Supervised and Unsupervised Transfer Learning for Question Answering

Yu-An Chung[1] **Hung-Yi Lee**[2] **James Glass**[1]
[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA
{andyyuan,glass}@mit.edu

[2]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
hungyilee@ntu.edu.tw

Abstract

Although transfer learning has been shown to be successful for tasks like object and speech recognition, its applicability to question answering (QA) has yet to be well-studied. In this paper, we conduct extensive experiments to investigate the transferability of knowledge learned from a source QA dataset to a target dataset using two QA models. The performance of both models on a TOEFL listening comprehension test (Tseng et al., 2016) and MCTest (Richardson et al., 2013) is significantly improved via a simple transfer learning technique from MovieQA (Tapaswi et al., 2016). In particular, one of the models achieves the state-of-the-art on all target datasets; for the TOEFL listening comprehension test, it outperforms the previous best model by 7%. Finally, we show that transfer learning is helpful even in unsupervised scenarios when correct answers for target QA dataset examples are not available.

1 Introduction

1.1 Question Answering

One of the most important characteristics of an intelligent system is to understand stories like humans do. A story is a sequence of sentences, and can be in the form of plain text (Trischler et al., 2017; Rajpurkar et al., 2016; Weston et al., 2016; Yang et al., 2015) or spoken content (Tseng et al., 2016), where the latter usually requires the spoken content to be first transcribed into text by automatic speech recognition (ASR), and the model will subsequently process the ASR output. To evaluate the extent of the model's understanding of the story, it is asked to answer questions about the story. Such a task is referred to as question answering (QA), and has been a long-standing yet challenging problem in natural language processing (NLP).

Several QA scenarios and datasets have been introduced over the past few years. These scenarios differ from each other in various ways, including the length of the story, the format of the answer, and the size of the training set. In this work, we focus on context-aware multi-choice QA, where the answer to each question can be obtained by referring to its accompanying story, and each question comes with a set of answer choices with only one correct answer. The answer choices are in the form of open, natural language sentences. To correctly answer the question, the model is required to understand and reason about the relationship between the sentences in the story.

1.2 Transfer Learning

Transfer learning (Pan and Yang, 2010) is a vital machine learning technique that aims to use the knowledge learned from one task and apply it to a different, but related, task in order to either reduce the necessary fine-tuning data size or improve performance. Transfer learning, also known as domain adaptation[1], has achieved success in numerous domains such as computer vision (Sharif Razavian et al., 2014), ASR (Doulaty et al., 2015; Huang et al., 2013), and NLP (Zhang et al., 2017; Mou et al., 2016). In computer vision, deep neural networks trained on a large-scale image classification dataset such as ImageNet (Russakovsky et al., 2015) have proven to be excellent feature extractors for a broad range of visual tasks such as image captioning (Lu et al., 2017; Karpathy and Fei-Fei, 2015; Fang et al., 2015) and visual

[1]In this paper, we do not distinguish conceptually between transfer learning and domain adaptation. A 'domain' in the sense we use throughout this paper is defined by datasets.

Proceedings of NAACL-HLT 2018, pages 1585–1594
New Orleans, Louisiana, June 1 - 6, 2018.

question answering (Xu and Saenko, 2016; Fukui et al., 2016; Yang et al., 2016; Antol et al., 2015), among others. In NLP, transfer learning has also been successfully applied to tasks like sequence tagging (Yang et al., 2017), syntactic parsing (McClosky et al., 2010) and named entity recognition (Chiticariu et al., 2010), among others.

1.3 Transfer Learning for QA

Although transfer learning has been successfully applied to various applications, its applicability to QA has yet to be well-studied. In this paper, we tackle the TOEFL listening comprehension test (Tseng et al., 2016) and MCTest (Richardson et al., 2013) with transfer learning from MovieQA (Tapaswi et al., 2016) using two existing QA models. Both models are pre-trained on MovieQA and then fine-tuned on each target dataset, so that their performance on the two target datasets are significantly improved. In particular, one of the models achieves the state-of-the-art on all target datasets; for the TOEFL listening comprehension test, it outperforms the previous best model by 7%.

Transfer learning without any labeled data from the target domain is referred to as unsupervised transfer learning. Motivated by the success of unsupervised transfer learning for speaker adaptation (Chen et al., 2011; Wallace et al., 2009) and spoken document summarization (Lee et al., 2013), we further investigate whether unsupervised transfer learning is feasible for QA.

Although not well studied in general, transfer Learning for QA has been explored recently. To the best of our knowledge, Kadlec et al. (2016) is the first work that attempted to apply transfer learning for machine comprehension. The authors showed only limited transfer between two QA tasks, but the transferred system was still significantly better than a random baseline. Wiese et al. (2017) tackled a more specific task of biomedical QA with transfer learning from a large-scale dataset. The work most similar to ours is by Min et al. (2017), where the authors used a simple transfer learning technique and achieved significantly better performance. However, none of these works study unsupervised transfer learning, which is especially crucial when the target dataset is small. Golub et al. (2017) proposed a two-stage synthesis network that can generate synthetic questions and answers to augment insufficient training data without annotations. In this work, we aim to handle the case that the questions from the target domain are available.

2 Task Descriptions and Approaches

Among several existing QA settings, in this work we focus on multi-choice QA (MCQA). We are particularly interested in understanding whether a QA model can perform better on one MCQA dataset with knowledge transferred from another MCQA dataset. In Section 2.1, we first formalize the task of MCQA. We then describe the procedures for transfer learning from one dataset to another in Section 2.2. We consider two kinds of settings for transfer learning in this paper, one is supervised and the other is unsupervised.

2.1 Multi-Choices QA

In MCQA, the inputs to the model are a story, a question, and several answer choices. The story, denoted by $\mathbf{S}$, is a list of sentences, where each of the sentences is a sequence of words from a vocabulary set V. The question and each of the answer choices, denoted by $\mathbf{Q}$ and $\mathbf{C}$, are both single sentences also composed of words from V. The QA model aims to choose one correct answer from multiple answer choices based on the information provided in $\mathbf{S}$ and $\mathbf{Q}$.

2.2 Transfer Learning

The procedure of transfer learning in this work is straightforward and includes two steps. The first step is to pre-train the model on one MCQA dataset referred to as the **source** task, which usually contains abundant training data. The second step is to fine-tune the same model on the other MCQA dataset, which is referred to as the **target** task, that we actually care about, but that usually contains much less training data. The effectiveness of transfer learning is evaluated by the model's performance on the target task.

Supervised Transfer Learning

In supervised transfer learning, both the source and target datasets provide the correct answer to each question during pre-training and fine-tuning, and the QA model is guided by the correct answer to optimize its objective function in a supervised manner in both stages.

Unsupervised Transfer Learning

We also consider unsupervised transfer learning where the correct answer to each question in the target dataset is not available. In other words, the entire process is supervised during pre-training, but unsupervised during fine-tuning. A self-labeling technique inspired by Lee et al. (2013); Chen et al. (2011); Wallace et al. (2009) is used during fine-tuning on the target dataset. We present the proposed algorithm for unsupervised transfer learning in Algorithm 1.

Algorithm 1 Unsupervised QA Transfer Learning

Input: Source dataset with correct answer to each question; Target dataset without any answer; Number of training epochs.

Output: Optimal QA model M^*

1: Pre-train QA model M on the source dataset.
2: **repeat**
3: For each question in the target dataset, use M to predict its answer.
4: For each question, assign the predicted answer to the question as the correct one.
5: Fine-tune M on the target dataset as usual.
6: **until** Reach the number of training epochs.

3 Datasets

We used MovieQA (Tapaswi et al., 2016) as the source MCQA dataset, and TOEFL listening comprehension test (Tseng et al., 2016) and MCTest (Richardson et al., 2013) as two separate target datasets. Examples of the three datasets are shown in Table 1.

MovieQA is a dataset that aims to evaluate automatic story comprehension from both video and text. The dataset provides multiple sources of information such as plot synopses, scripts, subtitles, and video clips that can be used to infer answers. We only used the plot synopses of the dataset, so our setting is the same as pure textual MCQA. The dataset contains 9,848/1,958 train/dev examples; each question comes with a set of five possible answer choices with only one correct answer.

TOEFL listening comprehension test is a recently published, very challenging MCQA dataset that contains 717/124/122 train/dev/test examples. It aims to test knowledge and skills of academic English for global English learners whose native languages are not English. There are only four answer choices for each question. The stories in this dataset are in audio form. Each story comes with two transcripts: manual and ASR transcriptions, where the latter is obtained by running the CMU Sphinx recognizer (Walker et al., 2004) on the original audio files. We use TOEFL-manual and TOEFL-ASR to denote the two versions, respectively. We highlight that the questions in this dataset are not easy because most of the answers cannot be found by simply matching the question and the choices without understanding the story. For example, there are questions regarding the gist of the story or the conclusion for the conversation.

MCTest is a collection of 660 elementary-level children's stories. Each question comes with a set of four answer choices. There are two variants in this dataset: MC160 and MC500. The former contains 280/120/240 train/dev/test examples, while the latter contains 1,200/200/600 train/dev/test examples and is considered more difficult.

The two chosen target datasets are challenging because the stories and questions are complicated, and only small training sets are available. Therefore, it is difficult to train statistical models on only their training sets because the small size limits the number of parameters in the models, and prevents learning any complex language concepts simultaneously with the capacity to answer questions. We demonstrate that we can effectively overcome these difficulties via transfer learning in Section 5.

4 QA Neural Network Models

Among numerous models proposed for multiple-choice QA (Trischler et al., 2016; Fang et al., 2016; Tseng et al., 2016), we adopt the End-to-End Memory Network (MemN2N)[2] (Sukhbaatar et al., 2015) and Query-Based Attention CNN (QACNN)[3] (Liu et al., 2017), both open-sourced, to conduct the experiments. Below we briefly introduce the two models in Section 4.1 and Section 4.2, respectively. For the details of the models, please refer to the original papers.

4.1 End-to-End Memory Networks

An End-to-End Memory Network (MemN2N) first transforms **Q** into a vector representation with

[2]MemN2N was originally designed to output a single word within a fixed vocabulary set. To apply it to MCQA, some modification is needed. We describe the modifications in Section 4.1.

[3]https://github.com/chun5212021202/ACM-Net

	Source Dataset	Target Dataset	
	MovieQA	TOEFL	MCTest
$\mathbf{S}$	After entering the boathouse, the trio witness Voldemort telling Snape that the elder Wand cannot serve Voldemort until Snape dies ... Before dying, Snape tells Harry to take his memories to the Pensieve ...	I just wanted to take a few minutes to meet with everyone to make sure your class presentations for next week are all in order and coming along well. And as you know, you're supposed to report on some areas of recent research on genetics ...	James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food ... Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home ...
$\mathbf{Q}$	What does Snape tell Harry before he dies?	Why does the professor meet with the student?	What did James do after he ordered the fries?
$\mathbf{C}_1$	To bury him in the forest	To find out if the student is interested in taking part in a genetics project	went to the grocery store
$\mathbf{C}_2$	That he always respected him	To discuss the student's experiment on the taste perception	**went home without paying**
$\mathbf{C}_3$	To remember to him for the good deeds	**To determine if the student has selected an appropriate topic for his class project**	ate them
$\mathbf{C}_4$	**To take his memories to the Pensieve**	To explain what the student should focus on for his class presentation	made up his mind to be a better turtle
$\mathbf{C}_5$	To write down his memories in a book		

Table 1: Example of the story-question-choices triplet from MovieQA, TOEFL listening comprehension test, and MCTest datasets. $\mathbf{S}, \mathbf{Q}$, and $\mathbf{C}_i$ denote the story, question, and one of the answer choices, respectively. For MovieQA, each question comes with five answer choices; and for TOEFL and MCTest, each question comes with only four answer choices. The correct answer is marked in bold.

an embedding layer B. At the same time, all sentences in $\mathbf{S}$ are also transformed into two different sentence representations with two additional embedding layers A and C. The first sentence representation is used in conjunction with the question representation to produce an attention-like mechanism that outputs the similarity between each sentence in $\mathbf{S}$ and $\mathbf{Q}$. The similarity is then used to weight the second sentence representation. We then obtain the sum of the question representation and the weighted sentence representations over $\mathbf{S}$ as $\mathbf{Q}'$. In the original MemN2N, $\mathbf{Q}'$ is decoded to provide the estimation of the probability of being an answer for each word within a fixed set. The word with the highest probability is then selected as the answer. However, in multiple-choice QA, $\mathbf{C}$ is in the form of open, natural language sentences instead of a single word. Hence we modify MemN2N by adding an embedding layer F to encode $\mathbf{C}$ as a vector representation $\mathbf{C}'$ by averaging the embeddings of words in $\mathbf{C}$. We then compute the similarity between each choice representation $\mathbf{C}'$ and $\mathbf{Q}'$. The choice $\mathbf{C}$ with the highest probability is then selected as the answer.

4.2 Query-Based Attention CNN

A Query-Based Attention CNN (QACNN) first uses an embedding layer E to transform $\mathbf{S}, \mathbf{Q}$, and $\mathbf{C}$ into a word embedding. Then a compare layer generates a story-question similarity map $\mathbf{SQ}$ and a story-choice similarity map $\mathbf{SC}$. The two similarity maps are then passed into a two-stage CNN architecture, where a question-based attention mechanism on the basis of $\mathbf{SQ}$ is applied to each of the two stages. The first stage CNN generates a word-level attention map for each sentence in $\mathbf{S}$, which is then fed into the second stage CNN to generate a sentence-level attention map, and yield choice-answer features for each of the choices. Finally, a classifier that consists of two fully-connected layers collects the information from every choice answer feature and outputs the most likely answer. The trainable parameters are the embedding layer E that transforms $\mathbf{S}, \mathbf{Q}$, and $\mathbf{C}$ into word embeddings, the two-stage CNN $W_{CNN}^{(1)}$ and $W_{CNN}^{(2)}$ that integrate information from the word to the sentence level, and from the sentence to the story level, and the two fully-connected layers $W_{FC}^{(1)}$ and $W_{FC}^{(2)}$ that make the final prediction. We mention the trainable parameters here because in Section 5 we will conduct experiments to analyze the transferability of the QACNN by fine-tuning some parameters while keeping others fixed. Since QACNN is a newly proposed QA model has a relatively complex structure, we illustrate its architecture in Figure 1, which is enough for understanding the rest of the paper. Please refer to the original paper (Liu et al., 2017) for more details.

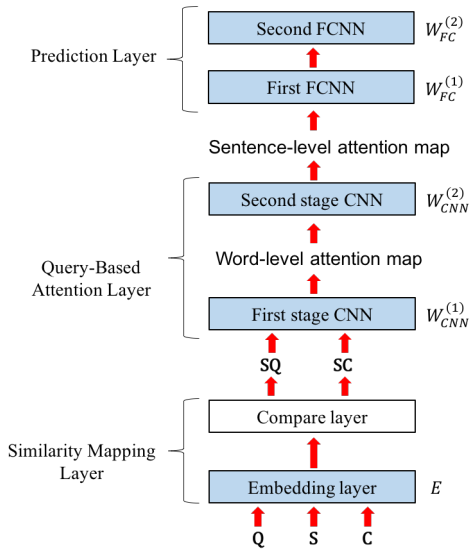


Figure 1: QACNN architecture overview. QACNN consists of a similarity mapping layer, a query-based attention layer, and a prediction layer. The two-stage attention mechanism takes place in the query-based attention layer, yielding word-level and sentence-level attention map, respectively. The trainable parameters, including $E, W_{CNN}^{(1)}, W_{CNN}^{(2)}, W_{FC}^{(1)}$, and $W_{FC}^{(2)}$, are colored in light blue.

5 Question Answering Experiments

5.1 Training Details

For pre-training MemN2N and QACNN on MovieQA, we followed the exact same procedure as in Tapaswi et al. (2016) and Liu et al. (2017), respectively. Each model was trained on the training set of the MovieQA task and tuned on the dev set, and the best performing models on the dev set were later fine-tuned on the target dataset. During fine-tuning, the model was also trained on the training set of target datasets and tuned on the dev set, and the performance on the testing set of the target datasets was reported as the final result. We use accuracy as the performance measurement.

5.2 Supervised Transfer Learning

Experimental Results

Table 2 reports the results of our transfer learning on TOEFL-manual, TOEFL-ASR, MC160, and MC500, as well as the performance of the previous best models and several ablations that did not use pre-training or fine-tuning. From Table 2, we have the following observations.

Model	Training	TOEFL		MCTest	
		manual	ASR	MC160	MC500
QACNN	(a) Target Only	48.9	47.5	57.5	56.4
	(b) Source Only	51.2	49.2	68.1	61.5
	(c) Source + Target	52.5	49.7	72.1	64.6
	(d) Fine-tuned (1)	53.4 (4.5)	51.5 (4.0)	**76.4** (18.9)	68.7 (12.3)
	(e) Fine-tuned (2)	**56.1** (7.2)	**55.3** (7.8)	73.8 (16.3)	**72.3** (15.9)
	(f) Fine-tuned (all)	56.0 (7.1)	55.1 (7.6)	69.3 (11.8)	67.7 (11.3)
MemN2N	(g) Target Only	45.2	44.4	57.2	53.6
	(h) Source Only	43.7	41.9	56.8	52.3
	(i) Source + Target	46.8	45.7	60.4	56.9
	(j) Fine-tuned	48.6 (3.4)	46.6 (2.2)	66.7 (9.5)	62.8 (9.2)
Fang et al. (2016)		49.1	48.8	-	-
Trischler et al. (2016)		-	-	74.6	71.0
Wang et al. (2015)		-	-	75.3	69.9

Table 2: Results of transfer learning on the target datasets. The number in the parenthesis indicates the accuracy increased via transfer learning (compared to rows (a) and (g)). The best performance for each target dataset is marked in bold. We also include the results of the previous best performing models on the target datasets in the last three rows.

Transfer learning helps. Rows (a) and (g) show the respective results when the QACNN and MemN2N are trained directly on the target datasets without pre-training on MovieQA. Rows (b) and (h) show results when the models are trained only on the MovieQA data. Rows (c) and (i) show results when the models are trained on both MovieQA and each of the four target datasets, and tested on the respective target dataset. We observe that the results achieved in (a), (b), (c), (g), (h), and (i) are worse than their fine-tuned counterparts (d), (e), (f), and (j). Through transfer learning, both QACNN and MemN2N perform better on all the target datasets. For example, QACNN only achieves 57.5% accuracy on MC160 without pre-training on MovieQA, but the accuracy increases by 18.9% with pre-training (rows (d) vs. (a)). In addition, with transfer learning, QACNN outperforms the previous best models on TOEFL-manual by 7%, TOEFL-ASR (Fang et al., 2016) by 6.5%, MC160 (Wang et al., 2015) by 1.1%, and MC500 (Trischler et al., 2016) by 1.3%, and becomes the state-of-the-art on all target datasets.

Which QACNN parameters to transfer? For the QACNN, the training parameters are $E, W_{CNN}^{(1)}, W_{CNN}^{(2)}, W_{FC}^{(1)}$, and $W_{FC}^{(2)}$ (Section 4.2). To better understand how transfer learning affects the performance of QACNN, we

also report the results of keeping some parameters fixed and only fine-tuning other parameters. We choose to fine-tune either only the last fully-connected layer $W_{FC}^{(2)}$ while keeping other parameters fixed (row (d) in Table 2), the last two fully-connected layers $W_{FC}^{(1)}$ and $W_{FC}^{(2)}$ (row (e)), and the entire QACNN (row (f)). For TOEFL-manual, TOEFL-ASR, and MC500, QACNN performs the best when only the last two fully-connected layers were fine-tuned; for MC160, it performs the best when only the last fully-connected layer was fine-tuned. Note that for training the QACNN, we followed the same procedure as in Liu et al. (2017), whereby pre-trained GloVe word vectors (Pennington et al., 2014) were used to initialize the embedding layer, which were not updated during training. Thus, the embedding layer does not depend on the training set, and the effective vocabularies are the same.

Fine-tuning the entire model is not always best. It is interesting to see that fine-tuning the entire QACNN doesn't necessarily produce the best result. For MC500, the accuracy of QACNN drops by 4.6% compared to just fine-tuning the last two fully-connected layers (rows (f) vs. (e)). We conjecture that this is due to the amount of training data of the target datasets - when the training set of the target dataset is too small, fine-tuning all the parameters of a complex model like QACNN may result in overfitting. This discovery aligns with other domains where transfer learning is well-studied such as object recognition (Yosinski et al., 2014).

A large quantity of mismatched training examples is better than a small training set. We expected to see that a MemN2N, when trained directly on the target dataset without pre-training on MovieQA, would outperform a MemN2N pre-trained on MovieQA without fine-tuning on the target dataset (rows (g) vs. (h)), since the model is evaluated on the target dataset. However, for the QACNN this is surprisingly not the case - QACNN pre-trained on MovieQA without fine-tuning on the target dataset outperforms QACNN trained directly on the target dataset without pre-training on MovieQA (rows (b) vs. (a)). We attribute this to the limited size of the target dataset and the complex structure of the QACNN.

Percentage of the target dataset used for fine-tuning	TOEFL		MCTest	
	manual	ASR	MC160	MC500
0	51.2	49.2	68.1	61.5
25%	53.9 (2.7)	52.3 (3.1)	70.3 (2.2)	65.6 (4.1)
50%	54.8 (0.9)	54.4 (2.1)	71.9 (1.6)	68.0 (2.4)
75%	55.3 (0.5)	54.8 (0.4)	72.5 (0.6)	71.1 (3.1)
100%	56.0 (0.7)	55.1 (0.3)	73.8 (1.3)	72.3 (1.2)

Table 3: Results of varying sizes of the target datasets used for fine-tuning QACNN. The number in the parenthesis indicates the accuracy increases from using the previous percentage for fine-tuning to the current percentage.

Varying the fine-tuning data size

We conducted experiments to study the relationship between the amount of training data from the target dataset for fine-tuning the model and the performance. We first pre-train the models on MovieQA, then vary the training data size of the target dataset used to fine-tune them. Note that for QACNN, we only fine-tune the last two fully-connected layers instead of the entire model, since doing so usually produces the best performance according to Table 2. The results are shown in Table 3[4]. As expected, the more training data is used for fine-tuning, the better the model's performance is. We also observe that the extent of improvement from using 0% to 25% of target training data is consistently larger than using from 25% to 50%, 50% to 75%, and 75% to 100%. Using the QACNN fine-tuned on TOEFL-manual as an example, the accuracy of the QACNN improves by 2.7% when varying the training size from 0% to 25%, but only improves by 0.9%, 0.5%, and 0.7% when varying the training size from 25% to 50%, 50% to 75%, and 75% to 100%, respectively.

Varying the pre-training data size

We also vary the size of MovieQA for pre-training to study how large the source dataset should be to make transfer learning feasible. The results are shown in Table 4. We find that even a small amount of source data can help. For example, by using only 25% of MovieQA for pre-training, the accuracy increases 6.3% on MC160. This is because 25% of MovieQA training set (2,462 examples) is still much larger than the MC160 training set (280 examples). As the size of the source dataset increases, the performance of QACNN continues to improve.

[4]We only include the results of QACNN in Table 3, but the results of MemN2N are very similar to QACNN.

Percentage of MovieQA used for pre-training	TOEFL		MCTest	
	manual	ASR	MC160	MC500
0	48.9	47.6	57.5	56.4
25%	51.7 (2.8)	50.7 (3.1)	63.8 (6.3)	62.4 (6.0)
50%	53.5 (1.8)	52.3 (1.6)	67.3 (3.5)	66.7 (4.3)
75%	54.8 (1.3)	54.6 (2.3)	71.2 (3.9)	70.2 (3.5)
100%	56.0 (1.2)	55.1 (0.5)	73.8 (2.6)	72.3 (2.1)

Table 4: Results of varying sizes of the MovieQA used for pre-training QACNN. The number in the parenthesis indicates the accuracy increases from using the previous percentage for pre-training to the current percentage.

Analysis of the Questions Types

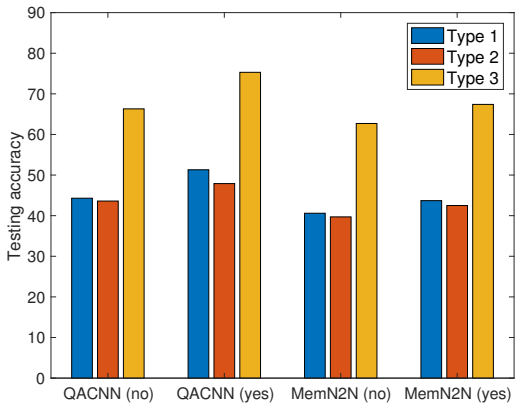
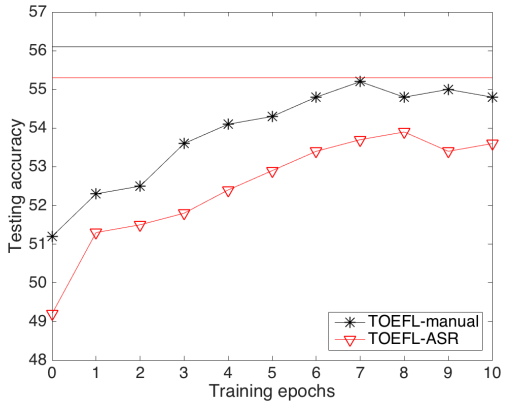


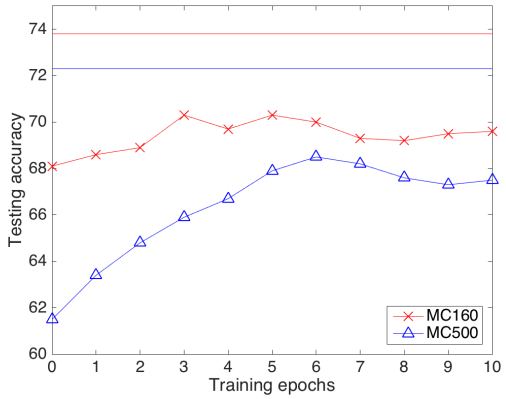
Figure 2: The performance of QACNN and MemN2N on different types of questions in TOEFL-manual with and without pre-training on MovieQA. 'No' in the parenthesis indicates the models are not pre-trained, while 'Yes' indicates the models are pre-trained on MovieQA.

We are interested in understanding what types of questions benefit the most from transfer learning. According to the official guide to the TOEFL test, the questions in TOEFL can be divided into 3 types. Type 1 questions are for basic comprehension of the story. Type 2 questions go beyond basic comprehension, but test the understanding of the functions of utterances or the attitude the speaker expresses. Type 3 questions further require the ability of making connections between different parts of the story, making inferences, drawing conclusions, or forming generalizations. We used the split provided by Fang et al. (2016), which contains 70/18/34 Type 1/2/3 questions. We compare the performance of the QACNN and MemN2N on different types of questions in TOEFL-manual with and without pre-training on MovieQA, and show the results in Figure 2. From Figure 2 we can observe that for both the QACNN and MemN2N, their performance on all three types of questions improves after pre-training, showing that the effectiveness of transfer learning is not limited to specific types of questions.

5.3 Unsupervised Transfer Learning



(a) Results of TOEFL-manual and TOEFL-ASR



(b) Results of MC160 and MC500

Figure 3: The figures show the results of unsupervised transfer learning. The x-axis is the number of training epochs, and the y-axis is the corresponding testing accuracy on the target dataset. When training epoch = 0, the performance of QACNN is equivalent to row (b) in Table 2. The horizontal lines, where each line has the same color to its unsupervised counterpart, are the performances of QACNN with supervised transfer learning (row (e) in Table 2), and are the upperbounds for unsupervised transfer learning.

So far, we have studied the property of supervised transfer learning for QA, which means

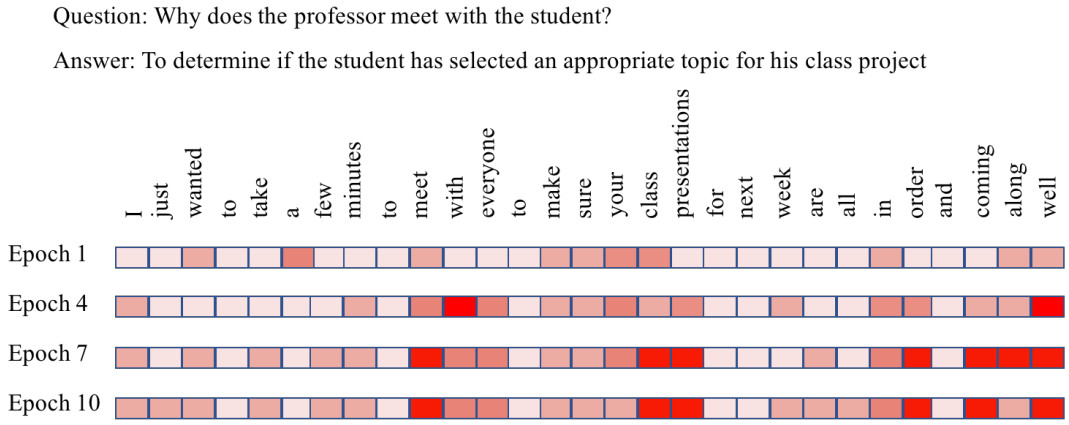


Figure 4: Visualization of the changes of the word-level attention map in the first stage CNN of QACNN in different training epochs. The more red, the more the QACNN views the word as a key feature. The input story-question-choices triplet is same as the one in Table 1.

that during pre-training and fine-tuning, both the source and target datasets provide the correct answer for each question. We now conduct unsupervised transfer learning experiments described in Section 2.2 (Algorithm 1), where the answers to the questions in the target dataset are not available. We used QACNN as the QA model and all the parameters ($E, W^{(1)}_{CNN}, W^{(2)}_{CNN}, W^{(1)}_{FC}$, and $W^{(2)}_{FC}$) were updated during fine-tuning in this experiment. Since the range of the testing accuracy of the TOEFL-series (TOEFL-manual and TOEFL-ASR) is different from that of MCTest (MC160 and MC500), their results are displayed separately in Figure 3(a) and Figure 3(b), respectively.

Experimental Results

From Figure 3(a) and Figure 3(b) we can observe that without ground truth in the target dataset for supervised fine-tuning, transfer learning from a source dataset can still improve the performance through a simple iterative self-labeling mechanism. For TOEFL-manual and TOEFL-ASR, QACNN achieves the highest testing accuracy at Epoch 7 and 8, outperforming its counterpart without fine-tuning by approximately 4% and 5%, respectively. For MC160 and MC500, the QACNN achieves the peak at Epoch 3 and 6, outperforming its counterpart without fine-tuning by about 2% and 6%, respectively. The results also show that the performance of unsupervised transfer learning is still worse than supervised transfer learning, which is not surprising, but the effectiveness of unsupervised transfer learning when no ground truth labels are provided is validated.

Attention Maps Visualization

To better understand the unsupervised transfer learning process of QACNN, we visualize the changes of the word-level attention map during training Epoch 1, 4, 7, and 10 in Figure 4. We use the same question from TOEFL-manual as shown in Table 1 as an example. From Figure 4 we can observe that as the training epochs increase, the QACNN focuses more on the context in the story that is related to the question and the correct answer choice. For example, the correct answer is related to "class project". In Epoch 1 and 4, the model does not focus on the phrase "class representation", but the model attends on the phrase in Epoch 7 and 10. This demonstrates that even without ground truth, the iterative process in Algorithm 1 is still able to lead the QA model to gradually focus more on the important part of the story for answering the question.

6 Conclusion and Future Work

In this paper we demonstrate that a simple transfer learning technique can be very useful for the task of multi-choice question answering. We use a QACNN and a MemN2N as QA models, with MovieQA as the source task and a TOEFL listening comprehension test and MCTest as the target tasks. By pre-training on MovieQA, the performance of both models on the target datasets improves significantly. The models also require much less training data from the target dataset to achieve similar performance to those without pre-training. We also conduct experiments to study the influence of transfer learning on different types

of questions, and show that the effectiveness of transfer learning is not limited to specific types of questions. Finally, we show that by a simple iterative self-labeling technique, transfer learning is still useful, even when the correct answers for target QA dataset examples are not available, through quantitative results and visual analysis.

One area of future research will be generalizing the transfer learning results presented in this paper to other QA models and datasets. In addition, since the original data format of the TOEFL listening comprehension test is audio instead of text, it is worth trying to initialize the embedding layer of the QACNN with semantic or acoustic word embeddings learned directly from speech (Chung and Glass, 2018, 2017; Chung et al., 2016) instead of those learned from text (Mikolov et al., 2013; Pennington et al., 2014).

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *ICCV*.

Langzhou Chen, Mark J. F. Gales, and K. K. Chin. 2011. Constrained discriminative mapping transforms for unsupervised speaker adaptation. In *ICASSP*.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *EMNLP*.

Yu-An Chung and James Glass. 2017. Learning word embeddings from speech. In *NIPS ML4Audio Workshop*.

Yu-An Chung and James Glass. 2018. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *CoRR* abs/1803.08976.

Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee. 2016. Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder. In *INTERSPEECH*.

Mortaza Doulaty, Oscar Saz, and Thomas Hain. 2015. Data-selective transfer learning for multi-domain speech recognition. In *INTERSPEECH*.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *CVPR*.

Wei Fang, Juei-Yang Hsu, Hung-Yi Lee, and Lin-Shan Lee. 2016. Hierarchical attention model for improved machine comprehension of spoken content. In *SLT*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*.

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine. In *EMNLP*.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *ICASSP*.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2016. From particular to general: A preliminary case study of transfer learning in reading comprehension. In *NIPS Machine Intelligence Workshop*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Hung-Yi Lee, Yu-Yu Chou, Yow-Bang Wang, and Lin-Shan Lee. 2013. Unsupervised domain adaptation for spoken document summarization with structured support vector machine. In *ICASSP*.

Tzu-Chien Liu, Yu-Hsueh Wu, and Hung-Yi Lee. 2017. Query-based attention CNN for text similarity map. In *ICCV Workshop*.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *NAACL HLT*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *ACL*.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *EMNLP*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPRW*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *NIPS*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. MovieQA: Understanding stories in movies through question-answering. In *CVPR*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *RepL4NLP*.

Adam Trischler, Zheng Ye, and Xingdi Yuan. 2016. A parallel-hierarchical model for machine comprehension on sparse data. In *ACL*.

Bo-Hsiang Tseng, Sheng-Syun Shen, Hung-Yi Lee, and Lin-Shan Lee. 2016. Towards machine comprehension of spoken content: Initial TOEFL listening comprehension test by machine. In *INTERSPEECH*.

Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. Technical report.

R. Wallace, Thambiratnam K., and F. Seide. 2009. Unsupervised speaker adaptation for telephone call transcription. In *ICASSP*.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *ACL*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards AI-complete question answering: A set of prerequisite toy tasks. In *ICLR*.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural domain adaptation for biomedical question answering. In *CoNLL*.

Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*.

Yi Yang, Wen-Tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *EMNLP*.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR*.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *NIPS*.

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2017. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics* 5:515–528.