addition to reasoning, and, making complex inferences about the text.

As pointed out by Weston et al. (2016), the Question Answering (QA) task on unstructured text is a sound benchmark on which to evaluate machine comprehension. The authors also introduced *bAbI*: a simulation dataset for QA with multiple toy tasks. These toy tasks require a machine to perform simple induction, deduction, multiple chaining of facts, and, complex reasoning; which make them a sound benchmark to measure progress towards AI-complete QA (Weston et al., 2016). The recently proposed Memory Network architecture and its variants have achieved close to ideal performance, i.e., more than 95% accuracy on 16 out of a total of 20 QA tasks (Sukhbaatar et al., 2015; Weston et al., 2016).

While this performance is impressive, and is indicative of the memory network having sufficient capacity for the machine comprehension task, the performance does not translate to real world text (Hill et al., 2016). Challenges in real-world datasets stem from the much larger vocabulary, the complex grammar, and the often ambiguous linguistic structure; all of which further impede high levels of generalization performance, especially with small datasets. For instance, the empirical results reported by Hill et al. (2016) show that an end-to-end memory network with a single hop surpasses the performance achieved using multiple hops (i.e, higher capacity), when the model is trained with a simple heuristic. Similarly, Tapaswi et al. (2015) show that a memory network heavily overfits on the MovieQA dataset and

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 850–859,
Valencia, Spain, April 3-7, 2017.

yields near random performance. These results suggest that achieving good performance may not always be merely a matter of training high capacity models with large volumes of data. In addition to exploring new models there is a pressing need for innovative training methods, especially when dealing with real world sparsely labelled datasets.

With the advent of deep learning, the state of art performance for various semantic NLP tasks has seen a significant boost (Collobert and Weston, 2008). However, most of these techniques are *data-hungry*, and require a large number of sufficiently diverse labeled training samples, e.g., for QA, training samples should not only encompass an entire range of possible questions but also have them in sufficient quantity (Bordes et al., 2015). Generating annotations for training deep models requires a tremendous amount of manual effort and is often too expensive. Hence, it is necessary to develop effective techniques to exploit data from a related domain in order to reduce dependence on annotations. Recently, Memory Networks have been successfully applied to QA and dialogue-systems to work with a variety of disparate data sources such as movies, images, structured, and, unstructured text (Weston et al., 2016; Weston, 2016; Tapaswi et al., 2015; Bordes et al., 2015). Inspired from the recent success of Memory Networks, we study methods to train memory networks with small datasets by allowing for knowledge transfer from related domains where labelled data is more abundantly available.

The focus of this paper is to improve generalization performance of memory networks via an improved learning procedure for small real-world datasets and knowledge transfer from a related domain. In the process, this paper makes the following major contributions:

(i) A curriculum inspired training procedure for memory network is introduced, which yields superior performance with smaller datasets.

(ii) The exploration of knowledge transfer methods such as pre-training, joint-training and the proposed curriculum joint-training with a related domain having abundant labeled data.

(iii) A modified loss function for joint-training to incorporate the asymmetric nature of knowledge transfer, and also investigate the application of a pre-trained memory network on very small datasets such as MCTest dataset.

The remainder of the paper is organized as follows: Firstly, we provide a summary of related work in Section 2. Next in Section 3, we describe the machine comprehension task and the datasets utilized in our experiments. An introduction to memory networks for machine comprehension is presented in Section 4. Section 5 outlines the proposed methods for learning and knowledge transfer. Experimental details are provided in Section 6. We summarize our conclusions in Section 7.

2 Related Work

Memory Networks have been successfully applied to a broad range of NLP and machine learning tasks. These tasks include but are not limited to: performing reasoning over a simulated environment for QA (Weston et al., 2016), factoid and non-factoid based QA using both knowledge bases and unstructured text (Kumar et al., 2015; Hill et al., 2016; Chandar et al., 2016; Bordes et al., 2015), goal driven dialog(Bordes and Weston, 2016; Dodge et al., 2016; Weston, 2016), automatic story comprehension from both video and text (Tapaswi et al., 2015), and, transferring knowledge from one knowledge-base while learning to answer questions on a different knowledge base (Bordes et al., 2015). Recently, various other attention based neural models (similar to Memory Networks) have been proposed to tackle the machine comprehension task by QA from unstructured text (Kadlec et al., 2016; Sordoni et al., 2016; Chen et al., 2016). To the best of our knowledge, knowledge transfer from an unstructured text dataset to another unstructured text dataset for machine comprehension is not explored yet.

Training deep networks is known to be a notoriously hard problem and often the success of these techniques hinges upon achieving higher generalization performance with high capacity models (Blundell et al., 2015; Larochelle et al., 2009; Glorot and Bengio, 2010). To address this issue, *Curriculum learning* was firstly introduced by Bengio et al. (2009), which showed that training with gradually increasing difficulty leads to a better local minima, specially when working with non-convex loss functions. Although devising a universal curriculum strategy is hard, as even humans do not converge to one particular order in which concepts should be introduced (Rohde and Plaut, 1999) some notion of concept difficulty is normally utilized. With similar motivations, this pa-

per makes an attempt to exploit *curriculum learning* for machine comprehension with a memory network. Recently, curriculum learning has also been utilized to avoid negative transfer and make use of task relatedness for multi-task learning (Lee et al., 2016). Concurrently, Sachan and Xing (2016) have also studied curriculum learning for QA and unlike this paper, they do not consider learning and knowledge transfer on small real-world machine comprehension dataset in the setting of memory networks.

Pre-training & word2vec: Pre-training can often mitigate the issue that comes with random initialization used for network weights, by guiding the optimization process towards the basins of better local minima (Mishkin and Matas, 2016; Krahenbuhl et al., 2016; Erhan et al., 2010). An inspiration from the ripples created by the success of pre-training and as well as word2vec, this paper explores pre-training to utilize data from a related domain and also pre-trained vectors from word2vec tool (Mikolov et al., 2013). However, finding an optimal dimension for these pre-trained vectors and other involved hyper-parameters requires computationally extensive experiments.

Joint-training / Co-training / Multi-task learning / Domain adaptation: Previously, the utilization of common structures and similarities across different tasks / domains has been instrumental for various closely related learning tasks refereed as joint-training, co-training, multi-task learning and domain adaptation (Collobert and Weston, 2008; Liu et al., 2015; Chen et al., 2011; Maurer et al., 2016). To mitigate this ambiguity, in this paper, we limit ourselves to using "joint-training" and refrain from co-training, as unlike this work, co-training was initially introduced to exploit unlabelled data in the presence of small labelled data and two different and complementary views about the instances (Blum and Mitchell, 1998).

While this work looks conceptually similar, the proposed method tries to exploit information from a related domain and aims to achieve an asymmetric transfer only towards the specified domain, without any interest in the source domain, and hence should not be confused with the long-standing pioneering work on multi-task learning (Caruana, 1997). Another field of work that is related to this paper is on domain adaptation which appears to have two major related branches. The first branch is the recent work that has primarily focused on unsupervised domain adaptation (Nguyen and Grishman, 2015; Zhang et al., 2015), and the other is the traditional work on domain adaptation which has focussed on problems like entity recognition and not on machine comprehension and modern neural architectures (Ben-David et al., 2010; Daume III, 2007).

3 Machine Comprehension : Datasets and Tasks Description

Machine comprehension is the ability to read and comprehend text, i.e., understand its meaning, and can be evaluated by tasks involving the answering of questions posed on a context document. Formally, a set of tuples (q, C, S, s) is provided, where q is the question, C is the context document, S is a list of possible answers, and, s indicates the correct answer. Each of q, C, and S are sequence or words from a vocabulary V. Our aim is to train a memory network model to perform QA with small training datasets. We propose two primary ways to achieve this: 1) Improve the learning procedure to obtain better models, and 2) Demonstrate knowledge transfer from a related domain.

3.1 Data Description

Several corpora have been introduced for the machine comprehension task such as MCTest-160, MCTest-500, CNN, Dailymail, and, Children Boot Test (CBT) (Richardson et al., 2013; Hermann et al., 2015; Hill et al., 2016). The MCTest-160 and MCTest-500 have multiple-choice questions with associated narrative stories. Answers in these datasets can be one of these forms: a word, a phrase, or, a full sentence.

The remaining datasets are generated using Cloze-style questions; which are created by deleting a word from a sentence and asking the model to predict the deleted word. A place-holder token is substituted in place of the deleted word which is also the correct answer (Hermann et al., 2015). We have created three subsets of CNN namely, CNN-11K, CNN-22K and CNN-55K from the entire CNN dataset, and Dailymail-55K from the Dailymail dataset. Statistics on the number of samples comprising these datasets is presented in Table 1.

3.2 Improve Learning Procedure

It has been shown in the context of language modelling that presenting the training samples in an easy to hard ordering allows for shielding

	MCTest-160	MCTest-500	CNN-11K	CNN-22K	CNN-55K	Dailymail-55K
# Train	280	1400	11,000	22,000	55,000	55,000
# Validation	120	200	3,924	3,924	3,924	2,500
# Test	200	400	3,198	3,198	3,198	2,000
# Vocabulary	2856	4279	26,550	31,932	40,833	42,311
# Words $\notin$ Dailymail-55K	—	—	1,981	2,734	6,468	—

Table 1: Number of samples in training, valdiation, and, test samples in the MCTest-160, MCTest-500, CNN-11K, CNN-22K, CNN-55K, and, Dailymail-55K datasets; along with the size of vocabulary.

the model from very hard samples during training, yielding faster convergence and better models (Bengio et al., 2009). We investigate a curriculum learning inspired training procedure for memory networks to improve performance on the three subsets of the CNN dataset described below.

3.3 Demonstrate Knowledge Transfer

We plan to demonstrate knowledge transfer from Dailymail-55K to three subsets of CNN of varying sizes utilizing the proposed join-training method. For learning, we make use of smaller subsets of the CNN dataset. The smaller size of these subsets enables us to assess the performance boost due to knowledge transfer: As our aim is to demonstrate transfer when less labelled data is available, choosing the complete dataset would render gains from knowledge transfer as insignificant. We also demonstrate knowledge transfer for the case of MCTest dataset using embeddings obtained after training the memory network with CNN datasets.

4 End-to-end Memory Network for Machine Comprehension

End-to-end Memory Network is a recently introduced neural network model that can be trained in an end-to-end fashion; directly on the tuples (q, C, S, s) using standard back-propagation (Sukhbaatar et al., 2015). The complete training procedure can be described in the three steps: i) encoding the training tuples into the contextual memory, ii) attending context in memory to retrieve relevant information with respect to a question, and, iii) predicting the answer using the retrieved information. To accomplish the first step, an embedding matrix $A \in \mathbb{R}^{p \times d}$ is used to map both question and context into a p-dimensional embedding space; by applying the following transformations: $\overrightarrow{q} = A\Phi(q)$ and $\{\overrightarrow{m_i} = A\Phi(c_i)\}_{i=1,2,...,n}$. Where n is the number of items in context C and Φ is a bag-of-words representation in d-dimensional space, where d is typically the size of the vocabulary V. In the second step, the network senses relevant information present in the memory $\overrightarrow{m_i}$ for query $\overrightarrow{q}$, by computing the attention distribution $\{\alpha_i\}_{i=1,2,...,n}$, where $\alpha_i = \texttt{softmax}(\overrightarrow{m_i}^T \overrightarrow{q})$. Thereafter, α_i is used to aggregate the retrieved information into a vector representation $\overrightarrow{r_o}$ by utilizing another memory $\overrightarrow{r_i}$; as stated in Equation 1. The memory representation $\overrightarrow{r_i}$ is also defined as $\{\overrightarrow{r_i} = B\Phi(c_i)\}_{i=1,2,...,n}$ in a manner similar to $\overrightarrow{m_i}$ using another embedding matrix $B \in \mathbb{R}^{p \times d}$.

$$\overrightarrow{r_o} = \sum_{i=1}^{n} \alpha_i \overrightarrow{r_i} \tag{1}$$

$$\hat{a}_i = \texttt{softmax}((\overrightarrow{r_o} + \overrightarrow{q})^T U \Phi(s_i)) \tag{2}$$

In the last step, prediction distribution $\hat{a}_i$ is computed as in Equation 2, where $U \in \mathbb{R}^{p \times d}$ is an embedding matrix similar to A and can potentially be tied with A, and s_i is one of the answers in S. Using the prediction step, a probability distribution $\hat{a}_i$ over all s_i can be obtained and the final answer is selected as the one with the highest probability $\hat{a}_i$ corresponding to the option s_i.

$$\begin{aligned} L(P, D) = \frac{1}{N_D} \sum_{n=1}^{N_D} & a_n \times \texttt{log}(\hat{a}_n(P, D)) \\ & + (1 - a_n) \times \texttt{log}(1 - \hat{a}_n(P, D)) \end{aligned} \tag{3}$$

To train a memory network, the cross-entropy loss function L between the true label distribution $a_i \in \{0, 1\}^s$ (which is a one hot vector to indicate the correct label s in the training tuples) and the predicted distribution $\hat{a}_i$ is used, as in Equation 3. Where P, D and N_D represent the set of model parameters to learn, training dataset, and the number of tuples in the training set respectively. Such an objective can be easily optimized using stochastic gradient descent (SGD). A memory network can easily be extended to perform several hops over the memory before predicting the answer. For details, we refer to Hill et al. (2016). However, we constrain this study to use a single-hop network in order to reduce number of

parameters to learn and also the chances of overfitting; as we are dealing with small scale datasets.

Self-Supervision is a heuristic introduced to provide memory supervision and the rationale behind is that if the memory supporting the correct answer is retrieved than the model is more likely to predict the correct answer (Hill et al., 2016). More precisely, this is achieved by keeping a hard attention over memory while training, i.e., $m_o^{'} = \text{argmax}\ \alpha_i$. At each step of SGD, the model computes $m_o^{'}$ and updates only using those examples which do not select the memory $m_o^{'}$ having the correct answer in the corresponding c_i.

5 Proposed Methods

We attempt to improve the training procedure for Memory Networks in order to increase the performance for machine comprehension by QA with small scale datasets. Firstly, we introduce an improved training procedure for memory networks using curriculum learning which is termed as *Curriculum Inspired Training* (CIT) and offer details about this in Section 5.1. Thereafter, Section 5.2 explains joint-training method for knowledge transfer from an abundantly labelled dataset to another dataset with limited label information .

5.1 CIT: Curriculum Inspired Training

Curriculum learning makes use of the fact that model performance can be significantly improved if the training samples are not presented randomly but in such a way so as to make the learning task gradually more difficult by presenting examples in an easy to hard ordering (Bengio et al., 2009). Such a training procedure allows the learner to waste less time with noisy or hard to predict data when the model is not ready to incorporate such samples. However, what remains unanswered and is left as a matter of further exploration is how to devise an effective strategy for a given task?

$$SF(q, S, C, s) = \frac{\sum_{\texttt{word} \in \{q \cup S \cup C\}} \texttt{log}(\texttt{Freq.}(\texttt{word}))}{\#\{q \cup S \cup C\}} \quad (4)$$

In this work, we formulate a curriculum strategy to train a memory network for machine comprehension. Formally, we rank training tuples (q, S, C, s) from easy to hard based on the normalized word frequency for passage, question, and context initially; using the score function (SF) mentioned in Equation 4 (i.e. easier passages have more frequent words). The training data is then divided into a fixed number of chapters, with each successive chapter resulting in addition of more difficult tuples. The model is then trained sequentially on each chapter with the final chapter containing the complete training data. The presence of both the number of chapters and the fixed number of epochs per chapter makes such a strategy flexible and allows to be tailored to different data after optimizing the like other hyper-parameters.

$$L(P, D, en) = \frac{1}{N_D} \sum_{n=1}^{N_D} (a_n \times \texttt{log}(\hat{a}_n(P, D)) + (1 - a_n) \times \texttt{log}(1 - \hat{a}_n(P, D)) \times \mathbf{1}(en, c(n) \times epc) \quad (5)$$

The loss function used for curriculum inspired training varies with epoch number; as mentioned in Equation 5. Note, in Equation 5, en and $c(n)$ represents the current epoch number and chapter number for n^{th} tuple assigned using rank allocated based on SF mentioned in Equation 4 respectively. epc, P, D, and $\mathbf{1}$ is the number of epochs per chapter, model parameters, training set, and an indicator function which is one if first argument is $>=$ the second argument or else zero; respectively.

5.2 Joint-Training for Knowledge Transfer

While joint-training methods offer knowledge transfer by exploiting similarities and regularities across different tasks or datasets, the asymmetric nature of transfer and skewed proportion of datasets is usually not handled in a sound way. Here, we devise a training loss function $\hat{L}$ to relieve both of these involved issues while doing joint-training with a target dataset (TD) with fewer training samples and a source dataset (SD) having label information for higher number of examples; as mentioned in Equation 6.

$$\hat{L}(P, TD, SD) = 2 \times \gamma \times L(P, TD) + 2 \times (1 - \gamma) \times L(P, SD) \times F(N_{TD}, N_{SD}) \quad (6)$$

Where $\hat{L}$ represents the devised loss function for joint-training for transfer, L the cross-entropy loss function also mentioned earlier in Equation 3, γ is a weighting factor which varies between zero and one, $F(N_{TD}, N_{SD})$ is an another weighting factor which is a function of number of samples in the target domain N_{TD} and in the source domain N_{SD}. The rationale behind γ factor is to control the relative update in the network due to

samples from source and target datasets; which permits biasing of the model performance towards one dataset. $F(N_{TD}, N_{SD})$ factor can be independently utilized to mitigate the effect of skewed proportion in the number of samples present in both target and source domains. Note, maintaining both γ and $F(N_{TD}, N_{SD})$ as separate parameters allows for restricting γ within (0,1) without any extra computation as described below.

5.3 Improved Loss Functions

This paper explores the following variants of the introduced loss function $\hat{L}$ for knowledge transfer via joint-training:

1. Joint-training (Jo-Train):- $\gamma = 1/2$ and $F(N_{TD}, N_{SD})$ = 1.
2. Weighted joint-training (W+Jo-Train):- $\gamma = (0, 1)$ and $F(N_{TD}, N_{SD}) = N_{TD}/N_{SD}$.
3. Curriculum joint-training (CIT+Jo-Train):- $L(P, TD)$ & $L(P, SD)$ of Equation 6 are replaced by their analogous terms $L(P, TD, en)$ & $L(P, SD, en)$ generated using Equation 5; γ = $1/2$ and $F(N_{TD}, N_{SD})$ = 1.
4. Weighted curriculum joint-training (W+CIT+Jo-Train):- $L(P, TD)$ & $L(P, SD)$ of Equation 6 are replaced by analogous $L(P, TD, en)$ & $L(P, SD, en)$ generated using Equation 5; γ = (0,1) and $F(N_{TD}, N_{SD})$ = N_{TD}/N_{SD}.
5. Source only (SrcOnly) :- γ = 0.

The $F(N_{TD}, N_{SD})$ factor does not increase computation as it is not optimized for any of the cases. Jo-Train (Liu et al., 2015), SrcOnly and a method similar to W+Jo-Train (Daume III, 2007) have also been explored previously for other NLP tasks and models.

6 Experiments

We evaluate the performance on datasets introduced earlier in Section 3. We first present baseline methods, pre-processing and training details. In Section 6.3, we present results on CNN-11/22/55K, MCTest-160 and MCTest-50 to validate our claims mentioned in Section 1. All of the methods presented here are implemented in Theano (Bastien et al., 2012) and Lasagne (Dieleman et al., 2015) and are run on a single GPU (Tesla K40c) server with 500GB of memory.

6.1 Baseline Methods

We implemented Sliding Window (SW) and Sliding Window + Distance (SW+D)(Richardson et al., 2013) as baselines to compare against our experiments. Further, we augment SW (or SW+D) to incorporate distances between word vectors of the question and the context over the sliding window; in a manner similar to the way SW+D is augmented from SW by Richardson et al. (2013). These approaches are named based upon the source of pre-trained word vectors, e.g., SW+D+CNN-11K+W2V utilizes vectors estimated from both CNN-11K and word2vec pre-trained vectors[1]. In case of more than one source, individual distances are summed and utilized for final scoring. Results on MCTest for SW, SW+D, and their augmented approaches are reported using online available scores for all answers [2].

Meaningful Comparisons: To ascertain that the improvement is due to the proposed training methods, and not merely because of addition of more data, we built multiple baselines, namely, initialization using word vectors from word2vec, pre-training, Jo-train, and SrcOnly. For pre-training and word2vec, words $\in$ target dataset and $\notin$ source dataset are initialized, by a uniform random sampling with the limits set to the extremes spanned by the word vectors in the source domain. *It is worth to note that the pre-training and Jo-train utilizes as much label information and data as other proposed variants of joint-training. Also, SrcOnly method is an indicative of how much direct knowledge transfer from source domain to target domain can be achieved without any learning.*

6.2 Pre-processing & Training Details

While processing data, we replace words occurring less than 5 times by <unk> token except for MCTest datasets. Additionally, all entities are included in vocabulary. All models are trained by carrying out the optimization using SGD with learning rate in $\{10^{-4}, 10^{-3}\}$, momentum value set to 0.9, weight decay in $\{10^{-5}, 10^{-4}\}$, and, max norm in $\{1, 10, 40\}$. We kept length of window equal to 5 for CNN / Dailymail datasets(Hill et al., 2016) and for MCTest datasets is chosen from $\{3, 5, 8, 10, 12\}$. For embedding size, we look for the optimal value in $\{50, 100, 150, 200, 300\}$ for

[1] http://code.google.com/p/word2vec

[2] http://research.microsoft.com/en-us/um/redmond/projects/mctest/results.htm

	CNN-11 K			CNN-22 K			CNN-55 K		
Model + Training Methods	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
SW §	21.33	20.35	21.48	21.80	20.61	20.76	21.54	19.87	20.66
SW+D §	25.45	25.40	25.90	25.61	25.25	26.47	25.85	25.74	26.94
SW+W2V §	43.90	43.01	42.60	45.70	44.10	42.23	45.06	44.50	43.50
MemNN §	98.98	45.96	46.08	98.07	49.28	51.42	97.31	54.98	56.69
MemNN+CIT §	96.44	47.17	**49.04**	98.36	52.43	**52.73**	91.14	57.26	**57.68**
SW+Dailymail ‡	30.19	31.21	30.60	31.70	30.87	32.01	31.56	33.07	31.08
MemNN+W2V ‡	86.57	43.78	45.99	94.1	49.98	51.06	95.2	51.47	53.66
MemNN+SrcOnly ‡	25.12	26.78	27.08	25.43	26.78	27.08	24.79	26.78	27.08
MemNN+Pre-train ‡	92.82	52.87	52.06	95.12	53.59	55.35	96.33	56.64	59.19
MemNN+Jo-train ‡	65.78	53.85	55.06	64.85	55.94	55.69	77.32	57.76	57.99
MemNN+CIT+Jo-train ‡	77.74	55.93	55.74	78.96	55.98	56.85	71.89	56.83	59.07
MemNN+W+Jo-train‡	71.72	54.30	55.70	79.64	55.91	56.73	71.15	57.62	58.34
MemNN+W+CIT+Jo-train ‡	80.14	56.91	**57.02**	79.04	57.90	**57.71**	76.91	58.14	**59.88**

Table 2: Train, validation and test percentage accuracy on CNN-11/22/55K datasets. § and ‡ indicate that the data used comes from either of CNN-11/22/55K and also from Dailymail-55K along with either of CNN-11/22/55K respectively. Random test accuracy on these datasets is 3.96% approximately.

CNN / Dailymail datasets. For CNN / Dailymail, we have trained memory network using a single batch with self-supervision heuristic (Hill et al., 2016). In case of curriculum learning, the number of chapters are optimized out of $\{3, 5, 8, 10\}$ and number of epochs per chapter is set equal to $\frac{2M}{M+1} \times \frac{ed_{ncl}}{ed_{cl}} \times EN$ which is estimated by equating to the number of network update found for the optimal case of non-curriculum learning. Here M and ed_{cl} represents the number of chapter and embedding size for curriculum learning, and ed_{ncl} & EN represents the optimal value found for embedding size and number of epochs without curriculum learning. We use early stopping with a validation set while training the network.

6.3 Results & Discussion

In this section, we present results to validate contributions mentioned in Section 1. Table 2 presents the results of our approaches along with results from baseline methods SW, SW+D, SW+W2V, and a standard memory network (MemNN). Results for CIT on CNN-11/22/55K (MemNN+CIT) show an absolute improvement of 2.96%, 1.31%, and, 1.00% respectively, when compared with the memory network (MemNN) (*contribution (i)*). Figure 1 shows that the CIT leads to better convergence when compared without CIT on CNN-11K.

As baselines for knowledge transfer from the Dailymail-55K dataset to CNN-11/22/55K datasets, Table 2 presents results for SW+Dailymail, memory network initialized with word2vec (MemNN+W2V), memory network trained on Dailymail (MemNN+SrcOnly), memory network initialized with pre-trained embeddings from Dailymail (MemNN+Pre-train) and memory network jointly-trained with both Dailymail and CNN (MemNN+Jo-train) (*contribution (ii)*). Further, results show the knowledge transfer observed when MemNN+CIT+Jo-train and MemNN+W+Jo-Train are utilized to train Dailymail-55K with CNN-11/22/55K. On combining the MemNN+CIT+Jo-train with MemNN+W+Jo-Train (which is MemNN+W+CIT+Jo-Train), a significant and consistent improvement can be observed; as the performance goes up by 1.96%, 2.03%, and, 1.89% on CNN-11/22/55K respectively; when compared against the other competitive baselines (*contribution (ii) & (iii)*).

Results empirically support the major premise of this study, i.e., CIT and knowledge transfer from a related dataset with memory network can significantly improve the performance; improvements of 10.94%, 6.28%, and, 3.19% are observed with CNN-11/22/55K respectively when compared with the standard memory network. The improvement in knowledge transfer decreases as the amount of data in the target domain starts increasing from 11K to 55K, as the volume of data in the target domain starts becoming comparable to source domain, and is enough to achieve similar level of performance without knowledge transfer.

Previously, Chen et al. (2016) annotated a sample of 100 questions on CNN stories based on the type of capabilities required to answer the question. We report results for all 6 specific categories in Table 3. Even with CNN-11K and Dailymail-55K which is roughly 20% of the complete CNN dataset, the proposed methods achieve similar per-

Model + Training Methods	Exact	Para.	Part.Clue	Multi.Sent.	Co-ref.	Ambi./Hard
SW §	3(23.1%)	12(29.2%)	2(10.5%)	0(0.0%)	0(0.0%)	2(11.7%)
SW+D §	6(46.1%)	14(34.1%)	2(10.5%)	0(0.0%)	0(0.0%)	3(17.6%)
SW+W2V §	10(76.9%)	20(48.7%)	5(26.3%)	0(0.0%)	0(0.0%)	7(41.1%)
MemNN §	8(61.5%)	20(48.7%)	12(63.1%)	1(50.0%)	0(0.0%)	2(11.7%)
MemNN+CIT §	10(76.9%)	19(46.3%)	12(63.1%)	1(50.0%)	3(37.5%)	2(11.7%)
SW+Dailymail ‡	6(46.1%)	19(46.3%)	5(26.3%)	0(0.0%)	0(0.0%)	2(11.7%)
MemNN+W2V ‡	6(46.1%)	27(65.8%)	5(26.3%)	0(0.0%)	0(0.0%)	7(41.1%)
MemNN+SrcOnly §	6(46.1%)	12(29.2%)	2(10.5%)	0(0.0%)	0(0.0%)	2(11.7%)
MemNN+Pre-train ‡	11(84.6%)	25(60.9%)	12(63.1%)	0(0.0%)	0(0.0%)	1(5.9%)
MemNN+Jo-train ‡	8(61.5%)	29(70.7%)	10(52.6%)	2(100%)	0(0.0%)	5(29.4%)
MemNN+CIT+Jo-train ‡	10(76.9%)	27(65.8%)	10(52.6%)	0(0.0%)	3(37.5%)	5(29.4%)
MemNN+W+Jo-train ‡	11(84.6%)	29(70.7%)	10(52.6%)	2(100%)	0(0.0%)	5(29.4%)
MemNN+W+CIT+Jo-train ‡	**11(84.6%)**	27(65.8%)	10(52.6%)	**2(100%)**	**3(37.5%)**	**5(29.4%)**
Chen et al. (2016) $	13(100%)	39(95.1%)	17(89.5%)	1(50.0%)	3(37.5%)	1(5.9%)
Sordoni et al. (2016) $	13(100%)	39(95.1%)	16(84.2%)	1(50.0%)	3(37.5%)	5(29.4%)
Total Number Of Samples	13	41	19	2	8	17

Table 3: Question-specific category analysis of percentage test accuracy with only learning and knowledge transfer methods on CNN-11K dataset. § and ‡ indicates that the data used comes from CNN-11K and from Dailymail-55K along with CNN-11K respectively. $ indicate results from Sordoni et al. (2016).

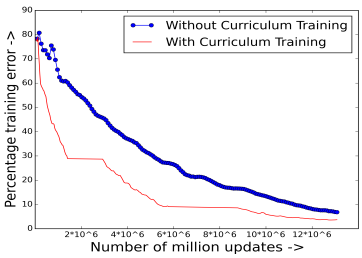

Figure 1: Percentage training error v/s number of million updates while training on CNN-11K with or without curriculum inspired training.

	MCTest-160			MCTest-500		
Training Methods	One	Multi.	All	One	Multi.	All
SW	66.07	53.12	59.16	54.77	53.04	53.83
SW+D	75.89	60.15	67.50	63.23	57.01	59.83
SW+D+W2V	79.46	59.37	68.75	65.07	58.84	61.67
SW+D+CNN-11K	79.78	59.37	**67.67**	64.33	57.92	**60.83**
SW+D+CNN-22K	76.78	60.93	**68.33**	64.70	59.45	**61.83**
SW+D+CNN-55K	78.57	59.37	**68.33**	65.07	59.75	**62.16**
SW+D+CNN-11K+W2V	77.67	59.41	68.69	65.07	61.28	63.00
SW+D+CNN-22K+W2V	78.57	60.16	69.51	66.91	60.00	63.13
SW+D+CNN-55K+W2V	79.78	60.93	**70.51**	66.91	60.67	**63.50**

Table 4: Knowledge transfer results on MCTest-160 and MCTest-500 datasets. One and Multi. indicates the questions that require one and multiple supporting facts. Random test accuracy is 25% here, as number of options are 4.

formance on 4 out of 6 categories, when compared to latest models (2^{nd} & 3^{rd} last rows of Table 3).

On very small datasets such as MCTest-160 and MCTest-500, it is not feasible to train memory network (Smith et al., 2015), therefore, we explore the use of word vectors from the embedding matrix of a model pre-trained on CNN datasets. Here, the embedding matrix refers to the encoding matrix A used in the first step of memory network as mentioned in Section 4. SW+D+CNN-11/22/55K are the results when the similarity measures comes from SW+D as mentioned in Section 6.1 and also using the word vectors from encoding matrix A obtained after training on CNN-11/22/55K. From table 4, it is evident that performance improves as the amount of data increases in CNN domain (*contribution(iii)*). Further, on combining with word2vec distance (SW+D+CNN-11/22/55K+W2V), an improvement is observed.

7 Conclusion

Looking at the widespread applications of Memory Networks and the prohibitive data requirements for training them, this paper seeks to improve the performance of memory networks on small datasets in two different ways. Firstly, this paper introduces an effective CIT procedure for machine comprehension. Secondly, this paper explores various methods to exploit labelled data from closely related domains; in order to perform knowledge transfer and improve performance. Additionally, this paper suggests the use of a modified loss function to further incorporate the asymmetric nature of knowledge transfer. Beyond machine comprehension, we believe that the proposed methods are likely to achieve higher generalization for other tasks utilizing memory network style architectures, by virtue of the proposed CIT method and joint-training for knowledge transfer.

References

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning Neural Information Processing Systems (NIPS) Workshop.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1):151–175.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 41–48. ACM.

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

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Antonie Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. arXiv preprint arXiv:1606.03126.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75, July.

Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. 2016. Hierarchical memory networks. arXiv preprint arXiv:1605.07427.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. *Advances in Neural Information Processing Systems (NIPS)*, pages 2456–2464.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 160–167, New York, NY, USA. ACM.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June. Association for Computational Linguistics.

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, J Kelly, et al. 2015. Lasagne: First release. *Zenodo: Geneva, Switzerland*.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating prerequisite qualities for learning end-to-end dialog systems. *In Proceedings of International Conference on Learning Representations (ICLR)*.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research (JMLR)*, 11:625–660.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems (NIPS)*, pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. *In Proceedings of International Conference on Learning Representations (ICLR)*.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*.

Philipp Krahenbuhl, Carl Doersch, Jeff Donahue, and Trevor Darrell. 2016. Data-dependent initializations of convolutional neural networks. *In Proceedings of International Conference on Learning Representations (ICLR)*.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. 2009. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research (JMLR)*, 10:1–40.

Giwoong Lee, Eunho Yang, and Sung Ju Hwang. 2016. Asymmetric multi-task learning based on task relatedness and loss. In *Proceedings of the 33rd Annual International Conference on Machine Learning (ICML)*, pages 230–238.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921, Denver, Colorado, May–June. Association for Computational Linguistics.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The benefit of multitask representation learning. *Journal of Machine Learning Research (JMLR)*, 17(81):1–32.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Dmytro Mishkin and Jiri Matas. 2016. All you need is a good init. *In Proceedings of International Conference on Learning Representations (ICLR)*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China, July. Association for Computational Linguistics.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October. Association for Computational Linguistics.

Douglas L.T. Rohde and David C. Plaut. 1999. Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109.

Mrinmaya Sachan and Eric P. Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. *In Proceedings of Association for Computational Linguistics (ACL)*.

Ellery Smith, Nicola Greco, Matko Bosnjak, and Andreas Vlachos. 2015. A strong lexical matching method for the machine comprehension test. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1693–1698, Lisbon, Portugal, September. Association for Computational Linguistics.

Alessandro Sordoni, Phillip Bachman, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. arXiv preprint arXiv:1606.02245.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 2440–2448.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Movieqa: Understanding stories in movies through question-answering. *arXiv preprint arXiv:1512.02902*.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *In Proceedings of International Conference on Learning Representations (ICLR)*.

Jason Weston. 2016. Dialog-based language learning. *arXiv preprint arXiv:1604.06045*.

Xu Zhang, Felix X. Yu, Shih-Fu Chang, and Shengjin Wang. 2015. Deep transfer network: Unsupervised domain adaptation. *arXiv preprint arXiv:1503.00591*.