MacNet: Transferring Knowledge from Machine Comprehension to Sequence-to-Sequence Models

Boyuan Pan[†], **Yazheng Yang**[‡], **Hao Li**[†], **Zhou Zhao**[‡], **Yueting Zhuang**[‡], **Deng Cai**[†♯], [⋆], **Xiaofei He**[⋆†]
[†]State Key Lab of CAD&CG, Zhejiang University
[‡]College of Computer Science, Zhejiang University
[♯]Alibaba-Zhejiang University Joint Institute of Frontier Technologies
[⋆]Fabu Inc., Hangzhou, China
{panby, yazheng_yang, haolics, zhaozhou, yzhuang, dcai}@zju.edu.cn
xiaofeihe@fabu.ai

Abstract

Machine Comprehension (MC) is one of the core problems in natural language processing, requiring both understanding of the natural language and knowledge about the world. Rapid progress has been made since the release of several benchmark datasets, and recently the state-of-the-art models even surpass human performance on the well-known SQuAD evaluation. In this paper, we transfer knowledge learned from machine comprehension to the sequence-to-sequence tasks to deepen the understanding of the text. We propose *MacNet*: a novel encoder-decoder supplementary architecture to the widely used attention-based sequence-to-sequence models. Experiments on neural machine translation (NMT) and abstractive text summarization show that our proposed framework can significantly improve the performance of the baseline models, and our method for the abstractive text summarization achieves the state-of-the-art results on the *Gigaword* dataset.

1 Introduction

Machine comprehension (MC) has gained significant popularity over the past few years and it is a coveted goal in the field of natural language understanding. Its task is to teach the machine to understand the content of a given passage and then answer a related question, which requires deep comprehension and accurate information extraction towards the text. With the release of several high-quality benchmark datasets [Hermann et al., 2015; Rajpurkar et al., 2016; Joshi et al., 2017], end-to-end neural networks [Wang et al., 2017; Xiong et al., 2017; Cui et al., 2017] have achieved promising results on the MC tasks and some even outperform humans on the SQuAD [Rajpurkar et al., 2016], which is one of the most popular machine comprehension tests. Table 1 shows a simple example from the SQuAD dataset.

Sequence-to-sequence (seq2seq) models [Sutskever et al., 2014] with attention mechanism [Bahdanau et al., 2015], in which an encoder compresses the source text and a decoder with an attention mechanism generates target words, have shown great capability to handle many natural language generation tasks such as machine translation [Luong et al., 2015; Xia et al., 2017], text summarization [Rush et al., 2015; Nallapati et al., 2016] and dialogue systems [Williams et al., 2017], *etc*. However, these encoder-decoder networks directly map the source input to a fixed target sentence to learn the relationship between the natural language texts, which makes them hard to capture a lot of deep intrinsic details and understand the potential implication of them [Li et al., 2017; Shi et al., 2016].

[⋆]corresponding author

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Passage: This was the first Super Bowl to feature a quarterback on both teams who was the #1 pick in their draft classes. **Manning was the #1 selection of the 1998 NFL draft**, while Newton was picked first in 2011. The matchup also pits the top two picks of the 2011 draft against each other: Newton for Carolina and Von Miller for Denver.
Question: Who was considered to be the first choice in the NFL draft of 1998?
Answer: Manning

Table 1: An example from the SQuAD dataset.

Inspired by the recent success of the approaches for the machine comprehension tasks, we focus on exploring whether MC knowledge can further help the attention-based seq2seq models deeply comprehend the text. Machine comprehension requires to encode words from the passage and the question firstly, then many methods [Seo et al., 2017; Wang et al., 2016; Xiong et al., 2018] employ attention mechanism with an RNN-based modeling layer to capture the interaction among the passage words conditioned on the question and finally use an MLP classifier or pointer networks [Vinyals et al., 2015] to predict the answer span. The MC-encoder mentioned above is a common component in the seq2seq models, while the RNN-based modeling layer whose input is the attention vectors is also supposed to augment the performance of the outputs of the seq2seq models. Intuitively, MC knowledge could improve seq2seq models through measuring the relevance between the generated sentence and the input source. Moreover, while quesiton answering and text generation have different training data distributions, they can still benefit from sharing their model's high-level semantic components [Guo, Pasunuru, and Bansal, 2018].

In this paper, we propose *MacNet*, a machine comprehension augmented encoder-decoder supplementary architecture that can be applied to a variety of sequence generation tasks. We begin by pre-training an MC model that contains both the RNN-based encoding layer and modeling layer as the transferring source. In the sequence-to-sequence model, for encoding, we concatenate the outputs of the original encoder and the transferred MC encoder; for decoding, we first input the attentional vectors from the seq2seq model into the transferred MC modeling layer, and then combine its outputs with the attentional vectors to formulate the predictive vectors. Moreover, to solve the class imbalance resulted by the high-frequency phrases, we adopt the *focal loss* [Lin et al., 2017] which reshapes the standard cross entropy to improve the weights of the loss distribution.

To verify the effectiveness of our approach, we conduct experiments on two representative sequence generation tasks.

(1) *Neural Machine Translation*. We transfer the knowledge from the machine comprehension model to the attention-based Neural Machine Translation (NMT) model. Experimental results show that our method significantly improves the performance on several large-scale MT datasets.

(2) *Abstractive Text Summarization*. We modify the Pointer-Generator Networks recently proposed by See et al. [2017]. We evaluate this model on the *CNN/Daily Mail* [Hermann et al., 2015] and *Gigaword* [Rush et al., 2015] datasets. Our model obtains 37.97 ROUGE-1, 18.16 ROUGE-2 and 34.93 ROUGE-L scores on the English *Gigaword* dataset, which is an improvement over previous state-of-the-art results in the literature.

2 Related Work

2.1 Machine Comprehension

Teaching machines to read, process and comprehend text and then answer questions, which is called machine comprehension, is one of the key problems in artificial intelligence. Recently, Rajpurkar et al. [2016] released the Stanford Question Answering Dataset (SQuAD), which is a high-quality and large-scale benchmark, thus inspired many significant works [Xiong et al., 2017; Pan et al., 2017; Cui et al., 2017; Seo et al., 2017; Wang et al., 2016; Xiong et al., 2018; Shen et al., 2017; Wang et al., 2017]. Most of the state-of-the-art works are attention-based neural network models. Seo et al. [2017] propose a bi-directional attention flow to achieve a query-aware context representation. Wang et al. [2017] employ gated self-matching attention to obtain the relation between the question and

passage, and their model is the first one to surpass the human performance on the SQuAD. In this paper, we show that the pre-trained MC architecture can be transferred well to other NLP tasks.

2.2 Sequence-to-sequence Model

Existing sequence-to-sequence models with attention have focused on generating the target sequence by aligning each generated output token to another token in the input sequence. This approach has proven successful in many NLP tasks, such as neural machine translation [Bahdanau et al., 2015], text summarization [Rush et al., 2015] and dialogue systems [Williams et al., 2017], and has also been adapted to other applications, including speech recognition [Chan et al., 2016] and image caption generation [Xu et al., 2015]. In general, these models encode the input sequence as a set of vector representations using a recurrent neural network (RNN). A second RNN then decodes the output sequence step-by-step, conditioned on the encodings. In this work, we augment the natural language understanding of this encoder-decoder framework via transferring knowledge from another supervised task.

2.3 Transfer Learning in NLP

Transfer learning, which aims to build learning machines that generalize across different domains following different probability distributions, has been widely applied in natural language processing tasks [Collobert et al., 2011; Glorot et al., 2011; Min, Seo, and Hajishirzi, 2017; McCann et al., 2017; Pan et al., 2018]. Collobert et al. [2011] propose a unified neural network architecture and learned from unsupervised learning that can be applied to various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and semantic role labelling. Glorot et al. [2011] propose a deep learning approach which learns to extract a meaningful representation for each review in an unsupervised fashion. McCann et al. [2017] propose to transfer the pre-trained encoder from the neural machine translation (NMT) to the text classification and question answering tasks. Pan et al. [2018] propose to transfer the encoder of a pre-trained discourse marker prediction model to the natural language inference model. Unlike previous works that only focus on the encoding part or unsupervised knowledge source, we extract multiple layers of the neural networks from the machine comprehension model and insert them into the sequence-to-sequence model. Our approach not only makes the transfer more directly compatible with subsequent RNNs, but also augments the text understanding of the attention mechanism.

3 Machine Comprehension Model

3.1 Task Description

In the machine comprehension task, we are given a question $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_m\}$ and a passage $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$, where m and n are the length of the question and the passage. The goal is to predict the correct answer $\mathbf{a}^c$ which is a subspan of $\mathbf{P}$.

3.2 Framework

The state-of-the-art MC models are various in structures, but many popular works are essentially the combination of the encoding layer, the attention mechanism with and an RNN-based modeling layer and the output layer[Wang et al., 2016; Seo et al., 2017; Pan et al., 2017; Xiong et al., 2018]. Now we describe our MC model as follows.

Encoding Layer We use pre-trained word vectors *GloVe* [Pennington et al., 2014] and character-level embeddings to transfer the words into vectors, where the latter one applies CNN over the characters of each word and is proved to be helpful in handling out-of-vocab words [Kim, 2014]. We then use a bi-directional LSTM on top of the concatenation of them to model the temporal interactions between words:

$$\begin{aligned} \mathbf{u}_i &= G_{enc}(f_{rep}(\mathbf{q}_i), \mathbf{u}_{i-1}), i = 1, ..., m \\ \mathbf{h}_j &= G_{enc}(f_{rep}(\mathbf{p}_j), \mathbf{h}_{j-1}), j = 1, ..., n \end{aligned} \tag{1}$$

where G_{enc} is the bi-directional LSTM, $f_{rep}(\mathbf{x}) = [\mathrm{Glove}(\mathbf{x}); \mathrm{Char}(\mathbf{x})]$ is the concatenation of the word and character embedding vectors of the word $\mathbf{x}$; $\{\mathbf{u}_i\}_{i=1}^m$ and $\{\mathbf{h}_j\}_{j=1}^n$ are the contextual

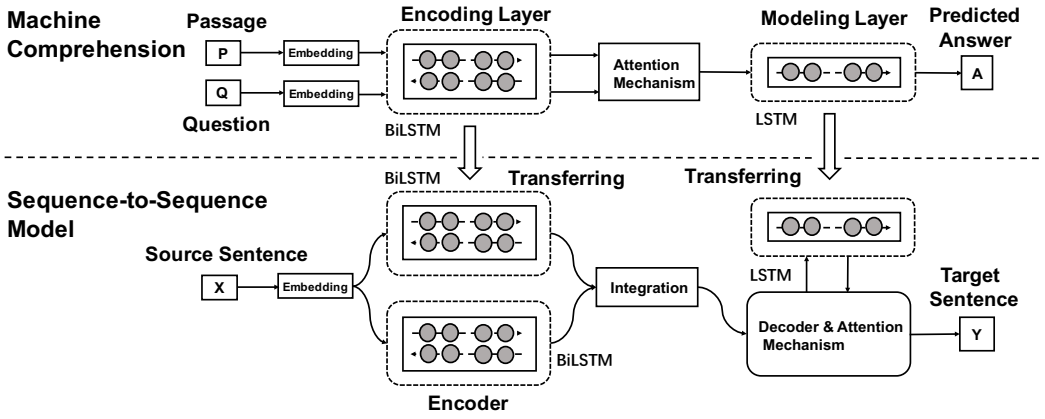

Figure 1: Overview of our MacNet framework, comprising the part of Machine Comprehension (upper) for pre-training and Sequence-to-Sequence model (bottom) to which the learned knowledge will be transferred.

representations of the question $\mathbf{Q}$ and the passage $\mathbf{P}$.

Attention Layer Attention mechanisms are commonly used in machine comprehenion to model the document so that its representation can emphasize the key information and capture long-distance dependencies:

$$\mathbf{G} = f_{att}(\{\mathbf{u}_i\}_{i=1}^{m}, \{\mathbf{h}_j\}_{j=1}^{n}) \tag{2}$$

Here, the attention function f_{att} represents a series of normalized linear and logical operations. We follow [Seo et al., 2017] to use a bi-directional attention flow (BiDAF), where the passage and the question are interacted each other with an alignment matrix, $\mathbf{G}$ is the query-aware context representation.

Modeling Layer In this step, we use the stacking LSTM on $\mathbf{G}$ to further capture the interaction among the passage words conditioned on the question:

$$\mathbf{m}_j = G_{model}(\mathbf{G}_j, \mathbf{m}_{j-1}), j = 1, ..., n \tag{3}$$

where G_{model} is two layers of uni-directional LSTM, each $\mathbf{m}_j$ is expected to represent the contexual information of the j-th word in the passage to the whole question.

We use a simple MLP classifier on the combination of $\{\mathbf{m}_j\}_{j=1}^{n}$ and $\mathbf{G}$ to locate the start and end positions of the answer. For training, we define the training loss as the sum of the negative log probability of the true positions by the predicted distributions.

4 MacNet Architecture

In this section, as shown in the Figure 1, we introduce how our MacNet transfers the knowledge from the MC model to the seq2seq model. The sequence-to-sequence models are typically implemented with a Recurrent Neural Network (RNN)-based encoder-decoder framework. Such a framework directly models the probability $P(y|x)$ of a target sentence $y = \{y_1, y_2, ..., y_{T_y}\}$ conditioned on the source sentence $x = \{x_1, x_2, ..., x_{T_x}\}$, where T_x and T_y are the length of the sentence x and y.

4.1 Encoder

For the seq2seq model, the encoder reads the source sentence x word by word and generates a hidden representation of each word x_s:

$$\tilde{\mathbf{h}}_s = F_{enc}(\mathrm{Emb}(x_s), \tilde{\mathbf{h}}_{s-1}) \tag{4}$$

where F_{enc} is the recurrent unit such as Long Short-Term Memory (LSTM) [Sutskever et al., 2014] unit or Gated Recurrent Unit (GRU) [Cho et al., 2014], $\mathrm{Emb}(x_s)$ is the embedding vector of x_s, $\tilde{\mathbf{h}}_s$ is the hidden state. In this paper, we use the bi-directional LSTM as the recurrent unit to be consistent with the encoding layer of the MC model described in Section 3.2.

To augment the performance of the encoding part, we use a simple method to exploit the word representations that learned from the MC task. For the source sentence x, we use the bi-directional LSTM of the equation (1) as another encoder and obtain:

$$\tilde{\mathbf{e}}_s = G_{enc}(\mathrm{Emb}(x_s), \tilde{\mathbf{e}}_{s-1}) \tag{5}$$

where $\tilde{\mathbf{e}}_s$ is the hidden state, which represents the word x_s from the perspective of the MC model. Instead of the conventional seq2seq models that directly send the results of the equation (4) to the decoder , we concatenate $\tilde{\mathbf{e}}_s$ and $\tilde{\mathbf{h}}_s$ and feed them into an integration layer:

$$\bar{\mathbf{h}}_s = F_{int}([\tilde{\mathbf{h}}_s; \tilde{\mathbf{e}}_s], \bar{\mathbf{h}}_{s-1}) \tag{6}$$

where F_{int} is a uni-directional LSTM, $[;]$ means concatenation. $\{\bar{\mathbf{h}}_s\}_{s=1}^{T_x}$ are the contextual representations of the sentence x which contain the information of the machine comprehension knowledge as well.

4.2 Decoder & Attention Mechanism

Initialized by the representations obtained from the encoder, the decoder with an attention mechanism receives the word embedding of the previous word (while training, it is the previous word of the reference sentence; while testing, it is the previous generated word) at each step and generates next word. The decoder states are computed via:

$$\bar{\mathbf{h}}_t = F_{dec}(\mathrm{Emb}(y_{t-1}), \bar{\mathbf{h}}_{t-1}) \tag{7}$$

where F_{dec} is a unidirectional LSTM, y_t is the t-th generated word, $\bar{\mathbf{h}}_s$ is the hidden state. For most seq2seq attentional models, the attention steps can be summarized by the equations below:

$$\alpha_{ts} = \frac{\exp(\mathrm{score}(\bar{\mathbf{h}}_s, \bar{\mathbf{h}}_t))}{\sum_{s'=1}^{T_x} \exp(\mathrm{score}(\bar{\mathbf{h}}_{s'}, \bar{\mathbf{h}}_t))} \tag{8}$$

$$\mathbf{c}_t = \sum_s \alpha_{ts} \bar{\mathbf{h}}_s \tag{9}$$

$$\mathbf{a}_t = g_a(\mathbf{c}_t, \bar{\mathbf{h}}_t) = \tanh(\mathbf{W}_a[\mathbf{c}_t; \bar{\mathbf{h}}_t] + \mathbf{b}_a) \tag{10}$$

Here, $\mathbf{c}_t$ is the source-side context vector, the attention vector $\mathbf{a}_t$ is used to derive the softmax logit and loss, $\mathbf{W}_a$ and $\mathbf{b}_a$ are trainable parameters, the function g_a can also take other forms. score is referred as a *content-based* function, usually implemented as a feed forward network with one hidden layer.

For the common seq2seq models, the attention vector $\mathbf{a}_t$ is then fed through the softmax layer to produce the predictive distribution formulated as:

$$P(y_t|y_{<t}, x) \propto \mathrm{softmax}(\mathbf{W}_p \mathbf{a}_t + \mathbf{b}_p) \tag{11}$$

In our MacNet, however, we additionally send the attention vector $\mathbf{a}_t$ into the modeling layer of the pre-trained MC model in the equation (3) to deeply capture the interaction of the source and the target states:

$$\mathbf{r}_t = G_{model}(\mathbf{a}_t, \mathbf{r}_{t-1}) \tag{12}$$

where $\mathbf{r}_t$ is another attention state with the augmentation of machine comprehension knowledge. We combine the results of the two attention vectors and the equation (11) becomes:

$$P(y_t|y_{<t}, x) \propto \mathrm{softmax}(\mathbf{W}_p \mathbf{a}_t + \mathbf{W}_q \mathbf{r}_t + \mathbf{b}_p) \tag{13}$$

where $\mathbf{W}_p$, $\mathbf{W}_q$ and $\mathbf{b}_p$ are all trainable parameters. The modeling layer helps deeply understand the interaction of the contextual information of the output sequence, which is different from the encoding layer whose inputs are independent source sentences.

4.3 Training

Denote Θ as all the parameters to be learned in the framework, D as the training dataset that contains source-target sequence pairs. The training process aims at seeking the optimal paramaters Θ^* that encodes the source sequence and provides an output sentence as close as the target sentence. For the formula form, the most popular objective is the maximum log likelihood estimation [Bahdanau et al., 2015; Xia et al., 2017]:

$$\begin{aligned}\Theta^* &= \arg\max_{\Theta} \sum_{(x,y)\in D} P(y|x;\Theta) \\ &= \arg\max_{\Theta} \sum_{(x,y)\in D} \sum_{t=1}^{T_y} \log P(y_t|y_{<t}, x;\Theta)\end{aligned} \tag{14}$$

However, this results in the high frequency of some commonly used expressions such as "I don't know" in the output sentences because of the nature of the class imbalance in the corpus. Inspired by the *focal loss* [Lin et al., 2017], which is recently proposed to solve the foreground-background class imbalance in the task of object detection, we add a modulating factor to the above cross entropy loss. Simplifying $P(y_t|y_{<t}, x;\Theta)$ as p_t, we modify the equation (14) as:

$$\Theta^* = \arg\max_{\Theta} \sum_{(x,y)\in D} \sum_{t=1}^{T_y} (1-p_t)^{\gamma} \log(p_t) \tag{15}$$

where γ is a tunable focusing parameter. In this case, the focusing parameter smoothly adjusts the rate at which high-frequency phrases are down-weighted.

5 Experiments

5.1 Machine Comprehension

We use the Stanford Question Answering Dataset (SQuAD)[Rajpurkar et al., 2016] as our training set[2], which has 100,000+ questions posed by crowd workers on 536 Wikipedia articles. The hidden state size of the LSTM is set as 100, and we select the 300d Glove as the word embeddings. We use 100 one dimensional filters for CNN in the character level embedding, with width of 5 for each one. The dropout ratio is 0.2. We use the AdaDelta [Zeiler, 2012] optimizer with an initial learning rate as 0.001. Our MC model achieves 67.08 of Exact Match (EM) and 76.79 of F1 score on the SQuAD development dataset.

5.2 Application to Neural Machine Translation

We first evaluate our method on the neural machine translation (NMT) task, which requires to encode a source language sentence and predict a target language sentence. We use the architecture from [Luong et al., 2015] as our baseline framework with the GNMT [Wu et al., 2016] attention to parallelize the decoder's computation. The datasets for our evaluation are the WMT translation tasks between English and German in both directions. Translation performances are reported in case-sensitive BLEU [Papineni et al., 2002] on newstest2014[3] and newstest2015[4].

Implementation details: When training our NMT systems, we split the data into subword units using BPE [Sennrich et al., 2016]. We train 4-layer LSTMs of 1024 units with bidirectional encoder, embedding dimension is 1024. We use a fully connected layer to transform the input vector size for the transferred neural networks. The model is trained with stochastic gradient descent with a learning rate that began at 1. We train for 340K steps; after 170K steps, we start halving learning rate every 17K step. Our batch size is set as 128, the dropout rate is 0.2. For the focal loss, the γ is set to be 5.

[2]The SQuAD dataset is referred at: `https://rajpurkar.github.io/SQuAD-explorer/`

[3]`http://www.statmt.org/wmt14/translation-task.html`

[4]`http://www.statmt.org/wmt15/translation-task.html`

NMT Systems	WMT14		WMT15	
	En→De	De→En	En→De	De→En
Baseline	22.1	26.0	24.5	27.5
Baseline + Encoding Layer	23.2	27.0	25.3	28.3
Baseline + Modeling Layer	22.4	26.4	24.8	27.8
Baseline + Encoding Layer + Modeling Layer	23.4	27.3	25.6	28.5
Baseline + Random Initalized Framework	21.6	25.6	24.2	27.0
Baseline + MacNet	**24.2**	**28.1**	**26.3**	**29.4**

Table 2: BLEU scores on official test sets (WMT English-German for `newstest2014` and `newstest2015`). In the top part, we show the performance of our baseline model; In the medium part, we present the ablation experiments; In the bottom part, we show the effectiveness of our MacNet.

Results: As shown in the Table 2, the baseline NMT model on all of the datasets performs much better with the help of our MacNet framework. In the medium part, we conduct an ablation experiment to evaluate the individual contribution of each component of our model. Both of the encoding layer and the modeling layer demonstrates their effectiveness when we ablate other modules. When we add both of them (still without the focal loss), the BLEU scores on all the test sets rise at least 1 point, which shows the significance of the transferred knowledge. Finally, we add the architecture of the encoding layer and the modeling layer to the baseline model but initialize them randomly as its other RNN layers. We observe that the performance drops around 0.5%, which indicates that the machine comprehension knowledge has deep connections with the machine translation tasks. From the ablation experiments we found that the improvement of the modeling layer in our architecture is a bit modest, but we believe transferring high-level networks (*e.g.* the modeling layer) can help a lot with a more suitable structure because those networks contains deeper semantic knowledge and more abstractive information compared with the lower-level layers (*e.g.* encoding layer).

MC Attention	EM	BLEU
Context to Query Attention	63.3	25.1
Query to Context Attention	56.9	25.3
BiDAF	67.1	27.5
BiDAF + Self-Attention	68.2	27.4
BiDAF + Memory Network	68.5	27.6

Table 3: Performance with different pre-trained machine comprehension models for our NMT model on De→En of WMT'14. **EM** means the exact match score, which represents the performance of the MC model on the SQuAD dev set, **BLEU** is the results of our NMT model.

In the Table 3, we explore how different choices of the attention architectures (f_{att} in the equation (2), which is usually the discrimination of different MC models) of the MC models impact the performance of our method. We first follow [Seo et al., 2017] to separate the two directions of the attention in BiDAF and use them to take place of the original attention mechanism respectively. Their performance on the machine comprehension task drops a lot, and it seems to affect the results of the NMT models as well. We then add the self-attention, which is proposed to fuse the context into itself, is widely used by many MC methods [Wang et al., 2017; Weissenborn et al., 2017]. Unfortunately, the result of the NMT model fails to keep pace with the performance of its pre-train MC model. Finally, we apply memory network, which is also very popular among MC models [Pan et al., 2017; Hu et al., 2017], the performance on the SQuAD rises a lot but the NMT result is similar to the original model. This series of experiments denote that the model's performance with our MacNet is not always in positive correlation to the improvement of the

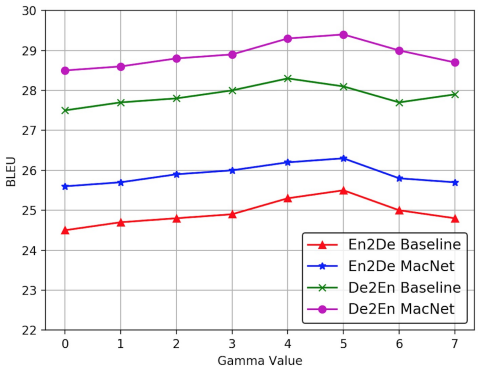

Figure 2: Performance on the WMT'15 with different γ values.

Summarization Models	CNN/Daily Mail			Gigaword		
	RG-1	**RG-2**	**RG-L**	**RG-1**	**RG-2**	**RG-L**
words-lvt5k[Nallapati et al., 2016]	35.46†	13.30†	32.65†	35.30†	16.64†	32.62†
SummaRuNNer[Nallapati et al., 2017]	39.60†	16.20†	35.30†	–	–	–
ConvS2S[Gehring et al., 2017]	–	–	–	35.88†	17.48†	33.29†
SEASS[Zhou et al., 2017]	–	–	–	36.15†	17.54†	33.63†
RL with intra-attn[Paulus et al., 2017]	**41.16**†	15.75†	**39.08**†	–	–	–
Pointer-Generator[See et al., 2017]	39.69	17.26	36.38	36.44	17.26	33.92
Pointer-Generator + Encoding Layer	40.38	17.75	37.24	37.30	17.83	34.41
Pointer-Generator + Modeing Layer	39.92	17.58	36.65	36.85	17.45	34.12
Pointer-Generator + MacNet	40.87	**18.02**	37.54	**37.97**	**18.16**	**34.93**

Table 4: ROUGE F_1 evaluation results on the CNN/Daily Mail test set and the English Gigaword test set. RG in the Table denotes ROUGE. Results with † mark are taken from the corresponding papers. The bottom part of the Table shows the performance of our MacNet and the ablation results.

MC architecture. We conjecture that it might depend on many potential factors such as the complexity of the extracted parts, the heterogeneity of different tasks, *etc*.

In the Figure 2, we present the models on the WMT'15 with different γ to show how the focal loss affects the performance. As we can see, the models increase as the γ enlarges until it arrives 4 or 5. Afterwards, the performance gets worse when we raise the γ, which means the modulating factor is close to zero so that its benefit is limited.

5.3 Application to Text Summarization

We then verify the effectiveness of our MacNet on the abstractive text summarization, which is also a typical application of the sequence-to-sequence model. We use the Pointer-Generator Networks[See et al., 2017] as our baseline model, which applies the encoder-decoder architecture and is one of the state-of-the-art models for the text summarization. The evaluation metric is reported with the F_1 scores for ROUGE-1, ROUGE-2 and ROUGE-L [Lin, 2004]. We evaluate our method on two high-quality datasets, *CNN/Daily Mail* [Hermann et al., 2015] and *Gigaword* [Rush et al., 2015]. For the CNN/Daily Mail dataset, we use scripts[5] supplied by See et al. [2017] to pre-process the data, which contains 287k training pairs, 13k validation pairs and 11k test pairs. For the English Gigaword dataset, we use the script[6] released by Rush et al. [2015] to pre-process and obtain 3.8M training pairs, 189k development set for testing.

Implementation details: Our training hyperparameters are similar to the Pointer-Generator Networks experiments, while some important details are as follows. The input and output vocabulary size is 50k, the hidden state size is 256. The word embedding size is 128, and we use a fully connected layer to transform the input vector size for the transferred neural networks. We train using Adagrad [Duchi et al., 2011] with learning rate 0.15 and an initial accumulator value of 0.1. The γ is set as 3.

Results: Table 4 shows the performance of our methods and the competing approaches on both datasets. Compared to the original Pointer-Generator model, the results with our MacNet architecture outperform around $0.7\% \sim 1.5\%$ on all kinds of the ROUGE scores. Especially, our approach achieves the state-of-the-art results on all the metrics on Gigaword and the ROUGE-2 on CNN/Daily Mail dataset. Similar to the NMT task, the encoding layer contributes most of the improvement, while the modeling layer also has stable gains in each evaluations.

In the Table 5, we present some summaries produced by our model and the original Pointer-Generator model. In the first example, the summary given by the Pointer-Generator model doesn't make sense from the perspective of logic, while our model accurately summarizes the article and even provides

[5] `https://github.com/abisee/cnn-dailymail`

[6] `https://github.com/facebook/NAMAS`

Article: Israeli warplanes raided Hezbollah targets in south Lebanon after guerrillas killed two militiamen and wounded seven other troops on Wednesday, police said. **Reference:** Israeli warplanes raid south Lebanon. **PG + MacNet:** Israeli warplanes attack Hezbollah targets in south Lebanon. **PG:** Hezbollah targets Hezbollah targets in south Lebanon.
Article: The dollar racked up some clear gains on Wednesday on the London forex market as operators waited for the outcome of talks between the White House and Congress on raising the national debt ceiling and on cutting the American budget deficit. **Reference:** Dollar gains as market eyes US debt and budget talks. **PG + MacNet:** : Dollar racked up some clear gains. **PG**: London forex market racked gains.

Table 5: Examples of summaries on English Gigaword, **PG** denotes the Pointer-Generator model.

with more details. In the second example, although the original PG model produces a logical sentence, the output sentence expresses completely different meanings from the information in the article. Our method, however, correctly comprehends the article and provides with a high-quality summary sentence.

6 Conclusion

In this paper, we propose *MacNet*, which is a supplementary framework for the sequence-to-sequence tasks. We transfer the knowledge from the machine comprehension task to a variety of seq2seq tasks to augment the text understanding of the models. The experimental evaluation shows that our method significantly improves the performance of the baseline models on several benchmark datasets for different NLP tasks. We hope this work can encourage further research into the transfer learning of multi-layer neural networks, and the future works involve the choice of other transfer learning sources and the transfer learning between different domains such as NLP, CV, *etc*.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China (Grant Nos: 61751307, 61602405 and U1611461) and in part by the National Youth Top-notch Talent Support Program. The experiments are supported by Chengwei Yao in the Experiment Center of the College of Computer Science and Technology, Zhejiang University.

References

Bahdanau, D.; Cho, K.; Bengio, Y.; et al. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Chan, W.; Jaitly, N.; Le, Q.; and Vinyals, O. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 4960–4964.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017. Attention-over-attention neural networks for reading comprehension. In *ACL*, volume 1, 593–602.

Duchi, J.; Hazan, E.; Singer, Y.; et al. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Glorot, X.; Bordes, A.; Bengio, Y.; et al. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 513–520.

Guo, H.; Pasunuru, R.; and Bansal, M. 2018. Soft layer-specific multi-task summarization with entailment and question generation. *arXiv preprint arXiv:1805.11004*.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *NIPS*, 1693–1701.

Hu, M.; Peng, Y.; Qiu, X.; et al. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, 1746–1751.

Li, J.; Xiong, D.; Tu, Z.; Zhu, M.; Zhang, M.; and Zhou, G. 2017. Modeling source syntax for neural machine translation. In *ACL*, volume 1, 688–697.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Luong, T.; Pham, H.; Manning, C. D.; et al. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, 1412–1421.

McCann, B.; Bradbury, J.; Xiong, C.; and Socher, R. 2017. Learned in translation: Contextualized word vectors. In *NIPS*, 6297–6308.

Min, S.; Seo, M.; and Hajishirzi, H. 2017. Question answering through transfer learning from large fine-grained supervision data. In *ACL*.

Nallapati, R.; Zhou, B.; dos Santos, C.; Gulcehre, C.; and Xiang, B. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 280–290.

Nallapati, R.; Zhai, F.; Zhou, B.; et al. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, 3075–3081.

Pan, B.; Li, H.; Zhao, Z.; Cao, B.; Cai, D.; and He, X. 2017. Memen: Multi-layer embedding with memory networks for machine comprehension. *arXiv preprint arXiv:1707.09098*.

Pan, B.; Yang, Y.; Zhao, Z.; Zhuang, Y.; Cai, D.; and He, X. 2018. Discourse marker augmented network with reinforcement learning for natural language inference. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 989–999.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318. Association for Computational Linguistics.

Paulus, R.; , C.; Socher, R.; et al. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Pennington, J.; Socher, R.; Manning, C. D.; et al. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2383–2392.

Rush, A. M.; Chopra, S.; Weston, J.; et al. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, 379–389.

See, A.; Liu, P. J.; Manning, C. D.; et al. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, volume 1, 1073–1083.

Sennrich, R.; Haddow, B.; Birch, A.; et al. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Shen, Y.; Huang, P.-S.; Gao, J.; and Chen, W. 2017. Reasonet: Learning to stop reading in machine comprehension. In *KDD*, 1047–1055. ACM.

Shi, X.; Padhi, I.; Knight, K.; et al. 2016. Does string-based neural mt learn source syntax? In *ACL*, 1526–1534.

Sutskever, I.; Vinyals, O.; Le, Q. V.; et al. 2014. Sequence to sequence learning with neural networks. In *NIPS*, 3104–3112.

Vinyals, O.; Fortunato, M.; Jaitly, N.; et al. 2015. Pointer networks. In *NIPS*, 2692–2700.

Wang, Z.; Mi, H.; Hamza, W.; and Florian, R. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *ACL*.

Weissenborn, D.; Wiese, G.; Seiffe, L.; et al. 2017. Making neural qa as simple as possible but not simpler. In *CoNLL*, 271–280.

Williams, J. D.; Asadi, K.; Zweig, G.; et al. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*, volume 1, 665–677.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Łukasz Kaiser; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; and Dean, J. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144.

Xia, Y.; Tian, F.; Wu, L.; Lin, J.; Qin, T.; Yu, N.; and Liu, T.-Y. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *NIPS*, 1782–1792.

Xiong, C.; Zhong, V.; Socher, R.; et al. 2017. Dynamic coattention networks for question answering. *ICLR 2017*.

Xiong, C.; Zhong, V.; Socher, R.; et al. 2018. DCN+: Mixed objective and deep residual coattention for question answering. In *ICLR*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2048–2057.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhou, Q.; Yang, N.; Wei, F.; and Zhou, M. 2017. Selective encoding for abstractive sentence summarization. In *ACL*, 1095–1104.