

# Two-Stage Synthesis Networks for Transfer Learning in Machine Comprehension

David Golub\*

Stanford University

golubd@cs.stanford.edu

Po-Sen Huang and Xiaodong He

Microsoft Research

{pshuang, xiaohex}@microsoft.com

Li Deng†

Citadel Holdings, LLC

l.deng@ieee.org

## Abstract

We develop a technique for transfer learning in machine comprehension (MC) using a novel two-stage synthesis network (SynNet). Given a high-performing MC model in one domain, our technique aims to answer questions about documents in another domain, where we use no labeled data of question-answer pairs. Using the proposed SynNet with a pretrained model on the SQuAD dataset, we achieve an F1 measure of 46.6% on the challenging NewsQA dataset, approaching performance of in-domain models (F1 measure of 50.0%) and outperforming the out-of-domain baseline by 7.6%, without use of provided annotations.

## 1 Introduction

Machine comprehension (MC), the ability to answer questions over a provided context paragraph, is a key task in natural language processing. The rise of high-quality, large-scale human-annotated datasets for this task (Rajpurkar et al., 2016; Trischler et al., 2016) has allowed for the training of data-intensive but expressive models such as deep neural networks (Wang et al., 2016; Xiong et al., 2016; Seo et al., 2016). Moreover, these datasets have the attractive quality that the answer is a short snippet of text within the paragraph, which narrows the search space of possible answer spans.

However, many of these models rely on large amounts of human-labeled data for training. Yet data collection is a time-consuming and expensive task. Moreover, direct application of a MC model

trained on one domain to answer questions over paragraphs from another domain may suffer performance degradation.

While understudied, the ability to transfer a MC model to multiple domains is of great practical importance. For instance, the ability to quickly use a MC model trained on Wikipedia to bootstrap a question-answering system over customer support manuals or news articles, where there is no labeled data, can unlock a great number of practical applications.

In this paper, we address this problem in MC through a two-stage synthesis network (SynNet). The SynNet generates synthetic question-answer pairs over paragraphs in a new domain that are then used in place of human-generated annotations to finetune a MC model trained on the original domain.

The idea of generating synthetic data to augment insufficient training data has been explored before. For example, for the target task of translation, Sennrich et al. (2016) present a method to generate synthetic translations given real sentences to refine an existing machine translation system.

However, unlike machine translation, for tasks like MC, we need to synthesize both the question and answers given the context paragraph. Moreover, while the question is a syntactically fluent natural language sentence, the answer is mostly a salient semantic concept in the paragraph, e.g., a named entity, an action, or a number, which is often a single word or short phrase<sup>1</sup>. Since the answer has a very different linguistic structure compared to the question, it may be more appropriate to view answers and questions as two different types of data. Hence, the synthesis of a (question,

\* Work performed while interning at Microsoft Research.

† Work performed at Microsoft Research.

<sup>1</sup>This assumption holds for MC datasets such as SQuAD and NewsQA, but there are exceptions in certain subdomains of MSMARCO.

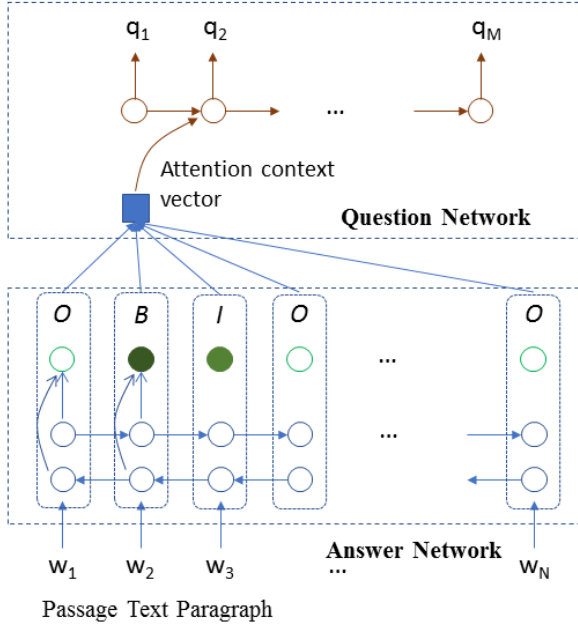


Figure 1: Illustration of the two-stage SynNet. The SynNet is trained to synthesize the answer and the question, given the paragraph. The first stage of the model, an answer synthesis module, uses a bi-directional LSTM to predict IOB tags on the input paragraph, which mark out key semantic concepts that are likely answers. The second stage, a question synthesis module, uses a uni-directional LSTM to generate the question, while attending on embeddings of the words in the paragraph and IOB ids. Although multiple spans in the paragraph could be identified as potential answers, we pick one span when generating the question.

answer) tuple is needed.

In our approach, we decompose the process of generating question-answer pairs into two steps, answer generation conditioned on the paragraph, and question generation conditioned on the paragraph and answer. We generate the answer first because answers are usually key semantic concepts, while questions can be viewed as a full sentence composed to inquire the concept.

Using the proposed SynNet, we are able to outperform a strong baseline of directly applying a high-performing MC model trained on another domain. For example, when we apply our algorithm using a pretrained model on the Stanford Question-Answering Dataset (SQuAD) (Rajpurkar et al., 2016), which consists of Wikipedia articles, to answer questions on the NewsQA dataset (Trischler et al., 2016), which consists of CNN/Daily Mail articles, we improve the performance of the SQuAD baseline from 39.0%

to 46.6% F1 and approach results of previously published work of Trischler et al. (2016) (50.0% F1), without use of labeled data in the new domain. Moreover, an error analysis reveals that we achieve higher accuracy over the baseline on all common question types.

## 2 Related Work

### 2.1 Question Answering

Question answering is an active area in natural language processing with ongoing research in many directions (Berant et al., 2013; Hill et al., 2015; Golub and He, 2016; Chen et al., 2016; Hermann et al., 2015). Machine comprehension, a form of extractive question answering where the answer is a snippet or multiple snippets of text within a context paragraph, has recently attracted a lot of attention in the community. The rise of large-scale human annotated datasets with over 100,000 realistic question-answer pairs such as SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), and MSMARCO (Nguyen et al., 2016), has led to a large number of successful deep learning models (Lee et al., 2016; Seo et al., 2016; Xiong et al., 2016; Dhingra et al., 2016; Wang and Jiang, 2016).

### 2.2 Semi-Supervised Learning

Semi-supervised learning has a long history (c.f. Chapelle et al. (2009) for an overview), and has been applied to many tasks in natural language processing such as dependency parsing (Koo et al., 2008), sentiment analysis (Yang et al., 2015), machine translation (Sennrich et al., 2016), and semantic parsing (Berant and Liang; Wang et al.; Jia and Liang, 2016). Recent work generated synthetic annotations on unsupervised data to boost the performance of both reading comprehension and visual question answering models (Yang et al., 2017; Ren et al., 2015), but on domains with some form of annotated data. There has also been work on generating high-quality questions (Yuan et al., 2017; Serban et al., 2016; Labutov et al.), but not how to best use them to train a model. In contrast, we use the two-stage SynNet to generate data tuples to directly boost performance of a model on a domain with no annotations.

### 2.3 Transfer Learning

Transfer learning (Pan and Yang, 2010) has been successfully applied to numerous domains in machine learning, such as machine translation (Zoph

et al., 2016), computer vision, (Sharif Razavian et al., 2014), and speech recognition (Doulaty et al., 2015). Specifically, object recognition models trained on the large-scale ImageNet challenge (Russakovsky et al., 2015) have proven to be excellent feature extractors for diverse tasks such as image captioning (i.e., Lu et al. (2016); Fang et al. (2015); Karpathy and Fei-Fei (2015)) and visual question answering (i.e., Zhou et al. (2015); Xu and Saenko (2016); Fukui et al. (2016); Yang et al. (2016)), among others. In a similar fashion, we use a model pretrained on the SQuAD dataset as a generic feature extractor to bootstrap a QA system on NewsQA.

### 3 The Transfer Learning Task for MC

We formalize the task of machine comprehension below. Our MC model takes as input a tokenized question  $q = \{q_0, q_1, \dots, q_n\}$ , a context paragraph  $p = \{p_0, p_1, \dots, p_n\}$ , where  $q_i, p_i$  are words, and learns a function  $f(p, q) \mapsto \{a_{start}, a_{end}\}$  where  $a_{start}$  and  $a_{end}$  are pointer indices into paragraph  $p$ , i.e., the answer  $a = p_{a_{start}} \dots p_{a_{end}}$ .

Given a collection of labeled paragraph, question, answer triples  $\{p, q, a\}_{i=1}^n$  from a particular domain  $s$ , i.e., Wikipedia articles, we can learn a MC model  $f_s(p, q)$  that is able to answer questions in that domain.

However, when applying the model trained in one domain to answer questions in another, the performance may degrade. On the other hand, labeling data to train a model in the new domain is expensive and time-consuming.

In this paper, we propose the task of transferring a MC system  $f_s(p, q)$  that is trained in a source domain to answer questions over another target domain,  $t$ . In the target domain  $t$ , we are given an unlabeled set  $p_t = \{p\}_{i=1}^k$  of  $k$  paragraphs. During test time, we are given an unseen set of paragraphs,  $p^*$ , in the target domain, over which we would like to answer questions.

## 4 The Model

### 4.1 Two-Stage SynNet

To bootstrap our model  $f_s$  we use a SynNet (Figure 1), which consists of answer synthesis and question synthesis modules, to generate data on  $p_t$ . Our SynNet learns the conditional probability of generating answer  $a = \{a_{start}, a_{end}\}$  and question  $q = \{q_1, \dots, q_n\}$  given paragraph  $p$ ,  $P(q, a|p)$ . We decompose the joint probability distribution

$P(q, a|p)$  into a conditional probability distribution  $P(q|p, a)P(a|p)$ , where we first generate the answer  $a$ , followed by generating the question  $q$  conditioned on the answer and paragraph.

#### 4.1.1 Answer Synthesis Module

In our answer synthesis module we train a simple IOB tagger to predict whether each word in the paragraph is part of an answer or not.

More formally, given a set of words in a paragraph  $p = \{p_1 \dots p_n\}$ , our IOB tagging model learns the conditional probability of labels  $y_1 \dots y_n$ , where  $y_i \in \text{IOB}_{\text{START}}, \text{IOB}_{\text{MID}}, \text{IOB}_{\text{END}}$  if a word  $p_i$  is marked as an answer by the annotator in our train set, NONE otherwise.

We use a bi-directional Long-Short Term Memory Network (Bi-LSTM) (Hochreiter and Schmidhuber, 1997) for tagging. Specifically, we project each word  $p_i \mapsto p_i^*$  into a continuous vector space via pretrained GloVe embeddings (Pennington et al., 2014). We then run a Bi-LSTM over the word embeddings  $p_1^*, \dots, p_n^*$  to produce a context-dependent word representation  $h_1, \dots, h_n$ , which we feed into two fully connected layers followed by a softmax to produce our tag likelihoods for each word.

We select all consecutive spans where  $y \neq \text{NONE}$  produced by the tagger as our candidate answer chunks, which we feed into our question synthesis module for question generation.

#### 4.1.2 Question Synthesis Module

Our question synthesis module learns the conditional probability of generating question  $q = \{q_1, \dots, q_n\}$  given answer  $a = a_{start}, a_{end}$  and paragraph  $p = p_1 \dots p_n$ ,  $P(q_1, \dots, q_n | p_1 \dots p_n, a_{start}, a_{end})$ . We decompose the joint probability distribution of generating all the question words  $q_1, \dots, q_n$  into generating the question one word at a time, i.e.  $\prod_{i=1}^n P(q_i | p, a, q_{1 \dots i-1})$ .

The model is similar to an encoder-decoder network with attention (Bahdanau et al., 2014), which computes the conditional probability  $P(q_i | p_1 \dots p_n, a_{start}, a_{end}, q_{1 \dots i-1})$ . We run a Bi-LSTM over the paragraph to produce context-dependent word representations  $h = \{h_1, \dots, h_n\}$ . To model where the answer is in the paragraph, similar to Yang et al. (2017), we insert answer information by appending a zero/one feature to the paragraph word embeddings. Then, at each time step  $i$ , a decoder network attends to

both  $h$  and the previously generated question token  $q_{i-1}$  to produce a hidden representation  $r_i$ . Since paragraphs may often have named entities and rare words not present during training, we incorporate a copy mechanism into our models (Gu et al., 2016).

We use an architecture motivated by latent predictor networks (Ling et al., 2016) to force the model to learn when to copy vs. directly predict the word, without direct supervision of what action to choose. Specifically, at every time step  $i$ , two latent predictors generate the probability of generating word  $w_i$ , a pointer network  $C_p$  (Vinyals et al., 2015) which can copy a word from the context paragraph, and a vocabulary predictor  $V_p$  which directly generates a probability distribution of choosing a word  $w_i$  from a predefined vocabulary. The likelihood of choosing predictor  $k$  at time step  $i$  is proportional to  $w_k r_i$ , and the likelihood of predicting question token  $q_i$  is given by  $q_i^* = p^v l^v(w_i) + (1 - p^v) l^c(w_i)$ , where  $v$  represents the vocabulary predictor and  $c$  represents the copy predictor, and  $l(w_i)$  is the likelihood of the word given by the predictor<sup>2</sup>. For training, since no direct supervision is given as to which predictor to choose, we minimize the cross entropy loss of producing the correct question tokens  $\sum_{j=1}^n -\log(q_j^*)$  by marginalizing out latent variables using a variant of the forward-backward algorithm (see Ling et al. (2016) for full details).

During inference, to generate a question  $q_1 \dots q_n$ , we use greedy decoding in the following manner. At time step  $i$ , we select the most likely predictor ( $C_p$  or  $V_p$ ), followed by the most likely word  $q_i$  given the predictor. We feed the predicted word as input at the next timestep back into the decoder until we predict the end symbol, END, after which we stop decoding.

## 4.2 Machine Comprehension Model

Our machine comprehension model  $f(p, q) \mapsto a$  learns the conditional likelihood of predicting answer pointers  $a = \{a_{start}, a_{end}\}$  given paragraph  $p$  and question  $q$ ,  $P(a|p, q)$ . In our experiments we use the open-source Bi-directional Attention Flow (BiDAF) network (Seo et al., 2016)<sup>3</sup> since it is one of the best-performing models on the SQuAD dataset<sup>4</sup>, although we note that our algorithm for

<sup>2</sup>Since we only have two predictors,  $p^c = 1 - p^v$

<sup>3</sup>See <https://github.com/allenai/bi-att-flow>

<sup>4</sup>See <https://rajpurkar.github.io/SQuAD-explorer/> for latest results

---

## Algorithm 1: Training Algorithm

---

**Input** :  $x_s = \{p_s, q_s, a_s\}_{i=1}^n$  triplets from source domain  $s$ ; pretrained MC model on  $s$ ,  
 $f_s(p, q) \mapsto \{a_{start}, a_{end}\}$ ;  
 paragraphs from target domain  $t$ ,  
 $p_{j=1}^m$

**Output**: MC model on target domain,  
 $f_t(p, q) \mapsto \{a_{start}, a_{end}\}$

- 1 Train SynNet  $g$  to maximize  $P(q, a|p)$  on source  $s$ ;
  - 2 Generate samples  $x_t = (q, a|p)_{i=1}^k$  on text in target domain  $t$ ;
  - 3 Use  $x_s \cup x_t$  to finetune MC model  $f_s$  on domain  $t$ . For every batch sampled from  $x_t$ , sample  $k$  batches from  $x_s$ ;
- 

data synthesis can be used with any MC model.

## 4.3 Algorithm Overview

Having given an overview of our SynNet and a brief overview of the MC model we describe our training procedure, which is illustrated in Algorithm 1.

## 4.4 Training

Our approach for transfer learning consists of several training steps. First, given a series of labeled examples  $x_s = \{p_s, q_s, a_s\}_{i=1}^n$  from domain  $s$ , paragraphs  $p_{j=1}^m$  from domain  $t$ , and pretrained MC model  $f_s(p, q)$ , we train the SynNet  $g_s$  to maximize the likelihood of the question-answer pairs in  $s$ .

Second, we fix our SynNet  $g_s$  and we sample  $x_t = \{p_t, q_t, a_t\}_{i=1}^k$  question-answer pairs on the paragraphs in domain  $t$ . Several examples of generated questions can be found in Table 1.

We then transfer the MC model originally learned on the source domain to the target domain  $t$  using SGD on the synthetic data. However, since the synthetic data is usually noisy, we alternatively train the MC model with mini-batches from  $x_s$  and  $x_t$ , which we call data-regularization. Every  $k$  batches from  $x_t$ , we sample 1 batch of synthetic data from  $x_s'$ , where  $k$  is a hyper-parameter, which we set to 4. Letting the model encounter many examples from source domain  $s$  serves to regularize the distribution of the synthetic data in the target domain with real data from  $s$ . We checkpoint finetuned model  $f_s^*$  every  $i$  mini-batches,  $i = 1000$  in



Snippet of context paragraph (answer in bold)	Generated questions (bold) vs. human questions
...A high court in northern India on Friday acquitted a wealthy businessman facing the death sentence for the killing of a teen in a case dubbed "the house of horrors". "Moninder Singh Pandher was sentenced to death by a lower court in February. The teen was <b>one of 19</b> victims – children and ...	<b>How many victims were in India ?</b> What was the amount of children murdered ?
...Rescuers have found the body of a man who was one of six people aboard a small airplane that crashed Sunday evening near the northern shore of <b>Puerto Rico</b> , the U.S. Coast Guard said. The Cessna 206 single-engine aircraft ...	<b>What was the body of Puerto Rico's airplane?</b> Where did the diver find the body?
...Shopping malls around the country were expected to review their emergency plans and consider additional security measures in light of Wednesday's shooting, which killed <b>eight</b> . Watch what experts say about keeping malls safe...	<b>How many experts died in the International Institute of Shopping War ?</b> How many died in mall shooting ?
...Former boxing champion Vernon Forrest, 38, was shot and killed in <b>southwest Atlanta, Georgia</b> , on July 25. A grand jury indicted the three suspects – Charman Sinkfield, 30; Demario Ware, 20; and Jquante...	<b>Where was the first person to be shot ?</b> Where was Forrest killed?

Table 1: Randomly sampled paragraphs and corresponding synthetic vs. human questions from the NewsQA train set.

our experiments, and save a copy of the model at each checkpoint.

At test time, to generate an answer, we feed paragraph  $p = \{p_0, p_1, \dots, p_n\}$  and question  $q$  through our finetuned MC model  $f^*(p, q)$  to get  $P(p_i = a_{start})$ ,  $P(p_i = a_{end})$  for all  $i \in 1 \dots n$ . We then use dynamic programming (Seo et al., 2016) to find the optimal answer span  $\{a_{start}, a_{end}\}$ . To improve the stability of using our model for inference, we average the predicted answer likelihoods from model copies at different checkpoints prior to running the dynamic programming algorithm.

## 5 Experimental Setup

We summarize the datasets we use in our experiments, parameters for our model architectures, and training details.

The SQuAD dataset consists of approximately 100,000 question-answer pairs on Wikipedia, 87,600 of which are used for training, 10,570 for development, and an unknown number in a hidden test set. The NewsQA dataset consists of 92,549 train, 5,166 development and 5,165 test questions on CNN/Daily Mail news articles. Both the domain type (i.e., news) and question types differ between the two datasets. For example, an analysis of a randomly generated sample of 1,000 questions from both NewsQA and SQuAD (Trischler et al., 2016) reveals that approximately 74.1% of questions in SQuAD require word matching or

paraphrasing to retrieve the answer, as opposed to 59.7% in NewsQA. As our test metrics, we report two numbers, exact match (EM) and F1 score.

We train a BIDAf model on the SQuAD train dataset and use a two-stage SynNet to finetune it on the NewsQA train dataset.

We initialize word-embeddings for the BIDAf model, answer synthesis module, and question synthesis module with 300-dimensional-GloVe vectors (Pennington et al., 2014) trained on the 840B Common Crawl corpus. We set all embeddings of unknown word tokens to zero.

For both the answer synthesis and question synthesis module, we use a vocabulary of size 110,179. We use LSTMs with hidden states of size 150 for the answer module vs. those of size 100 for the question module since the answer module is less memory intensive than the question module.

We train both the answer and question module with Adam (Kingma and Ba, 2014) and a learning rate of  $1e-2$ . We train a BIDAf model with the default hyperparameters provided in the open-source repository. To stop training of the question synthesis module, after each epoch, we monitor both the loss as well as the quality of questions generated on the SQuAD development set. To stop training of the answer synthesis module, we similarly monitor predictions on the SQuAD development set.

To train the question synthesis module, we only use the questions provided in the SQuAD train set. However, to train the answer synthesis module,

Method	System	EM	F1
Transfer Learning	$M_{sq}$ (baseline)	24.9	39.0
	$M_{sq} + A_{gen} + Q_{gen}$	30.6	44.3
	$M_{sq} + A_{gen} + A_{ner} + Q_{gen}$	32.8	46.6
	$M_{sq} + A_{gen} + A_{ner} + Q_{gen} + M_{sq}^*$	<b>33.0</b>	<b>46.6</b>
Supervised Learning	Barb-LSTM on NewsQA (Trischler et al., 2016)	34.9	50.0
	Match-LSTM on NewsQA (Trischler et al., 2016)	34.1	48.2
	BIDAF on NewsQA	37.1	52.3
	BIDAF on SQuAD finetuned on NewsQA	37.3	52.2

Table 2: **Main Results.** Exact match (EM) and span F1 scores on the NewsQA test set of a BIDAF model finetuned with our SynNet.  $M_{sq}$  refers to a baseline BIDAF model trained on SQuAD,  $A_{gen}$ ,  $Q_{gen}$  refers to using answers generated from our SynNet respectively to finetune the model on NewsQA,  $A_{ner}$  refers to using answers extracted from a standard NER system to generate questions.  $M_{sq}^*$  refers to using the baseline SQuAD model in the ensemble.

	EM	F1
$M_{newsqa}$	46.3	60.8
$M_{newsqa} + S_{net}$	47.9	61.5

Table 3: **NewsQA to SQuAD.** Exact match (EM) and span F1 results on SQuAD development set of a NewsQA BIDAF model baseline vs. one finetuned on SQuAD using the data generated by a 2-stage SynNet ( $S_{net}$ ).

we further augment the human-annotated labels of each paragraph with tags from a simple NER system<sup>5</sup> because labels of answers provided in the train set are underspecified, i.e., many words in the paragraph that could be potential answers are not labeled. Therefore, we assume any named entities could also be potential answers of certain questions, in addition to the answers explicitly labeled by annotators.

To generate question-answer pairs on the NewsQA train set using the SynNet, we first run every paragraph through our answer synthesis module. We then randomly sample up to 30 candidate answers extracted by our module, which we feed into the question synthesis module. This results in 250,000 synthetic question-answer pairs that we can use to finetune our MC model.

<sup>5</sup><https://spacy.io/>

A)	EM	F1	B)	EM	F1
k=0	27.2	40.5	2 sent + $A_{ner}$	22.8	36.1
k=2	29.8	43.9	context + $A_{ner}$	27.2	40.5
k=4	30.4	44.3	2 sent + $A_{oracle}$	31.3	45.2
			context + $A_{oracle}$	32.5	46.8

Table 4: **Ablation Studies.** Exact match (EM) and span F1 results on NewsQA test set of a BIDAF model finetuned with a 2-stage SynNet. In study A, we vary  $k$ , the number of mini-batches from SQuAD for every batch in NewsQA. In study B, we set  $k = 0$ , and vary the answer type and how much of the paragraph we use for question synthesis. 2-sent refers to using two sentences before answer span, while context refers to using the entire paragraph.  $A_{ner}$  refers to using an NER system and  $A_{oracle}$  refers to using the human-annotated answers to generate questions.

## 6 Experimental Results

We report the main results on the NewsQA test set (Table 2), report brief results on SQuAD (Table 3), conduct ablation studies (Table 4), and conduct an error analysis.

### 6.1 Results

We compare to the best previously published work, which trains BARB (Trischler et al., 2016) and Match-LSTM (Wang and Jiang, 2016) architectures, and a BIDAF model we train on NewsQA. Directly applying a BIDAF model

trained on SQuAD to predict on NewsQA leads to poor performance with an F1 measure of 39.0%, 13.2% lower than one trained on labeled NewsQA data. Using the 2-stage SynNet already leads to a slight boost in performance (F1 measure of 44.3%), which implies that having exposure to the new domain via question-answer pairs provides important signal for the model during training. When we augment the answers from our answer synthesis module with those from a generic NER system to produce questions, we have an additional 2.3% performance boost. Finally, when we ensemble with the original model, we boost the EM further by 0.2%. Our final system achieves an F1 measure of 46.6%, approaching previously published results of 50.0%. The results demonstrate that using the proposed architecture and training procedure, we can transfer a MC model from one domain to another, without use of annotated data.

We also evaluate the SynNet on the NewsQA-to-SQuAD direction. We directly apply the best setting from the other direction and report the result in Table 3. The SynNet improves over the baseline by 1.6% in EM and 0.7% in F1. Limited by space, we leave out ablation studies in this direction.

## 6.2 Ablation Studies

To better understand how various components in our training procedure and model impact overall performance we conduct several ablation studies, as summarized in Table 4.

### 6.2.1 Answer Synthesis

We experiment with using the answer chunks given in the train set,  $A_{oracle}$ , to generate synthetic questions, versus those from an NER system,  $A_{ner}$ . Results in Table 4(A) show that using human-annotated answers to generate questions leads to a significant performance boost over using answers from an answer generation module. This supports the hypothesis that the answers humans choose to generate questions for provide important linguistic cues for finetuning the machine comprehension model.

### 6.2.2 Question Synthesis

To see how copying impacts performance, we explore using the entire paragraph to generate the question vs. only the two sentences before and one sentence after the answer span and report re-

sults in Table 4(B). On the NewsQA train set, synthetic questions that use 2 sentences contain an average of 3.0 context words within 10 words to the left and right of the answer chunk, those that use the entire context have 2.1 context words, and human generated questions only have 1.7 words. Training with generated questions that have a large amount of overlap with words close to the answer span (i.e., those that use 2-sentences vs. entire context for generation) leads to models that perform worse, especially with synthetic answer spans and no data regularization (35.6% F1 vs. 34.3% F1). One possible reason is that, according to analysis in Trischler et al. (2016), significantly more questions in the NewsQA dataset require paraphrase, inference, and synthesis as opposed to word-matching.

### 6.2.3 Model Finetuning

To see how the quantity of synthetic questions encountered during training impacts performance, we use  $k = \{0, 2, 4\}$  mini-batches from SQuAD for every synthetic mini-batch from NewsQA to finetune our model, and average the prediction of 4 checkpointed models during testing. As we see from the results, letting the model to encounter data from human annotations, although from another domain, serves as a key form of data-regularization, yielding consistent improvement as  $k$  increases. We hypothesize this is because the data distribution of machine-generated questions is different than human-annotated ones; our batching scheme provides a simple way to prevent overfitting to this distribution.

## 6.3 Error Analysis

In this section we provide a qualitative analysis of some of our components to help guide further research in this task.

### 6.3.1 Answer Synthesis

We randomly sample and present a paragraph with answers extracted by our answer synthesis module (Tables 5 and 6). Although the module appears to have high precision, i.e., it picks up entities such as the “Atlantic Paranormal Society”, it misses clear entities such as “David Schrader”, which suggests training a system with full NER/POS tags as labels would yield better results, and also explains why augmenting synthetic data generated by SynNet with such tags leads to improved performance.

They are ghost hunters , or , as they prefer to be called , paranormal investigators . “ Ghost-Hunters ”, which airs a special live show at **7 p.m. Halloween night** , is helping lift the stigma once attached to paranormal investigators . The show has become so popular that the group featured in each episode – **Atlantic Paranormal Society** - has spawned imitators across **United States** and affiliates in **countries** . TAPS , as the “**Hunters**” group is informally known , even has its own “**Reality Radio**” show , magazine , lecture tours , T-shirts – and groupies . “**Hunters**” has made creepy cool , says David Schrader , a paranormal investigator and co-host of “**Radio** ”, a radio show that investigates paranormal activity.

Table 5: Sample predictions from our answer synthesis module.

What is Oklahoma ’s unemployment rate until Oklahoma City ?
What was the manager of the Oklahoma City agency ?
How many companies are in Oklahoma City ?
How many workers may Oklahoma have as fair hold ?
Who said the bureau has already hired civilians to choose
What was the average hour manager of Oklahoma City ?
How much would Oklahoma have a year to be held
What year did Oklahoma ’s census build job industry ?

Table 6: Predictions from the question synthesis module on a subset of a paragraph.

### 6.3.2 Question Synthesis

We randomly sample synthetic questions generated by our module and present our results in Table 6. Due to the copy mechanism, our module has the tendency to directly use many words from the paragraph, especially common entities, such as “Oklahoma” in the example. Thus, one way to generate higher-quality questions may be to introduce a cost function that promotes diversity during decoding, especially within a single paragraph. In turn, this would expose the RC model to a larger variety of training examples in the new domain, which can lead to better performance.

### 6.3.3 Machine Comprehension Model

We examine the performance over various question types of a finetuned BIDAf on NewsQA vs. one trained on NewsQA vs. one trained

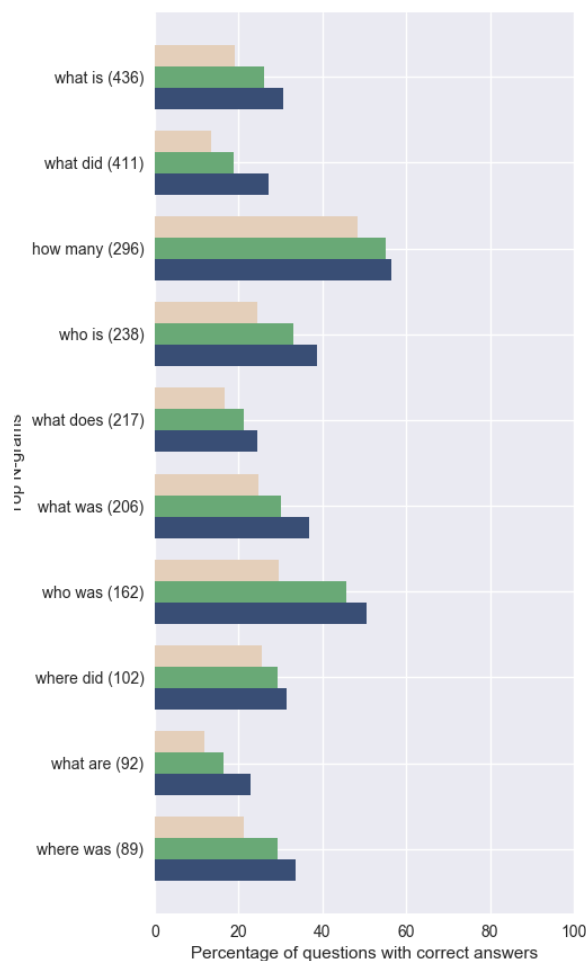


Figure 2: NewsQA accuracy of baseline BIDAf model trained on SQuAD (light yellow), vs. model finetuned with our method (green) vs. one trained from scratch on NewsQA (dark blue).

on SQuAD (Figure 2). Finetuning with SynNet improves performance over all question types given, with the largest performance boost on location and person-identification questions. Similarly, models trained on synthetic questions tend to approach in-domain performance on numeric and person-identification questions, but still struggle with questions that require higher-order reasoning, i.e. those starting with “what was” or “what did”. Designing a question generator that explicitly requires such reasoning may be one way to further bridge the gap in performance.

## 7 Conclusion

We introduce a two-stage SynNet for the task of transfer learning for machine comprehension, a task which is both challenging and of practical im-



portance. With our network and a simple training algorithm where we generate synthetic question-answer pairs on the target domain, we are able to generalize a MC model from one domain to another with no annotated data. We present strong results on the NewsQA test set, improving performance of a baseline BIDAf model by over 7.6% F1. Through ablation studies and error analysis, we provide insights into our methodology on the SynNet and MC models that can help guide further research in this task.

## Acknowledgments

We would like to thank Yejin Choi and Luke Zettlemoyer for their helpful comments and discussions concerning this work.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6.
- Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing.
- Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks* 20(3):542–542.
- 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*.
- Bhuwan Dhingra, Hanxiao Liu, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*.
- Mortaza Doulaty, Oscar Saz, and Thomas Hain. 2015. Data-selective transfer learning for multi-domain speech recognition. *arXiv preprint arXiv:1509.02409*.
- 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482.
- 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. *arXiv preprint arXiv:1606.01847*.
- David Golub and Xiaodong He. 2016. Character-level question answering with attention. *arXiv preprint arXiv:1604.00727*.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.
- Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children’s books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.
- Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Terry Koo, Xavier Carreras Pérez, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *46th Annual Meeting of the Association for Computational Linguistics*, pages 595–603.
- Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2014. Deep questions without deep understanding.
- Kenton Lee, Tom Kwiatkowski, Ankur Parikh, and Dipanjan Das. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.
- Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Andrew Senior, Fumin Wang, and Phil Blunsom. 2016. Latent predictor networks for code generation. *arXiv preprint arXiv:1603.06744*.

- Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2016. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *arXiv preprint arXiv:1612.01887*.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.
- 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 *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. <http://www.aclweb.org/anthology/D14-1162>.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*. pages 2953–2961.
- 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.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 86–96. <http://www.aclweb.org/anthology/P16-1009>.
- Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603.
- Iulian Vlad Serban, Alberto García-Durán, Çağlar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807*.
- Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pages 806–813.
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*. pages 2692–2700.
- Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match- lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.
- Yushi Wang, Jonathan Berant, Percy Liang, et al. ??? Building a semantic parser overnight.
- Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*.
- Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.
- Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*. Springer, pages 451–466.
- Min Yang, Wenting Tu, Ziyu Lu, Wenpeng Yin, and Kam-Pui Chow. 2015. Lcct: a semisupervised model for sentiment classification. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*. Association for Computational Linguistics (ACL).
- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*.
- Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 21–29.
- Xingdi Yuan, Tong Wang, Çağlar Gülçehre, Alessandro Sordoni, Philip Bachman, Sandeep Subramanian, Saizheng Zhang, and Adam Trischler. 2017. [Machine comprehension by text-to-text neural question generation](#). *CoRR* abs/1705.02012. <http://arxiv.org/abs/1705.02012>.
- Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2015. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*.
- Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.