Adversarial Domain Adaptation for Duplicate Question Detection

Darsh J Shah[1], **Tao Lei**[2], **Alessandro Moschitti**[3]*, **Salvatore Romeo**[4], **Preslav Nakov**[4]
[1]MIT CSAIL, Cambridge, MA, USA
[2]ASAPP Inc., New York, NY, USA
[3]Amazon, Manhattan Beach, CA, USA
[4]Qatar Computing Research Institute, HBKU, Doha, Qatar
darsh@csail.mit.edu, tao@asapp.com, amosch@amazon.com
sromeo@qf.org.qa, pnakov@qf.org.qa

Abstract

We address the problem of detecting duplicate questions in forums, which is an important step towards automating the process of answering new questions. As finding and annotating such potential duplicates manually is very tedious and costly, automatic methods based on machine learning are a viable alternative. However, many forums do not have annotated data, i.e., questions labeled by experts as duplicates, and thus a promising solution is to use domain adaptation from another forum that has such annotations. Here we focus on adversarial domain adaptation, deriving important findings about when it performs well and what properties of the domains are important in this regard. Our experiments with *StackExchange* data show an average improvement of 5.6% over the best baseline across multiple pairs of domains.

1 Introduction

Recent years have seen the rise of community question answering forums, which allow users to ask questions and to get answers in a collaborative fashion. One issue with such forums is that duplicate questions easily become ubiquitous as users often ask the same question, possibly in a slightly different formulation, making it difficult to find the best (or one correct) answer (Hoogeveen et al., 2018; Lai et al., 2018). Many forums allow users to signal such duplicates, but this can only be done after the duplicate question has already been posted and has possibly received some answers, which complicates merging the question threads. Discovering possible duplicates at the time of posting is much more valuable from the perspective of both (*i*) the forum, as it could prevent a duplicate from being posted, and (*ii*) the users, as they could get an answer immediately.

Duplicate question detection is a special case of the more general problem of question-question similarity. The latter was addressed using a variety of textual similarity measures, topic modeling (Cao et al., 2008; Zhang et al., 2014), and syntactic structure (Wang et al., 2009; Filice et al., 2016; Da San Martino et al., 2016; Barrón-Cedeño et al., 2016; Filice et al., 2017). Another approach is to use neural networks such as feed-forward (Nakov et al., 2016a), convolutional (dos Santos et al., 2015; Bonadiman et al., 2017; Wang et al., 2018), long short-term memory (Romeo et al., 2016), and more complex models (Lei et al., 2016; Nicosia and Moschitti, 2017; Uva et al., 2018; Joty et al., 2018; Zhang and Wu, 2018). Translation models have also been popular (Zhou et al., 2011; Jeon et al., 2005; Guzmán et al., 2016a,b).

The above work assumes labeled training data, which exists for question-question similarity, e.g., from SemEval-2016/2017 (Agirre et al., 2016; Nakov et al., 2016b, 2017), and for duplicate question detection, e.g., SemEval-2017 task 3 featured four StackExchange forums, *Android*, *English*, *Gaming*, and *Wordpress*, from CQADupStack (Hoogeveen et al., 2015, 2016). Yet, such annotation is not available for many other forums, e.g., the *Apple* community on StackExchange.

In this paper, we address this lack of annotation using adversarial domain adaptation (ADA) to effectively use labeled data from another forum. Our contributions can be summarized as follows:

- we are the first to apply adversarial domain adaptation to the problem of duplicate question detection across different domains;[1]
- on the StackExchange family of forums, our model outperforms the best baseline with an average relative improvement of 5.6% (up to 14%) across all domain pairs.

*Work conducted while the author was at QCRI.

[1]The code and the data are available at the following link: http://github.com/darsh10/qra_code

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1056–1063
Brussels, Belgium, October 31 - November 4, 2018.

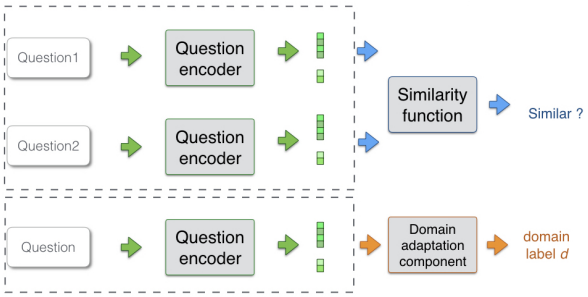

Figure 1: Our cross-domain question-question similarity model. The Question encoder is common for the questions from the source domain and from the target domain. The model and its training procedure are described in Section 2.

- we study when transfer learning performs well and what properties of the domains are important in this regard; and
- we show that adversarial domain adaptation can be efficient even for unseen target domains, given some similarity of the target domain with the source one and with the regularizing adversarial domain.

Adversarial domain adaptation (ADA) was proposed by Ganin and Lempitsky (2015), and was then used for NLP tasks such as sentiment analysis and retrieval-based question answering (Chen et al., 2016; Ganin et al., 2016; Li et al., 2017; Liu et al., 2017; Yu et al., 2018; Zhang et al., 2017), including cross-language adaptation (Joty et al., 2017) for question-question similarity.[2]

The rest of this paper is organized as follows: Section 2 presents our model, its components, and the training procedure. Section 3 describes the datasets we used for our experiments, stressing upon their nature and diversity. Section 4 describes our adaptation experiments and discusses the results. Finally, Section 5 concludes with possible directions for future work.

2 Method

Our ADA model has three components: (*i*) question encoder, (*ii*) similarity function, and (*iii*) domain adaptation component, as shown in Figure 1.

The encoder E maps a sequence of word tokens $x = (x_1, .., x_n)$ to a dense vector $\mathbf{v} = E(x)$. The similarity function f takes two question vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, and predicts whether the corresponding questions are duplicates.

[2]Prior work on cross-language adaptation for question-question similarity used cross-language tree kernels (Da San Martino et al., 2017).

The domain classifier g takes a question vector $\mathbf{v}$ and predicts whether the question is from the source or from the target domain. We train the encoder not only to do well on the task for the source data, but also to fool the domain classifier, as shown in Algorithm 1. We describe the design choices considered for our domain adaptation model in the following two subsections.

2.1 Question Similarity Function

We consider two options for our similarity function $f(\mathbf{v}_1, \mathbf{v}_2)$:
(*i*) a logistic function that computes the probability that two questions are similar/duplicates, which is trained with the cross-entropy loss:

$$\text{sigmoid}\left(\mathbf{W}^\top(\mathbf{v}_1 \odot \mathbf{v}_2) + \mathbf{b}\right)$$

where $\odot$ is an element-wise vector product between unit encodings of questions;

(*ii*) a simple cosine similarity function, i.e., $cosine(\mathbf{v}_1, \mathbf{v}_2)$, trained using the pairwise hinge loss with a margin m:

$$\sum_i max(\{(1-y^i)f(\mathbf{v}_1^i, \mathbf{v}_2^i)+m-y^i f(\mathbf{v}_1^i, \mathbf{v}_2^i)\}, 0)$$

Our experiments reported in Table 3 show that the $cosine$ similarity function performs far better.

2.2 Domain Adaptation Components

The adversarial component is responsible for reducing the difference between the source and the target domain distributions. There are two common approaches to achieve this: (*i*) classification-based (Ganin and Lempitsky, 2015) and (*ii*) Wasserstein (Arjovsky et al., 2017).

The main difference between them is in the way the domain discrepancy loss is computed. In the classification-based approach, the adversarial component is a classifier trained to correctly predict the domain (source vs. target) of the input question. In contrast, the question encoder is optimized to confuse the domain classifier, which, as a result, encourages domain invariance. Arjovsky and Bottou (2017) showed that this adversarial optimization process resembles minimizing the Jenson-Shannon (JS) divergence between the source P_s and the target distribution P_t:

$$JS(P_s, P_t) = KL(P_s, P_m) + KL(P_g, P_m)$$

where $P_m = (P_s + P_t)/2$ and KL is the Kullback-Leibler divergence.

Algorithm 1: Training Procedure

Input: source data X^s; target data X^t
Hyper-parameters: learning rates α_1, α_2; batch size m; adversarial importance λ
Parameters to be trained: question encoder θ_e, question similarity classifier θ_s and domain classifier θ_d
Similarity classification loss L_c is either the cross-entropy loss or hinge loss, described in Section 2.1
Adversarial loss L_d, described in Section 2.2
repeat
  for `each batch` **do**
    Construct a sub-batch of similar and dissimilar question pairs from the annotated source data $\{(x^s_{i_1}, x^s_{i_2}), y^s_i\}_{i=1}^m$
    Calculate the classification loss L_c using θ_e and θ_s for this sub-batch
    Construct a sub-batch of questions $\{x^s_i, x^t_j\}_{i=1}^m$ from the corpora of source and target domains
    Calculate the domain discrepancy loss L_d using θ_e and θ_d for this sub-batch
    Total loss $L = L_c - \lambda L_d$
    $\theta_e = \theta_e - \alpha_1 \nabla_{\theta_e} L$
    $\theta_s = \theta_s - \alpha_1 \nabla_{\theta_s} L$
    $\theta_d = \theta_d + \alpha_2 \nabla_{\theta_d} L$
  end for
until θ_e, θ_s and θ_d converge

In contrast, the Wasserstein method attempts to reduce the approximated Wasserstein distance (also known as *Earth Mover's Distance*) between the distributions for the source and for the target domain as follows:

$$W(P_s, P_t) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim P_s}[f(x)] - \mathbb{E}_{x \sim P_t}[f(x)]$$

where f is a Lipchitz-1 continuous function realized by a neural network.

Arjovsky et al. (2017) have shown that the Wasserstein method yields more stable training for computer vision tasks.

2.3 Training Procedure

Algorithm 1 describes the procedure to train the three components of our model. Adversarial training needs two kinds of training data: (*i*) annotated question pairs from the source domain, and (*ii*) unlabeled questions from the source and the target domains.

The question encoder is trained to perform well on the source domain using the similarity classification loss L_c. In order to enforce good performance on the target domain, the question encoder is simultaneously trained to be incapable in discriminating between question pairs from the source vs. the target domain. This is done through the domain classification loss L_d.

Dataset	Questions	Duplicates	Train	Dev	Test
AskUbuntu	257,173	27,289	9,106	1,000	1,000
SuperUser	343,033	11,407	9,106	1,000	1,000
Apple	80,466	2,267	–	1,000	1,000
Android	42,970	2,371	–	1,000	1,000
Sprint	31,768	23,826	9,100	1,000	1,000
Quora	537,211	149,306	9,100	–	–

Table 1: Statistics about the datasets. The table shows the number of question pairs that have been manually marked as similar/duplicates by the forum users (i.e., positive pairs). We further add 100 negative question pairs per duplicate question by randomly sampling from the full corpus of questions.

3 Datasets

The datasets we use can be grouped as follows:

- **Stack Exchange** is a family of technical community support forums. We collect questions (composed by title and body) from the XML dumps of four forums: *AskUbuntu*, *SuperUser*, *Apple*, and *Android*. Some pairs of similar/duplicate questions in these forums are marked by community users.

- **Sprint FAQ** is a newly crawled dataset from the Sprint technical forum website. It contains a set of frequently asked questions and their paraphrases, i.e., three similar questions, paraphrased by annotators.

- **Quora** is a dataset of pairs of similar questions asked by people on the Quora website. They cover a broad set of topics touching upon philosophy, entertainment and politics.

Note that these datasets are quite heterogeneous: the *StackExchange* forums focus on specific technologies, where questions are informal and users tend to ramble on about their issues, the *Sprint FAQ* forum is technical, but its questions are concise and shorter, and the *Quora* forum covers many different topics, including non-technical.

Statistics about the datasets are shown in Table 1. Moreover, in order to quantify the differences and the similarities, we calculated the fraction of unigrams, bigrams and trigrams that are shared by pairs of domains. Table 2 shows statistics about the n-gram overlap between *AskUbuntu* or *Quora* as the source and all other domains as the target. As one might expect, there is a larger overlap within the *StackExchange* family.

Source	Target	Unigrams	Bigrams	Trigrams
AskUbuntu	Android	0.989	0.926	0.842
	Apple	0.991	0.926	0.853
	SuperUser	0.990	0.921	0.822
	Sprint	0.959	0.724	0.407
	Quora	0.922	0.696	0.488
Quora	AskUbuntu	0.949	0.647	0.326
	Apple	0.969	0.721	0.426
	Android	0.973	0.762	0.473
	SuperUser	0.958	0.663	0.338
	Sprint	0.942	0.647	0.310

Table 2: Proportion of n-grams that are shared between the source and the target domains.

4 Experiments and Evaluation

4.1 Experimental Setup

Baselines We compare our ADA model to the following baselines: (*a*) *direct transfer*, which directly applies models learned from the source to the target domain without any adaptation; and (*b*) the standard unsupervised *BM25* (Robertson and Zaragoza, 2009) scoring provided in search engines such as Apache Lucene (McCandless et al., 2010).

Models We use a bi-LSTM (Hochreiter and Schmidhuber, 1997) encoder that operates on 300-dimensional GloVe word embeddings (Pennington et al., 2014), which we train on the combined data from all domains. We keep word embeddings fixed in our experiments. For the adversarial component, we use a multi-layer perceptron.

Evaluation Metrics As our datasets may contain some duplicate question pairs, which were not discovered and thus not annotated, we end up having false negatives. Metrics such as MAP and MRR are not suitable in this situation. Instead, we use AUC (area under the curve) to evaluate how well the model ranks positive pairs vs. negative ones. AUC quantifies how well the true positive rate (tpr) grows at various false positive rates (fpr) by calculating the area under the curve starting from $fpr = 0$ to $fpr = 1$. We compute the area integrating the false positive rate (x-axis) from 0 up to a threshold t, and we normalize the area to $[0, 1]$. This score is known as AUC(t). It is more stable than MRR and MAP in our case when there could be several false negatives.[3]

[3]For illustration, say of 100 candidates, 2 false negatives are ranked higher than the correct pair, the AUC score drops by 3 points (linear drop), as compared to the 66.67 point drop for MRR. We can avoid the expensive manually tagging of the negative pairs for experiments by using the AUC score.

Adaptation	Similarity	AUC(0.05)	AUC(0.1)
—	Sigmoid	0.431	0.557
—	Cosine	0.692	0.782
Classification	Cosine	0.791	0.862
Wasserstein	Cosine	0.795	0.869

Table 3: Duplicate question detection: direct transfer vs. adversarial domain adaptation from *AskUbuntu* to *Android*.

Source	Target	Direct	BM25	Adv.
AskUbuntu	Android	0.692	0.681	**0.790**
	Apple	0.828	0.747	**0.855**
	SuperUser	0.908	0.765	**0.911**
	Sprint	0.917	**0.956**	0.937
SuperUser	AskUbuntu	0.730	0.644	**0.796**
	Apple	0.828	0.747	**0.861**
	Android	0.770	0.681	**0.790**
	Sprint	0.928	**0.956**	0.932

Table 4: Domain adaptation for the *StackExchange* source-target domain pairs when using the *Direct* approach, *BM25*, and our adaptation model, measured with AUC(0.05).

4.2 Choosing the Model Components

Model Selection We select the best components for our domain adaptation model via experimentation on the *AskUbuntu–Android* domain pair. Then, we apply the model with the best-performing components across all domain pairs.

Hyperparameters We fine-tune the hyper-parameters of all models on the development set for the target domain.

Similarity Function Table 3 shows the AUC at 0.05 and 0.1 for different models of question similarity, training on *AskUbuntu* and testing on *Android*. The first row shows that using cosine similarity with a hinge loss yields much better results than using a cross-entropy loss. This is likely because (*i*) there are some duplicate question pairs that were not tagged as such and that have come up as negative pairs in our training set, and the hinge loss deals with such outliers better. (*ii*) The cosine similarity is domain-invariant, while the weights of the feed-forward network of the softmax layers capture source-domain features.

Domain Adaptation Component We can see that the Wasserstein and the classification-based methods perform very similarly, after proper hyper-parameter tuning. However, Wasserstein yields better stability, achieving an AUC variance 17 times lower than the one for classification across hyper-parameter settings. Thus, we chose it for all experiments in the following subsections.

Source	Target	Direct	BM25	Adv.
Sprint	AskUbuntu	0.615	**0.644**	0.615
	Apple	0.719	**0.747**	0.728
	Android	0.627	**0.681**	0.648
	Sprint	0.977	0.956	–
	SuperUser	0.795	0.765	**0.795**
Quora	AskUbuntu	0.446	**0.644**	0.446
	Apple	0.543	**0.747**	0.543
	Android	0.443	**0.681**	0.460
	Sprint	0.786	**0.956**	0.794
	SuperUser	0.624	**0.765**	0.649

Table 5: Domain adaptation results when using *Sprint* and *Quora* as the source domains with the *Direct* approach, *BM25*, and our adaptation model, measured with AUC(0.05).

4.3 When Does Adaptation Work Well?

Tables 4 and 5 study the impact of domain adaptation when applied to various source-target domain pairs, using the *Direct* approach, *BM25*, and our adaptation model. We can make the following observations:

- For almost all source–target domain pairs from the *StackExchange* family, domain adaptation improves over both baselines, with an average relative improvement of 5.6%. This improvement goes up to 14% for the *AskUbuntu–Android* source–target domain pair.

- Domain adaptation on the *Sprint* dataset performs better than direct transfer, but it is still worse than *BM25*.

- Domain adaptation from *Quora* performs the worst, with almost no improvement over direct transfer, which is far behind *BM25*.

- The more similar the source and the target domains, the better our adaptation model performs.

Table 2 shows that *AskUbuntu* has high similarity to other *StackExchange* domains, lower similarity to *Sprint*, and even lower similarity to *Quora*. The Pearson coefficient (Myers et al., 2010) between the n-gram fractions and the domain adaptation effectiveness for unigrams, bigrams and trigrams is 0.57, 0.80 and 0.94, respectively, which corresponds to moderate-to-strong positive correlation. This gives insight into how simple statistics can predict the overall effectiveness of domain adaptation.

Pivot\Target	SuperUser	Apple	Android
SuperUser	0.911	0.827	0.678
Apple	0.900	0.855	**0.711**
Android	0.904	**0.843**	0.790
Quora	0.906	0.815	0.673
Direct	0.908	0.828	0.692

Table 6: AUC(0.05) of ADA to unseen domains, with *AskUbuntu* as a source.

4.4 Adapting to Unseen Domains

We also experiment with domain adaptation to a target domain that was not seen during training (even adversarial training). We do so by training to adapt to a pivot domain different from the target. Table 6 shows that this yields better AUC compared to *direct transfer* when using *Apple* and *Android* as the pivot/target domains. We hypothesize that this is due to *Apple* and *Android* being closely related technical forums for iOS and Android devices. This sheds some light on the generality of adversarial regularization.

5 Conclusion and Future Work

We have applied and analyzed adversarial methods for domain transfer for the task of duplicate question detection; to the best of our knowledge, this is the first such work. Our experiments suggest that (*i*) adversarial adaptation is rather effective between domains that are similar, and (*ii*) the effectiveness of adaptation is positively correlated with the n-gram similarity between the domains.

In future work, we plan to develop better methods for adversarial adaptation based on these observations. One idea is to try source-pivot-target transfer, similarly to the way this is done for machine translation (Wu and Wang, 2007). Another promising direction is to have an attention mechanism (Luong et al., 2015) for question similarity which can be adapted across domains.[4]

Acknowledgments

This research was carried out in collaboration between the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) and the Qatar Computing Research Institute (QCRI), HBKU.

[4]In our experiments, we found that using attention was lowering the adaptation performance. Upon analysis, we found that adversarial techniques alone were not enough to make the attention weights domain-invariant.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, pages 497–511, San Diego, CA, USA.

Martín Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, Toulon, France.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17, pages 214–223, Sydney, NSW, Australia.

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad A. Al Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 2016. ConvKN at SemEval-2016 Task 3: Answer and question selection for question answering on Arabic and English fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 896–903, San Diego, CA, USA.

Daniele Bonadiman, Antonio Uva, and Alessandro Moschitti. 2017. Effective shared representations with multitask learning for community question answering. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '17, pages 726–732, Valencia, Spain.

Yunbo Cao, Huizhong Duan, Chin-Yew Lin, Yong Yu, and Hsiao-Wuen Hon. 2008. Recommending questions using the MDL-based tree cut model. In *Proceedings of the International Conference on World Wide Web*, WWW '08, pages 81–90, Beijing, China.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Salvatore Romeo, Antonio Uva, and Alessandro Moschitti. 2016. Learning to re-rank questions in community question answering using advanced features. In *Proceedings of the ACM Conference on Information and Knowledge Management*, CIKM '16, pages 1997–2000, Indianapolis, IN, USA.

Giovanni Da San Martino, Salvatore Romeo, Alberto Barroón-Cedeño, Shafiq Joty, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2017. Cross-language question re-ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1145–1148, Tokyo, Japan.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 1116–1123, San Diego, CA, USA.

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2017. KeLP at SemEval-2017 task 3: Learning pairwise patterns in community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, pages 327–334, Vancouver, Canada.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning*, ICML '15, pages 1180–1189, Lille, France.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030.

Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2016a. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 460–466, Berlin, Germany.

Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2016b. MTE-NN at SemEval-2016 Task 3: Can machine translation evaluation help community question answering? In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 887–895, San Diego, CA, USA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Doris Hoogeveen, Karin Verspoor, and Timothy Baldwin. 2016. CQADupStack: Gold or silver? In *Proceedings of the SIGIR Workshop on Web Question Answering Beyond Factoids*, WebQA '16, Pisa, Italy.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. CQADupStack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, pages 3:1–3:8, Parramatta, NSW, Australia.

Doris Hoogeveen, Li Wang, Timothy Baldwin, and Karin M. Verspoor. 2018. Web forum retrieval and text analytics: A survey. *Foundations and Trends in Information Retrieval*, 12(1):1–163.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the ACM Conference on Information and Knowledge Management*, CIKM '05, pages 84–90, Bremen, Germany.

Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2018. Joint multitask learning for community question answering using task-specific embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, Brussels, Belgium.

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. 2017. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, CoNLL '17, pages 226–237, Vancouver, Canada.

Tuan Manh Lai, Trung Bui, and Sheng Li. 2018. A review on deep learning techniques applied to answer selection. In *Proceedings of the International Conference on Computational Linguistics*, COLING '18, pages 2132–2144, Santa Fe, NM, USA.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '16, pages 1279–1289, San Diego, CA, USA.

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 2237–2243, Melbourne, Australia.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1–10, Vancouver, Canada.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 1412–1421, Lisbon, Portugal.

Michael McCandless, Erik Hatcher, and Otis Gospodnetic. 2010. *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co.

Jerome L Myers, Arnold Well, and Robert Frederick Lorch. 2010. *Research design and statistical analysis*. Routledge.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '17, pages 525–545, Vancouver, Canada.

Preslav Nakov, Lluís Màrquez, and Francisco Guzmán. 2016a. It takes three to tango: Triangulation approach to answer ranking in community question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 1586–1597, Austin, TX, USA.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016b. SemEval-2016 task 3: Community question answering. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, pages 525–545, San Diego, CA, USA.

Massimo Nicosia and Alessandro Moschitti. 2017. Accurate sentence matching with hybrid Siamese networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 2235–2238, Singapore.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1532–1543, Doha, Qatar.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, and James Glass. 2016. Neural attention for learning to rank questions in community question answering. In *Proceedings of the 26th International Conference on Computational Linguistics*, COLING '16, pages 1734–1745, Osaka, Japan.

Cicero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, pages 694–699, Beijing, China.

Antonio Uva, Daniele Bonadiman, and Alessandro Moschitti. 2018. Injecting relational structural representation in neural networks for question similarity. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL '18, pages 285–291, Melbourne, Australia.

Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. 2009. A syntactic tree matching approach to finding similar questions in community-based QA services. In *Proceedings of the 32nd International ACM SIGIR*

Conference on Research and Development in Information Retrieval, SIGIR '09, pages 187–194, Boston, MA, USA.

Pengwei Wang, Lei Ji, Jun Yan, Dejing Dou, Nisansa De Silva, Yong Zhang, and Lianwen Jin. 2018. Concept and attention-based CNN for question retrieval in multi-view learning. *ACM Trans. Intell. Syst. Technol.*, 9(4):41:1–41:24.

Hua Wu and Haifeng Wang. 2007. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181.

Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *Proceedings of the Conference on Web Search and Data Mining*, WSDM '18, pages 682–690, Marina Del Rey, CA, USA.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, CIKM '14, pages 371–380, Shanghai, China.

Minghua Zhang and Yunfang Wu. 2018. An unsupervised model with attention autoencoders for question retrieval. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '18, New Orleans, LA, USA.

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

Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, ACL '11, pages 653–662, Portland, OR, USA.