Question Retrieval with Adversarial Domain Adaptation

Yi-Shiuan Tung
ytung@mit.edu

Yafei Han
yafei@mit.edu

***Abstract*—In this paper, we build neural network models to tackle the question retrieval problem and explore domain adaptation methods to perform transfer learning. To learn question similarity, we use standard bidirectional LSTM and CNN as encoder to map the question title and body to a vector representation. This question representation is then compared to other questions via cosine similarity. Our best model achieves 13% performance improvement over a standard IR baseline. Second, we explore adversarial domain adaptation methods to transfer models to unlabeled data in a different domain. We implement *gradient reversal layer* (GRL) (Ganin and Lempitsky, 2014) and *adversarial discriminative domain adaptation* (ADDA) (Tzeng et al., 2017). Using direct transfer as baseline, GRL achieves 8% performance improvement on the target data.** [1]

***Keywords*—question retrieval; adversarial domain adaptation, gradient reversal, adversarial discriminative domain adaptation**

I. Introduction

Question answering (QA) forums are rapidly growing with many duplicated and related questions. Developing algorithms to learn question similarity enables us to reuse existing answers and avoid repeated work. Question similarity learning is still a challenging task for two main reasons (Lei et al., 2015). First, the body of a question can be long and contain many details. The irrelevant or erroneous information in the question body can confuse word-matching algorithms. While title can summarize the content concisely, it lacks crucial details from the body. As deep neural networks have shown good performance in learning feature representation, we design two deep neural network architecture - LSTM and CNN to learn question similarity.

The second challenge is due to a lack of labelled data for model training. For example, Lei et al. (2015) finds that only 5% of a sample set of questions from AskUbuntu are marked as similar by forum users. Given a large amount of unlabelled questions, we need *domain adaptation* (DA) methods to transfer model trained on a source domain, where labelled data is abundant, to a target domain with none or a few labelled data. We hope to learn features that are both discriminative in prediction task and domain-invariant. To perform transfer learning, we explore two adversarial domain adaptation methods: *gradient reversal* and *adversarial discriminative domain adaptation*, to improve transfer learning performance of neural networks.

[1] Code is available at: https://github.com/DeborahHan/NLP˙Final˙Project

II. Related Work

The major challenge in question retrieval is the lexical gap between the queried question and historical questions (Zhou et al., 2011). Previous approaches include word and phrased-based translation models, topic models, and knowledge-based systems (Zhou et al., 2011; Cao et al., 2010, 2009; Ji et al., 2012; Cai et al., 2011; Zhou et al., 2013). The translation models are analogous to machine translation and learn the probability of translating a word or phrase to another. The topic models map questions to a latent topic and are usually combined with translation models to improve performance (Cao et al., 2010; Ji et al., 2012; Cai et al., 2011). Knowledge-based models use available databases such as Wikipedia to get the semantic relations to measure question similarity.

More recent work focuses on learning representations through neural networks. For example, dos Santos et al. (2015) use a CNN and a bag-of-words representation for comparing questions. Lei et al. (2015) combines CNN and neural gates similar to LSTMs. The neural gates allow for context dependent weights by giving low weights to tokens that provide no relevant information and high weights to strong semantic content words. This paper uses a similar setup to Lei et al. (2015) but evaluates on CNN, LSTM, and bidirectional LSTMs.

Previous work in domain adaptation include Huang et al. (2007), which reweighs samples from the source domain to reduce the distance between source and target samples in a reproducing kernel Hilbert space (RKHS). Several approaches find an explicit feature mapping that maps source to target domains (Sinno-Jialin-Pan, 2011; Gopalan et al., 2011). Sinno-Jialin-Pan (2011) proposes learning a kernel transformation of both source and target domains that minimizes Maximum Mean Discrepancy (MMD) in RKHS. More recent work focuses on transferring representations generated by neural networks on labeled source datasets to a target dataset without labels or with sparse labels. In the unlabeled case, the main approach is to alter the representation learned from source domain such that the difference between source and target feature distributions is minimized (Ganin and Lempitsky, 2014; Tzeng et al., 2017). This paper explores the methods by Ganin and Lempitsky (2014) and Tzeng et al. (2017) in the context of question retrieval.

III. Question Retrieval

We use the setup as described in Lei et al. (2015). Given a query question q that consists of a title and a body, we

retrieve a candidate set of related questions $Q(q)$ using a standard IR engine. The goal is to rank the candidate questions $Q(q)$ such that similar questions are ranked higher than dissimilar ones. The ranking is based on a similarity score $s(q, p; \theta)$. During training, we use a set of annotated data $D = \{(q_i, p_i^+, Q_i^-)\}$, where p_i^+ is a question similar to q_i obtained from user-marked pairs, and Q_i^- is a set of negative samples drawn randomly from the corpus. The likelihood of drawing a similar question is small given the size of the corpus.

To learn the parameters θ for the similarity score, we minimize the max-margin loss $L(\theta)$ defined as

$$\max_{p \in Q(q_i)} \{s(q_i, p; \theta) - s(q_i, p_i^+) + \delta(p, p_i^+)\} \tag{1}$$

where $\delta(.,.)$ is a small constant when $p \neq p_i^+$ and 0 otherwise. Similar to Lei et al. (2015), we use a neural network as an $encoder$ to map each question to a vector, and the score $s(q, p; \theta)$ is the cosine similarity of the vectors for q and p. However, we do not use pre-training to initialize the parameters of the neural network.

In Lei et al. (2015), the $encoder$ is pretrained using an encoder-decoder model that predicts $P(title|context)$, where context can be any of (a) original title, (b) question body, (c) title/body of a similar question. The model is meant to be a denoising auto-encoder since the body generally contains more information but has more noise.

A. Model Description

The two models used for encoding each question to its vector representation are vanilla CNN and LSTM. We also explore bidirectional LSTM. Both models have a single hidden layer and use $tanh$ as the activation function. The hidden layer size and output layer size are kept the same. The input size is the size of the word embedding.

1) CNN: CNN has been shown to perform well in computer vision as well as in NLP tasks. Instead of performing a 2-D convolution over images, a 1-D convolution maps chunks of sentences into feature representations. Let n be the filter width and $W_1, ..., W_n$ be the corresponding filter matrices, the convolution operation is applied to each window of n consecutive words as follows:

$$\begin{aligned} c_t &= W_1 x_{t-n+1} + W_2 x_{t-n+2} + ... + W_n x_t \\ h_t &= tanh(c_t + b) \end{aligned} \tag{2}$$

Each vector h_t is then aggregated by max-pooling or average-pooling to obtain the final vector representation for the whole sentence. To get a single vector for each question, we average the outputs of the title and body from the neural network.

2) LSTM: LSTM captures longer dependencies in sentences and has been used successfully in applications such as machine translation and sentiment analysis (Sutskever et al., 2014; Tang et al., 2015). LSTM uses neural gates that can adaptively store or discard information as each word in the sentence is read successively. For each input word token x_t, internal state c_{t-1}, and visible state h_{t-1}, new states c_t and h_t are generated as follows:

$$\begin{aligned} i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\ f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\ o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o) \\ z_t &= tanh(W^z x_t + U^z h_{t-1} + b^z) \\ c_t &= i_t \odot z_t + f_t \odot c_{t-1} \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{3}$$

where i, f, o, and z are input, forget, output, and update gates. The final vector representation can either be an average of all h_t (mean-pooling) or simply the last h_t. Like CNN, the outputs of title and body are averaged to get a representation for the question.

B. Experimental Setup

1) Dataset and Evaluation: The dataset and evaluation are the same as Lei et al. (2015); a brief summary is provided below. We use the Stack Exchange AskUbuntu dataset (dos Santos et al., 2015), which contains 167,765 unique questions and a set of user-marked similar question pairs. In training, the user-marked similar question pairs are used as positive pairs. For each positive pair, randomly sampled 20 questions from the corpus are used as negative pairs. The fixed word embeddings as well as development and test sets are provided by Lei et al. (2015). The models are evaluated based on the following IR metrics: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Precision at 1 (P@1), and Precision at 5 (P@5).

2) Hyperparameters: For CNN and LSTM, we used Adam as the optimization method. The hyperparameters tried are hidden layer dimensions $\in \{100...700\}$ and different pooling strategies. The best training epoch and model configuration is determined by MRR.

IV. Transfer Learning

A. Domain Adaptation Models

We implement two adversarial domain adaptation techniques.

1) Gradient Reversal Layer (GRL): The network consists of two parts: a feature extractor M and a domain classifier D (Fig. 1, eqn. 4). Feature extractor M maps input x to hidden output h. M is a CNN and LSTM network as implemented in section III. Training input from source domain (x_s) and target domain (x_t) are mapped by M to a hidden layer h using shared weights θ_M.

Hidden outputs generated from source domain and target domain are fed into a domain classifier D, which is a feed-forward network with 1 hidden layer and 2 output units corresponding to a binary domain label d ($d = 1$ for source domain and 0 otherwise).

$$\begin{aligned} h &= M(x, \theta_M) \\ d &= D(h, \theta_D) \end{aligned} \tag{4}$$

The training loss of feature extractor L_Y is the max-margin loss similar to that used in question similarity learning in Part III (eqn. 1). It is computed only for training data from source

domain where similarity tags are available. Loss of domain classifier L_D is a binary cross entropy loss computed on both source and target domain (eqn. 6).

Model training is "adversarial" because M and D optimize competing goals. Feature extractor M minimizes similarity loss L_Y to improve similarity prediction; and maximize domain classification loss L_D to make source and target domain question representation indistinguishable. Domain classifier D aims to minimize the domain classification loss L_D to distinguish source and target domain.

We utilize a single loss function L to encode the losses (eqn. 7). λ is a hyperparameter that trades off these two losses. We use two optimizers to separately update θ_M and θ_D. Learning rate of optimizer for θ_D is set to be negative such that updating θ_D decreases L_D. Alternatively, we can use a gradient reversal layer between hidden layer and domain classifier to achieve the same goal.

$$L_Y(\theta_M) = E_{x_s} MaxMarginLoss(M(x_s)) \tag{5}$$

$$\begin{aligned} L_D(\theta_D, \theta_M) = &- E_{x_s}[\log D(M(x_s))] \\ &- E_{x_t}[\log(1 - D(M(x_t)))] \end{aligned} \tag{6}$$

$$\min_{\theta_M} \max_{\theta_D} L = L_Y(\theta_M) - \lambda L_D(\theta_D, \theta_M) \tag{7}$$

2) Adversarial Discriminative Domain Adaptation (ADDA): Tzeng et al. (2017) propose a generalized architecture for adversarial domain adaptation. Existing adversarial adaptation methods can be viewed as instances of a unified framework with three choices: whether to use a generative or discriminative base model; whether to tie or untie the weights; and which adversarial learning objective to use.

In *GRL*, we use a discriminative model for domain classification. Feature extraction weights are shared between source and target domain, which learns a symmetric mapping of questions to the shared feature space. The adversarial loss of feature extractor is exactly the reverse of the adversarial loss of domain classifier: $L_{advM} = -L_{advD} = -L_D$. The adversarial learning objective takes the minmax form.

In *ADDA*, we adopt a similar setting proposed by Tzeng et al. (2017): a discriminative base model, untied weights and the standard GAN loss. The sequential learning are carried out in two steps depicted in Fig. 2.

First, we pre-train a LSTM/CNN network to obtain feature mapping (M_s) for question similarity task on source domain only. Second, we allow independent source and target mappings by untying the weights, which allows more freedom to learn domain specific feature mapping. We fix the source feature mapping M_s, and learn the target feature mapping M_t so as to match the distribution of source domain question representation. The parameterization of M_t is similar to M_s, which is a LSTM or CNN.

Then we conduct adversarial learning with two losses: one for the discriminator (L_{advD}) and one for the feature

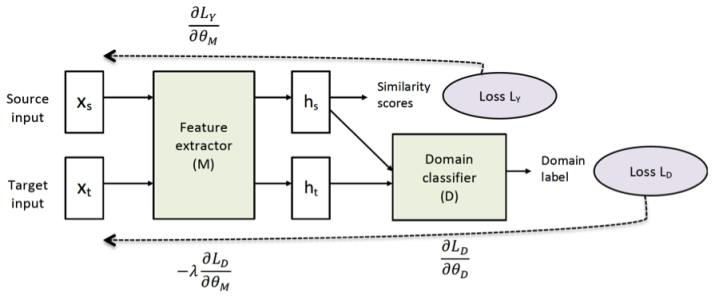

Fig. 1: Gradient Reversal Domain Adaptation Diagram

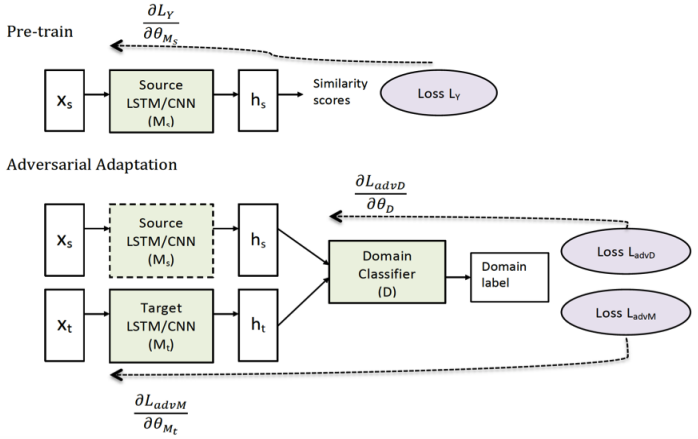

Fig. 2: Adversarial Discriminative Domain Adaptation Diagram

extractor (L_{advM}). L_{advD} has the same binary cross entropy form as L_D before, except that the source and target domain have different feature mapping weights (M_s and M_t). But the adversarial loss for feature extractor L_{advM} is no longer $-L_{advD}$. We use GAN loss function - the expected log-probability of classifying target as source. Minimizing GAN loss is to maximize the log-probability of the discriminator being mistaken, instead of directly minimizing the log-probability of the discriminator being correct. This objective has the same fixed-point properties as the minimax loss but provides stronger gradients to the target mapping(Goodfellow, 2016). The overall training objectives of the two steps are:

$$\min_{\theta_{M_s}} L_Y(\theta_{M_s}) = E_{x_s} MaxMarginLoss(M_s(x_s)) \tag{8}$$

$$\begin{aligned} \min_{\theta_D} L_{advD}(\theta_D, \theta_{M_s}, \theta_{M_t}) = &- E_{x_s}[\log D(M_s(x_s))] \\ &- E_{x_t}[\log(1 - D(M_t(x_t)))] \end{aligned} \tag{9}$$

$$\min_{\theta_{M_t}} L_{advM}(\theta_D, \theta_{M_t}) = -E_{x_t}[\log(D(M_t(x_t)))] \tag{10}$$

The model is evaluated on the development data from target domain to tune hyperparameters.

B. Experimental Setup

1) Baseline: We implement two baselines. The first (*TF-IDF*) is cosine similarity computed from TF-IDF vector

Model	Pooling	Hidden Size	Dev				Test			
			MAP	MRR	P@1	P@5	MAP	MRR	P@1	P@5
BM25	-	-	52.0	66.0	51.9	42.1	56.0	68.0	53.8	42.5
CNN	mean	300	53.95	68.81	56.61	42.96	52.21	66.01	50.54	40.54
	max	600	54.83	**70.35**	58.2	42.86	53.34	66.59	51.61	41.4
LSTM	mean	400	58.73	**74.08**	62.96	46.67	55.66	69.63	55.91	43.76
	last	300	50.98	63.2	47.62	41.16	48.97	62.57	48.39	37.63
Bi-LSTM	mean	100	58.58	**74.73**	65.08	46.14	56.1	68.9	53.76	43.01
	last	100	40.8	51.75	36.51	32.17	42	50.35	33.33	31.51

TABLE I: Comparative results on the question retrieval task. BM25 is the baseline taken from Lei et al. (2015) and is the BM25 similarity measure provided by Apache Lucene. For the neural network models, the result is shown for the best hidden size $\in \{100...700\}$. The best MRR for each model is bolded.

Model	Pooling	Hidden size
CNN	max	600
LSTM	mean	400
Bi-LSTM	mean	100

TABLE II: The best hyperparameters found for each model.

of each question from the target domain.

The second baseline (DT) is a direct transfer of model without domain adaptation. We use the same network architecture (LSTM/CNN) as in Part III, but replace the pre-trained word embedding for Ubuntu only with the GloVe word embedding(Pennington et al., 2014) to account for vocabulary difference across domains. The model is re-trained. We have tried different hidden sizes, pooling methods and learning rates with multiple random starting points as done in Part III. We evaluate the model on development data from target domain.

2) Transfer with Domain Adaptation: We implement GRL and ADDA. For GRL, we vary hidden size of feature extractor (CNN/LSTM), hidden size of domain classifier, trade-off parameter in loss function, and learning rate.

ADDA training is carried out sequentially. First, we train a network for question similarity task on source domain and obtain feature mapping. Then we train adversarial network - a feature extractor for target domain (M_t) and a domain classifier(D) with source feature extractor(M_s) fixed. We alternate the training of M_t and D. We have tried several alternatives, such as training D for one epoch (going through all training batches) and then training M_t for one epoch; or training D for 5 batches and then training M for 5 batches. We tried same learning rates (0.01, 0.001) for M_t and D and different learning rates (smaller learning rate for D than for M_t to strengthen M_t). At the end of each epoch, we evaluate the model on the development set from target domain. The model training is terminated when after 25 consecutive epochs without performance improvement.

3) Evaluation Metrics: We use Area Under Curve (AUC) measure for model evaluation on target domain. AUC can be interpreted as the probability that, given a randomly selected positive example and a randomly selected negative example, the positive example is assigned a higher score by the model than the negative example. Since we have a large number of false negative pairs in our target development data, we use AUC(0.05): the AUC score when the false positive ratio is less than 0.05.

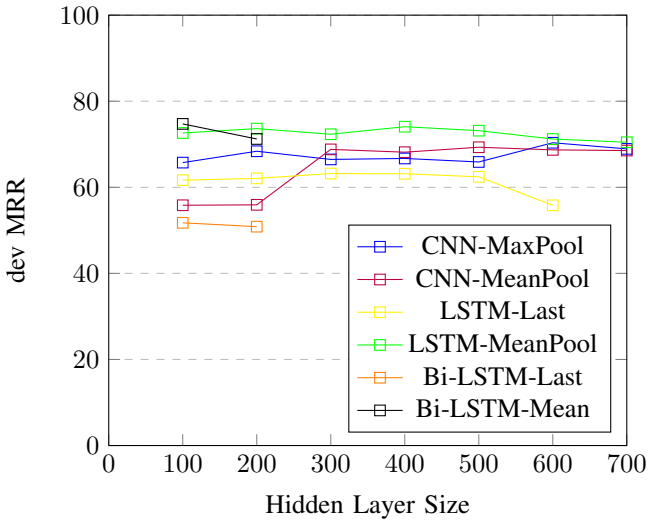

Fig. 3: dev MRR vs. hidden layer size. Only 100 and 200 hidden sizes were tried for bidirectional LSTM because of high memory usage for bigger sizes.

V. Results & Conclusion

A. Question Similarity

TABLE I shows the performance of the different neural network models on the question retrieval task. Most of the models achieved higher MRR than the baseline BM25. The results show that the model with the highest MRR is the bidirectional LSTM with mean pooling and hidden layer size of 100. The best configuration for each model is shown in TABLE II.

B. Transfer Learning

The first baseline for transfer learning is TF-IDF cosine similarity. The second baseline is a direct transfer of LSTM and CNN to target domain. Table 3 shows the AUC and AUC(0.05) on the dev/test data from target domain. The domain adaptation using gradient reversal achieves 8% improvement over direct transfer for LSTM and 5% improvement

Model	Method	Dev		Test	
		AUC	AUC(0.05)	AUC	AUC(0.05)
TF-IDF	-	0.96	0.72	0.96	0.74
CNN	DT	0.94	0.57	0.94	0.57
	DA	0.95	0.60	0.95	0.60
LSTM	DT	0.95	0.62	0.94	0.60
	DA	0.96	0.67	0.95	0.65
	ADDA	0.94	0.58	0.94	0.58

TABLE III: Comparative results on the domain adaptation task. The baseline is TF-IDF cosine similarity. DT is direct transfer, DA is domain adaptation using gradient reversal from Ganin and Lempitsky (2014), and ADDA is adversarial discriminative domain adaptation from Tzeng et al. (2017).

for CNN. We are not able to obtain good performance for ADDA and suspect that it requires longer training time and more parameter tuning (e.g. learning rates; number of times to train discriminator vs. feature extractor)

We have implemented neural network models for the question retrieval task and evaluated two approaches for domain adaptation, transferring a learned model using StackExchange AskUbuntu dataset to perform question retrieval on the StackExchagne Android dataset. We are able to reproduce results from Lei et al. (2015), getting similar performance for LSTM and CNN based on IR metrics. Both Ganin and Lempitsky (2014) and Tzeng et al. (2017) evaluate their methods on images, and we are able to apply their domain adaptation methods on text. Future work includes further exploration into adversarial techniques as well as evaluating current techniques on other types of datasets.

ACKNOWLEDGMENT

The authors would like to thank Professor Regina Barzilay and the course staff for MIT's Advanced Natural Language Processing (6.864) class.

REFERENCES

L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the Latent Topics for Question Retrieval in Community QA. *Ijcnlp*, pages 273–281, 2011. URL http://www.nlpr.ia.ac.cn/2011papers/gjhy/gh135.pdf.

X. Cao, G. Cong, B. Cui, C. S. Jensen, and C. Zhang. The use of categorization information in language models for question retrieval. *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, page 265, 2009. doi: 10.1145/1645953.1645989. URL http://portal.acm.org/citation.cfm?doid=1645953.1645989.

X. Cao, G. Cong, B. Cui, and C. S. Jensen. A Generalized Framework of Exploring Category Information for Question Retrieval in Community Question Answer Archives. *the International World Wide Web Conference 2010*, (December 2005):201–210, 2010. ISSN 10468188. doi: 10.1145/1772690.1772712.

C. dos Santos, L. Barbosa, D. Bogdanova, and B. Zadrozny. Learning Hybrid Representations to Retrieve Semantically Equivalent Questions. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, (January): 694–699, 2015. doi: 10.3115/v1/P15-2114. URL http://aclweb.org/anthology/P15-2114.

Y. Ganin and V. Lempitsky. Unsupervised Domain Adaptation by Backpropagation. (i), 2014. ISSN 1938-7228. doi: 10.1109/CVPR.2012.6247911. URL http://arxiv.org/abs/1409.7495.

I. Goodfellow. Tutorial: GANs. *arXiv*, 2016. ISSN 0253-0465. doi: 10.1001/jamainternmed.2016.8245.

R. Gopalan, R. N. Li, and R. Chellappa. Domain Adaptation for Object Recognition: An Unsupervised Approach. *2011 Ieee International Conference on Computer Vision (Iccv)*, pages 999–1006, 2011.

J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting Sample Selection Bias by Unlabeled Data. *Advances in Neural Information Processing Systems 19*, pages 601–608, 2007.

Z. Ji, F. Xu, and B. Wang. A category-integrated language model for question retrieval in community question answering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7675 LNCS:14–25, 2012. ISSN 03029743. doi: 10.1007/978-3-642-35341-3_2.

T. Lei, H. Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, A. Moschitti, and L. Marquez. Semi-supervised Question Retrieval with Gated Convolutions. 2015. URL http://arxiv.org/abs/1512.05726.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Sinno-Jialin-Pan. Domain Adaptation via Transfer Component Analysis_TNN2011. 22(2):199–210, 2011. doi: 10.1109/TNN.2010.2091281.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014. ISSN 09205691. doi: 10.1007/s10107-014-0839-0. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural.

D. Tang, B. Qin, and T. Liu. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September):1422–1432, 2015. ISSN 10495258. doi: 10.18653/v1/D15-1167. URL http://aclweb.org/anthology/D15-1167.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial Discriminative Domain Adaptation. 2017. ISSN 10495258. doi: 10.1109/CVPR.2017.316. URL http://arxiv.org/abs/1702.05464.

G. Zhou, L. Cai, J. Zhao, and K. Liu. Phrase-based translation model for question retrieval in community question answer archives. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 653–662, 2011. URL http://dl.acm.org/citation.cfm?id=2002472.2002555.

G. Zhou, Y. Liu, F. Liu, D. Zeng, and J. Zhao. Improving question retrieval in community question answering using world knowledge. *IJCAI International Joint Conference on Artificial Intelligence*, pages 2239–2245, 2013. ISSN 10450823.