Transfer Learning for Context-Aware Question Matching in Information-seeking Conversations in E-commerce

Minghui Qiu[1], Liu Yang[2], Feng Ji[1], Weipeng Zhao[1], Wei Zhou[1]
Jun Huang[1], Haiqing Chen[1], W. Bruce Croft[2], Wei Lin[1]
[1]Alibaba Group, Hangzhou, China
[2]Center for Intelligent Information Retrieval, University of Massachusetts Amherst
{minghui.qmh,zhongxiu.jf}@alibaba-inc.com
{lyang,croft}@cs.umass.edu

Abstract

Building multi-turn information-seeking conversation systems is an important and challenging research topic. Although several advanced neural text matching models have been proposed for this task, they are generally not efficient for industrial applications. Furthermore, they rely on a large amount of labeled data, which may not be available in real-world applications. To alleviate these problems, we study transfer learning for multi-turn information seeking conversations in this paper. We first propose an efficient and effective multi-turn conversation model based on convolutional neural networks. After that, we extend our model to adapt the knowledge learned from a resource-rich domain to enhance the performance. Finally, we deployed our model in an industrial chatbot called AliMe Assist [1] and observed a significant improvement over the existing online model.

1 Introduction

With the popularity of online shopping, there is an increasing number of customers seeking information regarding their concerned items. To efficiently handle customer questions, a common approach is to build a conversational customer service system (Li et al., 2017; Yang et al., 2018). In the E-commerce environment, the information-seeking conversation system can serve millions of customer questions per day. According to the statistics from a real e-commerce website (Qiu et al., 2017), the majority of customer questions (nearly 90%) are business-related or seeking information about logistics, coupons etc. Among these conversation sessions, 75% of them are more than one turn[2]. Hence it is important to handle multi-turn conversations or context information in these conversation systems.

Recent researches in this area have focused on deep learning and reinforcement learning (Shang et al., 2015; Yan et al., 2016; Li et al., 2016a,b; Sordoni et al., 2015; Wu et al., 2017). One of these methods is Sequential Matching Network(Wu et al., 2017), which matches a response with each utterance in the context at multiple levels of granularity and leads to state-of-the-art performance on two multi-turn conversation corpora. However, such methods suffer from at least two problems: they may not be efficient enough for industrial applications, and they rely on a large amount of labeled data which may not be available in reality.

To address the problem of efficiency, we made three major modifications to SMN to boost the efficiency of the model while preserving its effectiveness. First, we remove the RNN layers of inputs from the model; Second, SMN uses a Sentence Interaction based (SI-based) Pyramid model (Pang et al., 2016) to model each utterance and response pair. In practice, a Sentence Encoding based (SE-based) model like BCNN (Yin and Schütze, 2015) is complementary to the SI-based model. Therefore, we extend the component to incorporate an SE-based BCNN model, resulting in a hybrid CNN (hCNN) (Yu et al., 2017); Third, instead of using a RNN to model the output representations, we consider a CNN model followed by a fully-connected layer to further boost the efficiency of our model. As shown in our experiments, our final model yields comparable results

[1] Interested readers can access AliMe Assist through the Taobao App, or the web version via https://consumerservice.taobao.com/online-help

[2] According to a statistic in AliMe Assist in Alibaba Group

but with higher efficiency than SMN.

To address the second problem of insufficient labeled data, we study transfer learning (TL) (Pan and Yang, 2010) to utilize a source domain with adequate labeling to help the target domain. A typical TL approach is to use a shared NN (Mou et al., 2016; Yang et al., 2017) and domain-specific NNs to derive shared and domain-specific features respectively. Recent studies (Ganin et al., 2016; Taigman et al., 2017; Chen et al., 2017; Liu et al., 2017) consider adversarial networks to learn more robust shared features across domains. Inspired by these studies, we extended our method with a Transfer Learning module to leverage information from a resource-rich domain. Similarly, our TL module consists of a shared NN and two domain-specific NNs for source and target domains. The output of the shared NN is further linked to an adversarial network as used in (Liu et al., 2017) to help learn domain invariant features. Meanwhile, we also use domain discriminators on both source and target features derived by domain-specific NNs to help learn domain-specific features. Experiments show that our TL method can further improve the model performance on a target domain with limited data.

To the best of our knowledge, our work is the first to study transfer learning for context-aware question matching in conversations. Experiments on both benchmark and commercial data sets show that our proposed model outperforms several baselines including the state-of-the-art SMN model. We have also deployed our model in an industrial bot called AliMe Assist [3] and observed a significant improvement over the existing online model.

2 Model

Our model is designed to address the following general problem. Given an input sequence of utterances $\{u_1, u_2, \ldots, u_n\}$ and a candidate question r, our task is to identify the matching degree between the utterances and the question. When the number of utterances is one, our problem is identical to paraphrase identification (PI) (Yin and Schütze, 2015) or natural language inference (NLI) (Bowman et al., 2015). Furthermore, we consider a transfer learning setting to transfer knowledge from a source domain to help a target domain.

2.1 Multi-Turn hCNN (MT-hCNN)

We present an overview of our model in Fig. 1. In a nutshell, our model first obtains a representation for each utterance and candidate question pair using hybrid CNN (hCNN), then concatenates all the representations, and feeds them into a CNN and fully-connected layer to obtain our final output.

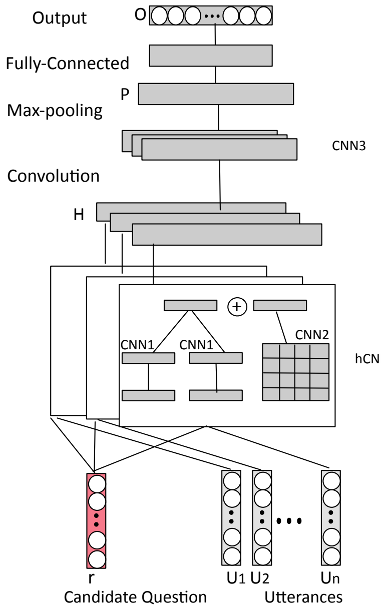

Figure 1: Our proposed multi-turn hybrid CNN.

The hybrid CNN (hCNN) model (Yu et al., 2017) is based on two models: a modified SE-based BCNN model (Yin et al., 2016) and a SI-based Pyramid model (Pang et al., 2016). The former encode the two input sentences separately with a CNN and then combines the resulting sentence embeddings as follows:

$$\begin{aligned} \mathbf{h_1} &= \mathrm{CNN}_1(\mathbf{X}_1); \quad \mathbf{h_2} = \mathrm{CNN}_1(\mathbf{X}_2). \\ \mathbf{H_b} &= \mathbf{h_1} \oplus \mathbf{h_2} \oplus (\mathbf{h_1} - \mathbf{h_2}) \oplus (\mathbf{h_1} \cdot \mathbf{h_2}). \end{aligned}$$

where '$-$' and '$\cdot$' refer to element-wise subtraction and multiplication, and '$\oplus$' refers to concatenation.

Furthermore, we add a SI-base Pyramid component to the model, we first produce an interaction matrix $\mathbf{M} \in \mathbb{R}^{m \times m}$, where $\mathbf{M_{i,j}}$ denotes the dot-product score between the i^{th} word in $\mathbf{X}_1$ and the j^{th} word in $\mathbf{X}_2$. Next, we stack two 2-D convolutional layers and two 2-D max-pooling layers on it to obtain the hidden representation $\mathbf{H_p}$. Finally, we concatenate the hidden representations as output for each input sentence pair: $\mathbf{Z_{X_1,X_2}} = \mathrm{hCNN}(X_1, X_2) = \mathbf{H_b} \oplus \mathbf{H_p}$.

[3] https://consumerservice.taobao.com/online-help

We now extend hCNN to handle multi-turn conversations, resulting MT-hCNN model. Let $\{u_1, u_2, u_3, \ldots, u_n\}$ be the utterances, r is the candidate question.

$$
\begin{aligned}
\mathbf{h_{u_i,r}} &= \text{hCNN}(\mathbf{u_i}, r). \quad for \quad i \in [1, n] \\
H &= [h_{u_1,r}; h_{u_2,r}; \cdots; h_{u_n,r}]. \\
P &= \text{CNN}_3(H). \\
O &= \text{Fully-Connected}(P)
\end{aligned}
$$

Note that H is obtained by stacking all the $\mathbf{h}$, CNN$_3$ is another CNN with a 2-D convolutional layer and a 2-D max-pooling layer, the output of CNN$_3$ is feed into a fully-connected layer to obtain the final representation O.

2.2 Transfer with Domain Discriminators

We further study transfer learning (TL) to learn knowledge from a source-rich domain to help our target domain, in order to reduce the dependency on a large scale labeled training data. As similar to (Liu et al., 2017), we use a shared MT-hCNN and two domain-specific MT-hCNNs to derive shared features $\mathbf{O^c}$ and domain-specific features $\mathbf{O^s}$ and $\mathbf{O^t}$. The domain specific output layers are:

$$
\hat{y}^k = \begin{cases} \sigma(\mathbf{W^{sc}O^c} + \mathbf{W^sO^s} + \mathbf{b^s}), \text{ if } k = s \\ \sigma(\mathbf{W^{tc}O^c} + \mathbf{W^tO^t} + \mathbf{b^t}), \text{ if } k = t \end{cases} \tag{1}
$$

where $\mathbf{W^{sc}}$, $\mathbf{W^{tc}}$, $\mathbf{W^s}$, and $\mathbf{W^t}$ are the weights for shared-source, shared-target, source, and target domains respectively, while $\mathbf{b^s}$ and $\mathbf{b^t}$ are the biases for source and target domains respectively.

Following (Liu et al., 2017), we use an adversarial loss L_a to encourage the shared features learned to be indiscriminate across two domains:

$$
L_a = \frac{1}{n}\sum_{i=1}^{n}\sum_{d \in s,t} p(d_i = d|\mathbf{U}, r)\log p(d_i = d|\mathbf{U}, r).
$$

where d_i is the domain label and $p(d_i|\cdot)$ is the domain probability from a domain discriminator.

Differently, to encourage the specific feature space to be discriminable between different domains, we consider applying domain discrimination losses on the two specific feature spaces. We further add two negative cross-entropy losses: L_s for source and L_t for target domain:

$$
L_s = -\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbb{I}^{d_i = s}\log p(d_i = s|\mathbf{U}^s, r^s).
$$

$$
L_t = -\frac{1}{n_t}\sum_{i=1}^{n_t}\mathbb{I}^{d_i = t}\log p(d_i = t|\mathbf{U}^t, r^t).
$$

where $\mathbb{I}^{d_i=d}$ is an indicator function set to 1 when the statement ($d_i = d$) holds, or 0 otherwise.

Finally, we obtain a combined loss as follows:

$$
\begin{aligned}
\mathcal{L} = \sum_{\mathbf{k}\in \mathbf{s},\mathbf{t}} -\frac{1}{n_\mathbf{k}}\sum_{j=1}^{n_\mathbf{k}}\frac{1}{2}(y_j^k - \hat{y}_j^k)^2 + \frac{\lambda_1}{2}L_a \\
+ \frac{\lambda_2}{2}L_s + \frac{\lambda_3}{2}L_t + \frac{\lambda_4}{2}||\mathbf{\Theta}||_F^2.
\end{aligned}
$$

where $\mathbf{\Theta}$ denotes model parameters.

3 Experiments

We evaluate the efficiency and effectiveness of our base model, the transferability of the model, and the online evaluation in an industrial chatbot.

Datasets: We evaluate our methods on two multi-turn conversation corpus, namely Ubuntu Dialog Corpus (UDC) (Lowe et al., 2015) and AliMe data.

Ubuntu Dialog Corpus: The Ubuntu Dialog Corpus (UDC) (Lowe et al., 2015) contains multi-turn technical support conversation data collected from the chat logs of the Freenode Internet Relay Chat (IRC) network. We used the data copy shared by Xu et al. (Xu et al., 2016), in which numbers, urls and paths are replaced by special placeholders. It is also used in several previous related works (Wu et al., 2017; Yang et al., 2018)[4]. It consists of 1 million context-response pairs for training, 0.5 million pairs for validation and 0.5 million pairs for testing.

AliMe Data: We collect the chat logs between customers and a chatbot called AliMe from "2017-10-01" to "2017-10-20" in Alibaba [5]. The chatbot is built based on a question-to-question matching system (Li et al., 2017), where for each query, it finds the most similar candidate question in a QA database and return its answer as the reply. It indexes all the questions in our QA database using

[4] The data can be downloaded from https://www.dropbox.com/s/2fdn26rj6h9bpvl/ubuntu%20data.zip?dl=0

[5] The textual contents related to user information are filtered.

Table 1: Comparison of base models on Ubuntu Dialog Corpus (UDC) and an E-commerce data (AliMe).

Data	UDC					AliMeData				
Methods	MAP	R@5	R@2	R@1	Time	MAP	R@5	R@2	R@1	Time
ARC-I	0.2810	0.4887	0.1840	0.0873	16	0.7314	0.6383	0.3733	0.2171	23
ARC-II	0.5451	0.8197	0.5349	0.3498	17	0.7306	0.6595	0.3671	0.2236	24
Pyramid	0.6418	0.8324	0.6298	0.4986	17	0.8389	0.7604	0.4778	0.3114	27
Duet	0.5692	0.8272	0.5592	0.4756	20	0.7651	0.6870	0.4088	0.2433	30
MV-LSTM	0.6918	0.8982	0.7005	0.5457	1632	0.7734	0.7017	0.4105	0.2480	2495
SMN	**0.7327**	**0.9273**	0.7523	0.5948	64	0.8145	0.7271	0.4680	0.2881	91
MT-hCNN-d	0.7027	0.8992	0.7512	0.5838	20	0.8401	0.7712	0.4788	0.3238	31
MT-hCNN	0.7323	0.9172	**0.7525**	**0.5978**	24	**0.8418**	**0.7810**	**0.4796**	**0.3241**	36

Lucene[6]. For each given query, it uses TF-IDF ranking algorithm to call back candidates. To form our data set, we concatenated utterances within three turns [7] to form a query, and used the chatbot system to call back top 15 most similar candidate questions as candidate "responses". [8] We then asked a business analyst to annotate the candidate responses, where a "response" is labeled as positive if it matches the query, otherwise negative. In all, we have annotated 63,000 context-response pairs. This dataset is used as our *Target* data.

Furthermore, we build our *Source* data as follows. In the AliMe chatbot, if the confidence score of answering a given user query is low, i.e. the matching score is below a given threshold[9], we prompt top three related questions for users to choose. We collected the user click logs as our source data, where we treat the clicked question as positive and the others as negative. We collected 510,000 query-question pairs from the click logs in total as the source. For the source and target datasets, we use 80% for training, 10% for validation, and 10% for testing.

Compared Methods: We compared our multi-turn model (MT-hCNN) with two CNN based models ARC-I and ARC-II (Hu et al., 2014), and several advanced neural matching models: MV-LSTM (Wan et al., 2016), Pyramid (Pang et al., 2016) Duet (Mitra et al., 2017), SMN (Wu et al., 2017)[10], and a degenerated version of our model that removes CNN_3 from our MT-hCNN model (MT-hCNN-d). All the methods in this paper are implemented with TensorFlow and are trained with NVIDIA Tesla K40M GPUs.

Settings: We use the same parameter settings of hCNN in (Yu et al., 2017). For the CNN_3 in our model, we set window size of convolution layer as 2, ReLU as the activation function, and the stride of max-pooling layer as 2. The hidden node size of the Fully-Connected layer is set as 128. AdaDelta is used to train our model with an initial learning rate of 0.08. We use MAP, Recall@5, Recall@2, and Recall@1 as evaluation metrics. We set $\lambda_1 = \lambda_2 = \lambda_3 = 0.05$, and $\lambda_4 = 0.005$.

3.1 Comparison on Base Models

The comparisons on base models are shown in Table 1. First, the RNN based methods like MV-LSTM and SMN have clear advantages over the two CNN-based approaches like ARC-I and ARC-II, and are better or comparable with the state-of-the-art CNN-based models like Pyramid and Duet; Second, our MT-hCNN outperforms MT-hCNN-d, which shows the benefits of adding a convolutional layer to the output representations of all the utterances; Third, we find SMN does not perform well in AliMeData compared to UDC. One potential reason is that UDC has significantly larger data size than AliMeData (1000k vs. 51k), which can help to train a complex model like SMN; Last but not least, our proposed MT-hCNN shows the best results in terms of all the metrics in AliMeData, and the best results in terms of R@2 and R@1 in UDC, which shows the effectiveness of MT-hCNN.

We further evaluate the inference time [11] of these models. As shown in Table 1, MT-hCNN has comparable or better results when compared with SMN (the state-of-the-art multi-turn conversation model), but is much more efficient than SMN (∼60% time reduction). MT-hCNN also has similar efficiency with CNN-based methods but with better performance. As a result, our MT-

[6] https://lucene.apache.org/core/

[7] Around 85% of conversations are within 3 turns.

[8] A "response" here is a question in our system.

[9] The threshold is determined by a business analyst

[10] The results are based on the TensorFlow code from authors, and with no over sampling of negative training data.

[11] The time of scoring a query and N candidate questions, where N is 10 in UDC, and 15 in AliMeData.

hCNN module is able to support a peak QPS [12] of 40 on a cluster of 2 service instances, where each instance reserves 2 cores and 4G memory on an Intel Xeon E5-2430 machine. This shows the model is applicable to industrial bots. In all, our proposed MT-hCNN is shown to be both efficient and effective for question matching in multi-turn conversations.

3.2 Transferablity of our model

To evaluate the effectiveness of our transfer learning setting, we compare our full model with three baselines: Src-only that uses only source data, Tgt-only that uses only target data, and TL-S that uses both source and target data with the adversarial training as in (Liu et al., 2017).

As in Table 2, Src-only performs worse than Tgt-only. This shows the source and target domains are related but different. Despite the domain shift, TL-S is able to leverage knowledge from the source domain and boost performance; Last, our model shows better performance than TL-S, this shows the helpfulness of adding domain discriminators on both source and target domains.

Table 2: Transferablity of our model.

Data	E-commerce data (AliMeData)			
Methods	MAP	R@5	R@2	R@1
Src-only	0.7012	0.7123	0.4343	0.2846
Tgt-only	0.8418	0.7810	0.4796	0.3241
TL-S	0.8521	0.8022	0.4812	0.3255
Ours	**0.8523**	**0.8125**	**0.4881**	**0.3291**

3.3 Online Evaluations

We deployed our model online in AliMe Assist Bot. For each query, the bot uses the TF-IDF model in Lucene to return a set of candidates, then uses our model to rerank all the candidates and returns the top. We set the candidate size as 15 and context length as 3. To accelerate the computation, we bundle the 15 candidates into a mini-batch to feed into our model. We compare our method with the online model - a degenerated version of our model that only uses the current query to retrieve candidate, i.e. context length is 1. We have run 3-day A/B testing on the Click-Through-Rate (CTR) of the models. As shown in Table 3, our method consistently outperforms the online model, yielding 5% ∼ 10% improvement.

Table 3: Comparison with the online model.

CTR	Day1	Day2	Day3
Online Model	0.214	0.194	0.221
Our Model	**0.266**	**0.291**	**0.288**

4 Related Work

Recent research in multi-turn conversations has focused on deep learning and reinforcement learning (Shang et al., 2015; Yan et al., 2016; Li et al., 2016a,b; Sordoni et al., 2015; Wu et al., 2017; Yang et al., 2018). The recent proposed Sequential Matching Network (SMN) (Wu et al., 2017) matches a response with each utterance in the context at multiple levels of granularity, leading to state-of-the-art performance on two multi-turn conversation corpora. Different from SMN, our model is built on CNN based modules, which has comparable results but with better efficiency.

We study transfer learning (TL) (Pan and Yang, 2010) to help domains with limited data. TL has been extensively studied in the last decade. With the popularity of deep learning, many Neural Network (NN) based methods are proposed (Yosinski et al., 2014). A typical framework uses a shared NN to learn shared features for both source and target domains (Mou et al., 2016; Yang et al., 2017). Another approach is to use both a shared NN and domain-specific NNs to derive shared and domain-specific features (Liu et al., 2017). This is improved by some studies (Ganin et al., 2016; Taigman et al., 2017; Chen et al., 2017; Liu et al., 2017) that consider adversarial networks to learn more robust shared features across domains. Our TL model is based on (Liu et al., 2017), with enhanced source and target specific domain discrimination losses.

5 Conclusion

In this paper, we proposed a conversation model based on Multi-Turn hybrid CNN (MT-hCNN). We extended our model to adapt knowledge learned from a resource-rich domain. Extensive experiments and an online deployment in AliMe E-commerce chatbot showed the efficiency, effectiveness, and transferablity of our proposed model.

Acknowledgments

The authors would like to thank Juwei Ren, Zhiyu Min and other members of AliMe team for their

[12] Queries Per Second

help in deploying our model online. This work was supported in part by NSF grant IIS-1419693. We would also like to thank reviewers for their valuable comments.

References

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

X. Chen, Z. Shi, X. Qiu, and X. Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. *CoRR* abs/1704.07556.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59):1–35.

B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS '14*.

F. Li, M. Qiu, H. Chen, X. Wang, X. Gao, J. Huang, J. Ren, Z. Zhao, W. Zhao, L. Wang, and G. Jin. 2017. Alime assist: An intelligent assistant for creating an innovative e-commerce experience. In *CIKM 2017. Demo*.

J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and W. B. Dolan. 2016a. A persona-based neural conversation model. In *ACL'16*.

J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. 2016b. Deep reinforcement learning for dialogue generation. In *EMNLP'16*.

P. Liu, X. Qiu, and X. Huang. 2017. Adversarial multi-task learning for text classification. In *ACL*.

R. Lowe, N. Pow, I. Serban, and J. Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR* abs/1506.08909.

B. Mitra, F. Diaz, and N. Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW '17*.

L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. 2016. How transferable are neural networks in nlp applications? In *EMNLP*.

S. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng. 2016. Text matching as image recognition. In *AAAI*.

M. Qiu, F. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu. 2017. Alime chat: A sequence to sequence and rerank based chatbot engine. In *ACL*.

L. Shang, Z. Lu, and H. Li. 2015. Neural responding machine for short-text conversation. In *ACL '15*.

A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *NAACL '15*.

Y. Taigman, A. Polyak, and L. Wolf. 2017. Unsupervised cross-domain image generation. *ICLR* .

S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI '16*.

Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *ACL '17*.

Z. Xu, B. Liu, B. Wang, C. Sun, and X. Wang. 2016. Incorporating loose-structured knowledge into LSTM with recall gate for conversation modeling. *CoRR* .

R. Yan, Y. Song, and H. Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR '16*.

L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, and H. Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *SIGIR '18*.

Z. Yang, R. Salakhutdinov, and W. W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *ICLR* .

W. Yin and H. Schütze. 2015. Convolutional neural network for paraphrase identification. In *NAACL-HLT*.

W. Yin, H. Schütze, B. Xiang, and B. Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of ACL* 4:259–272.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. 2014. How transferable are features in deep neural networks? In *NIPS*.

J. Yu, M. Qiu, J. Jiang, J. Huang, S. Song, W. Chu, and H. Chen. 2017. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. *WSDM* .