Cross-Domain Review Helpfulness Prediction based on Convolutional Neural Networks with Auxiliary Domain Discriminators

Cen Chen[1], **Yinfei Yang**[2], **Jun Zhou**[1], **Xiaolong Li**[1], **Forrest Sheng Bao**[3]
[1]Ant Financial Services Group, Hangzhou, China
[2]1600 Amphitheatre Pkwy, Mountain View, CA 94043 *
[3]Iowa State University, Ames, IA 50011
{chencen.cc,jun.zhoujun,xl.li}@antfin.com
{yangyin7,forrest.bao}@gmail.com

Abstract

With the growing amount of reviews in e-commerce websites, it is critical to assess the helpfulness of reviews and recommend them accordingly to consumers. Recent studies on review helpfulness require plenty of labeled samples for each domain/category of interests. However, such an approach based on close-world assumption is not always practical, especially for domains with limited reviews or the "out-of-vocabulary" problem. Therefore, we propose a convolutional neural network (CNN) based model which leverages both word-level and character-based representations. To transfer knowledge between domains, we further extend our model to jointly model different domains with auxiliary domain discriminators. On the Amazon product review dataset, our approach significantly outperforms the state of the art in terms of both accuracy and cross-domain robustness.

1 Introduction

Product reviews significantly help consumers finalize their purchasing decisions. With online reviews being ubiquitous, it is critical to examine the quality of reviews and present consumers more useful information. Both academia and industry have drawn close attention to the task of review helpfulness prediction (Liu et al., 2017a; Yang et al., 2015, 2016; Martin and Pu, 2014).

Recent studies on review helpfulness prediction have been shown effective by using hand-crafted features. For example, semantic features like LIWC, INQUIRER, and GALC (Yang et al., 2015; Martin and Pu, 2014), aspect- (Yang et al., 2016) and argument-based (Liu et al., 2017a) features. However, those methods require a large amount of labeled samples which is not always practical and yields models limited to product domains/categories of interests. For example, the "Electronics" category used in our experiment from Amazon.com Review Dataset (McAuley and Leskovec, 2013) has more than 354k labeled reviews, while the "Watches" category has under 10k. For domains with limited data, labeled samples may be too few to build good estimators and the "out-of-vocabulary" (OOV) problem is often observed.

To alleviate the aforementioned issues, in this work, we propose an end-to-end approach for review helpfulness prediction requiring no prior knowledge nor manual feature crafting. In recent years, convolutional neural networks (CNNs), able to extract deep features from raw text contents, have demonstrated remarkable results in many tasks of natural language processing, for its high efficiency and performance comparable to Recurrent Neural Networks (RNNs) (Kim, 2014; Zhang et al., 2015). We thus employ CNNs as the basis of this work. As character-level representations are notably beneficial for alleviating the OOV problem for tasks such as text classification and machine translation (Ballesteros et al., 2015; Ling et al., 2015; Kim et al., 2016; Lee et al., 2017), we specifically enrich the word-level representation of CNNs by adding character-based representation. Experiments show that our CNN-based method significantly outperforms those using hand-crafted features and yields better results than the ensemble models.

To tackle the problem of insufficient data in some domains, we develop a cross-domain transfer learning (TL) approach to leverage knowledge from a domain with sufficient data. It is worth noting that, existing studies on this task only focus on a single product category or largely ignore the inter-domain correlations. Previous works also show that some features are domain-specific while others are sharable across domains. For example, image quality features are only useful for categories covering products like cameras (Yang et al.,

* Yinfei Yang is now with Google.

Proceedings of NAACL-HLT 2018, pages 602–607
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

2016), while semantic features and argument-based features usually work for all domains (Yang et al., 2015; Liu et al., 2017a). Thus it is important for a TL approach to learn shared features for different domains. A typical TL model uses both a shared neural network (NN) and domain-specific NNs to derive shared and domain-specific features (Ganin et al., 2016; Taigman et al., 2017). Recently, Liu et al. (2017b) and Chen et al. (2017) apply adversarial loss and domain discriminators to specific shared models using RNNs for text classification and word segmentation tasks, respectively. Inspired by them, we study the cross-domain review helpfulness task with both adversarial loss and domain discriminators in a specific shared framework.

In a nutshell, our main novelty is in the first end-to-end cross-domain model for review helpfulness prediction. Our model consists of two components: a feature transformation network (CNN) to represent the input reviews and a transfer learning module to adapt domain knowledge. In addition, shared and specific-shared features are confined with adversarial and domain discrimination losses. Extensive experiments show that our model is able to transfer knowledge between domains, and outperforms the state of the arts.

The remainder of the paper is organized as follows. Section 2 formally defines the problem and presents our model. Section 3 illustrates the effectiveness of the proposed model in the experiments. Section 4 presents related work, and finally Section 5 concludes our paper.

2 Model

We define review helpfulness prediction as a regression task that predicts the helpfulness score of a given review. The ground truth of helpfulness is determined using the "a of b approach": a of b users think a review is helpful.

Formally, we consider a cross-domain review helpfulness prediction task where we have a set of labeled reviews from a source domain and a target domain. We seek to transfer knowledge from a source domain with adequate data to train a better model for a target domain, which has relatively insufficient amount of data. For a review $\mathbf{X}$, our goal is to predict its helpfulness score y.

As shown in Figure 1, our base model is a multi-granularity CNN, which combines both word-level and character-level representations.

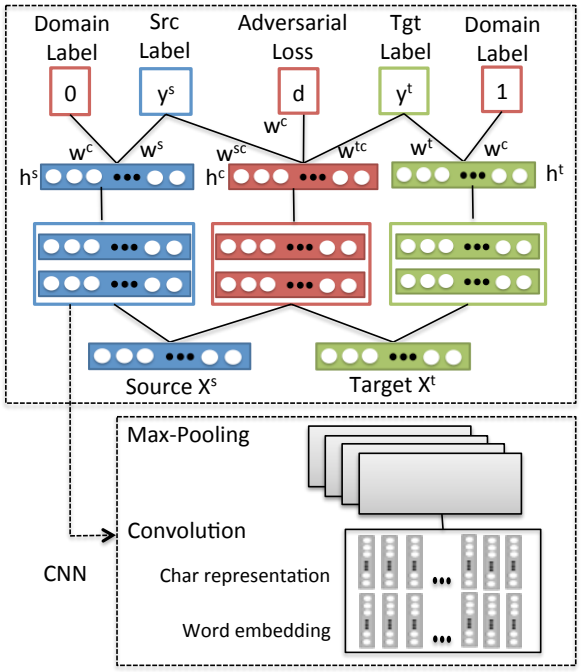

Figure 1: Our proposed end-to-end cross-domain model for review helpfulness prediction.

2.1 CNN with Character Representations

In many applications, such as text classification (Bojanowski et al., 2017) and machine reading comprehension (Seo et al., 2016), it is beneficial to enrich word embeddings with subword information. Inspired by that, we use a character embedding layer to enrich word representations.

Let $\mathbf{X}$ be a review, consisting of a sequence of words $(x_1, x_2, \ldots, x_m)$. Following the CNN model in (Kim, 2014), for words in a review $\mathbf{X}$, we first lookup the embeddings of all words $(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m)$ from an embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times l}$ where $|\mathcal{V}|$ is the vocabulary size and l is the embedding dimension.

The characters of the i-th word x_i are embedded into vectors and then fed into a convolutional layer and a max-pooling layer to obtain a fixed-sized vector $\text{CharEmb}(x_i)$. This vector is concatenated with the original word embedding $\mathbf{e}_i$ to form a new word embedding. This representation is advantageous in two folds: it helps group words with shared subwords, and it alleviates the OOV problem. Hence, we obtain a review's final representation by concatenating the embeddings of words in the review: $\mathbf{e}_X = [\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3, \ldots, \mathbf{e}'_m]$ where $\mathbf{e}'_i = \text{CharEmb}(x_i) \oplus \mathbf{e}_i, \forall i \in [1..m]$, $\mathbf{e}'_i$ is a column vector, and $\oplus$ is a stacking operator.

Next, we stack two 2-D convolutional layers

and two 2-D max-pooling layers on the matrix $\mathbf{e}_X$ to obtain the hidden representation $\mathbf{h}_X$. Multiple filters are used here. For each filter, we obtain a hidden representation:

$$\mathbf{g}_f = \text{MaxPool}(\text{Conv}(\mathbf{e}_X, \text{filterSize} = [f, l, c]))$$

where $f \in \{2, 3, 4, 5\}$ is window size, l is embedding dimension, c is channel size, $\text{Conv}(\cdot)$ represents a convolution layer, $\text{MaxPool}(\cdot)$ is a max-pooling layer. All the representations are then concatenated to form the final representation $\mathbf{h}_X$, i.e., $\mathbf{h}_X = [\mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4, \mathbf{g}_5]$.

In all, for each input $\mathbf{X}$, our CNN model outputs a hidden feature representation $h_X = \text{CNN}(\mathbf{X})$.

2.2 Knowledge Transfer with Domain Discriminators

A typical transfer learning framework is to use both a shared neural network and domain-specific neural networks to learn shared and domain-specific features (Liu et al., 2017b). In our model, we use a shared CNN and domain-specific CNNs to derive shared features $\mathbf{h^c}$ and domain-specific features $\mathbf{h^s}$ and $\mathbf{h^t}$. The domain-specific output layers are defined as:

$$\hat{y}^k = \begin{cases} \sigma(\mathbf{W^{sc}h^c} + \mathbf{W^s h^s} + \mathbf{b^s}), \text{ if } k = 0 \\ \sigma(\mathbf{W^{tc}h^c} + \mathbf{W^t h^t} + \mathbf{b^t}), \text{ if } k = 1 \end{cases}$$

where $k \in \{0, 1\}$ is the domain label indicating whether a data instance is from the source domain (i.e., $k = 0$) or the target domain (i.e., $k = 1$). $\mathbf{W^{sc}}$, $\mathbf{W^{tc}}$, $\mathbf{W^s}$, 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. The $\sigma(\cdot)$ represents the sigmoid function.

Recent studies (Ganin et al., 2016; Taigman et al., 2017; Liu et al., 2017b) consider to apply domain discriminators on shared features to prevent domain-specific features from creeping into shared feature space. The main idea of using a domain discriminator $p(d \mid \mathbf{h^c})$ is to predict the domain label d on the shared features $\mathbf{h^c}$. Here the domain discriminator is defined as a fully connected layer with weights $\mathbf{W}^c$ and bias vector $\mathbf{b}^c$:

$$p(d \mid \mathbf{h^c}) = \text{softmax}(\mathbf{W^c h^c} + \mathbf{b^c}).$$

Since the goal is to encourage the shared feature space indiscriminate across two domains, we define the adversarial loss L_{adv} as:

$$L_{adv} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{1} p(d = k|\mathbf{h}_i^c) \log p(d = k|\mathbf{h}_i^c).$$

where $\mathbf{h}_i^c$ is the derived shared features from an input $\mathbf{X}_i$.

Furthermore, to encourage the specific feature space to discriminate 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 the source domain and L_t for the target domain:

$$L_s = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{k=0}^{1} \mathbb{I}^{(d_i=k)} \log p(d = k|\mathbf{h}_i^s).$$

$$L_t = -\frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{k=0}^{1} \mathbb{I}^{(d_i=k)} \log p(d = k|\mathbf{h}_i^t).$$

where $\mathbb{I}^{(d_i=k)}$ is an indicator function set to 1 when $d_i = k$ holds, or 0 otherwise, and $\mathbf{h}_i^s$ and $\mathbf{h}_i^t$ are the derived domain-specific features from an input $\mathbf{X}_i$ from source and target domains respectively.

Nevertheless, studies in (Bousmalis et al., 2016; Liu et al., 2017b) show that adding orthogonality constraints on learned shared features $\mathbf{H^c}$ and specific features $\mathbf{H^k}$ for each domain $k \in \{s, t\}$ can help learn domain-invariant features. We thus adopt the constraint $L_{orth} = \sum_{k \in \{s,t\}} \mathbf{H^c}^\top \mathbf{H^k}$ in our model. $\mathbf{H^c}$ and $\mathbf{H^k}$ are obtained by stacking the hidden features from all the input instances.

Finally, we obtain a combined loss as follows:

$$\mathcal{L} = \sum_{\mathbf{k} \in \mathbf{s}, \mathbf{t}} -\frac{1}{n_\mathbf{k}} \sum_{j=1}^{n_\mathbf{k}} \frac{1}{2}(y_j^k - \hat{y}_j^k)^2 + \frac{\lambda_1}{2} L_{adv} + \frac{\lambda_2}{2} L_s + \frac{\lambda_3}{2} L_t + \frac{\lambda_4}{2} L_{orth} + \frac{\lambda_5}{2} ||\mathbf{\Theta}||_F^2.$$

where all λ's are weights for different losses, and $\mathbf{\Theta}$ denotes model parameters.

3 Experiments

Following previous work (Yang et al., 2015, 2016), experiments are done on reviews from five categories of products in Amazon review dataset (McAuley and Leskovec, 2013). Data statistics are summarized in Table 1.

The empirical study is done in two steps. Without TL, Part 1 (Sections 3.1 and 3.2) shows that embedding-based feature of CNN outperforms

hand-crafted features. After validating the advantage of the CNN-based model, we demonstrate that our TL approach (introduced in Section 2.2) is more effective in boosting the advantage farther than other TL approaches in Part 2 (Section 3.3). In Part 2, the same CNN-based model is used for all TL approaches.

General category	# of reviews with 5+ votes	Total # of reviews
Watches (Watch)	9,737	68,356
Cellphones (Phone)	18,542	78,930
Outdoor	72,796	510,991
Home	219,310	991,784
Electronics (Elec.)	354,301	1,241,778

Table 1: Amazon reviews from 5 different categories.

The lookup table $\mathbf{E}$ is initialized with pre-trained vectors from GloVe (Pennington et al., 2014) by setting $l = 100$. For CNNs, the activation function is ReLU, and the channel size is set to 128. We also set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.05$, and $\lambda_5 = 0.0008$. AdaGrad (Duchi et al., 2011) is used in training with an initial learning rate of 0.08. Fowllowing the previous work (Yang et al., 2015, 2016), ten-fold cross-validation is performed for all experiments and all the results are evaluated in correlation coefficients between the predicted helpfulness score and the ground truth score computed by "a of b approach" from the dataset.

3.1 Comparison with hand-crafted features

We first compare our base CNN model with regression baselines that use hand-crafted features which are STR, UGR, LIWC, INQUIRER (Yang et al., 2015), and aspect-based feature ASP (Yang et al., 2016), and the vanilla CNN (CNN) in (Kim, 2014). As shown in Table 2, both CNN-based models outperform the baselines, indicating CNN-based models have better expressiveness than these hand-crafted features for this task.

Our CNN-based model outperforms the vanilla CNN based one on relatively small domains (e.g., "Watches", "Cellphones") and achieves comparable results on large ones (e.g., "Electronics"). This is because the OOV problem is severe on small domains and our model with character-level representations can help more on them. In all, our CNN-based method shows better performance compared to the baselines.

	Watch	Phone	Outdoor	Home	Elec.
STR	0.276	0.349	0.277	0.222	0.338
UGR	0.425	0.466	0.412	0.309	0.355
LIWC	0.378	0.464	0.382	0.331	0.400
INQ	0.403	0.506	0.419	0.366	0.405
ASP	0.406	0.437	0.385	0.283	0.406
CNN	0.480	0.562	0.501	0.459	0.524
our CNN	**0.495**	**0.566**	**0.511**	**0.464**	**0.521**

Table 2: Comparison with linguistic features.

3.2 Comparison with ensemble features

We further compare our CNN-based model with two groups of ensemble features: Fusion_1 comprising of STR, UGR, LIWC, and INQUIRER features (Yang et al., 2015), and Fusion_2 further comprising of the ASP feature (Yang et al., 2016). As shown in Table 3.2, our CNN-based model consistently outperforms the models based on ensemble features.

	Watch	Phone	Outdoor	Home	Elec.
Fusion_1	0.488	0.539	0.497	0.432	0.484
Fusion_2	0.493	0.550	0.501	0.436	0.491
our CNN	**0.495**	**0.566**	**0.511**	**0.464**	**0.521**

Table 3: Comparison with ensemble features.

3.3 Comparison with TL models

To evaluate the effectiveness of our transfer learning approach, 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 use both source and target data with the adversarial training as in (Liu et al., 2017b). For TL based approaches, we use the "Electronics" category as the source domain and all other categories as target domains.

	Watch	Phone	Outdoor	Home
Src-only	0.471	0.459	0.447	0.365
Tgt-only	0.495	0.566	0.511	0.464
TL-S	0.501	0.564	0.511	0.468
Ours	**0.515**	**0.571**	0.510	**0.472**

Table 4: Comparison of TL models.

According to Table 4, due to the domain shift, Src-only performs worse than Tgt-only. This is intuitive as those domains are related but different. Our model achieves better or comparable results than Tgt-only and TL-S. This supports the benefits of transfer learning and demonstrates the usefulness of adding domain discriminators on both source and target domains.

Last but not the least, our model shows less improvement over Tgt-only when target domain data size increases. For example, our model yields an improvement of 4% over Tgt-only on the smallest domain "Watches." But the improvement drops to 1.7% on the largest domain "Home." To investigate this, we pick the category "Outdoor" as the target domain and track how our TL approach looses its edge as the amount (in terms of percentage in Table 5) of data from the target domain used in training increases. The full set of data from the source domain "Electronics" is constantly used.

	10%	30%	50%	70%	100%
Tgt-only	0.425	0.463	0.475	0.493	0.511
Ours	0.454	0.481	0.491	0.497	0.510
Improve	6.8%	3.7%	3.4%	0.6%	-0.2%

Table 5: Comparison of TL with respect to the amount of training data of the "Outdoor" category.

According to Table 5, the more data from the target domain, the less advantage our approach has over the Tgt-only model. It is more beneficial to leverage knowledge from another relevant domain when there is less data in the target domain. This also demonstrates that our model is able to learn transferable features from a relevant domain to help the task on a target domain which often has limited data.

4 Related Work

Review Helpfulness Prediction: The recent studies on review helpfulness prediction focus on hand-crafted features from the review texts. For example, (Yang et al., 2015) and (Martin and Pu, 2014) examined semantic features like LIWC, INQUIRER, and GALC. Subsequently, aspect- (Yang et al., 2016) and argument-based (Liu et al., 2017a) features are demonstrated to improve the prediction performance. However, these methods rely on sufficient labeled data and may not perform ideally for domains with limited data. To alleviate this issue, we employ Convolutional Neural Networks (CNNs) (Kim, 2014; Zhang et al., 2015) as the base model and further considers character-level representations (Ballesteros et al., 2015; Ling et al., 2015; Kim et al., 2016; Lee et al., 2017).

Transfer Learning: Transfer learning (TL) has been extensively studied in the last decade, interested readers can refer to (Pan and Yang, 2010) for a detailed survey. With the popularity of deep learning, a great amount of Neural Network (NN) based methods are proposed for TL (Yosinski et al., 2014; Wang and Zheng, 2015; Mou et al., 2016; Yang et al., 2017; Liu et al., 2017b). A simple but widely used framework is referred to as *fine-tuning* approaches, which first use the parameters of the well-trained models on the source domain to initialize the model parameters of the target domain, and then fine-tune the parameters based on labeled data in the target domain (Yosinski et al., 2014; Mou et al., 2016). Another typical framework is to use a shared NN to learn shared features for both source and target domains (Mou et al., 2016; Yang et al., 2017; Qiu et al., 2017). On top of that, specific shared framework use both a shared NN and domain-specific NNs to derive shared and domain-specific features (Ganin et al., 2016; Taigman et al., 2017; Yu et al., 2018). However it may not be ideal to separate shared and specific features, recent studies (Ganin et al., 2016; Taigman et al., 2017; Liu et al., 2017b) consider the adversarial networks to learn more robust shared features across domains. Inspired by this, our method adopts adversarial network on the shared features. In the meanwhile, we also use domain discriminators on both source and target features to help learn domain-specific features.

To the best of our knowledge, our work is the first to study cross-domain review helpfulness prediction. Without any hand-crafted features, our CNN-based method achieves better results than the existing approaches.

5 Conclusion

In this work, we proposed a convolutional neural network (CNN) based approach that combines both word- and character-level representations, for review helpfulness prediction. We studied transfer learning for the task and used auxiliary domain discriminators on both shared and specific representations. Experiments showed our CNN-based models outperform the existing approaches. In the near future, we will look at multi-task helpfulness prediction to further transfer knowledge across domains. Meanwhile, it is also worth studying domain correlation in the transfer learning (Yu et al., 2018) or multi-task settings (Qiu et al., 2017).

Acknowledgments

Bao's work in this paper is partially sponsored by National Science Foundation under Grant No. 1616216.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *EMNLP*. Association for Computational Linguistics, pages 349–359.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *NIPS*. MIT Press, pages 343–351.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *ACL*. Association for Computational Linguistics, pages 1193–1203.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12(7):2121–2159.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *JMLR* 17(59):1–35.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. Association for Computational Linguistics, pages 1746–1751.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *AAAI*. AAAI Press, pages 2741–2749.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics* 5:365–378.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586* .

Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. 2017a. Using argument-based features to predict and analyse review helpfulness. In *EMNLP*. Association for Computational Linguistics, pages 1358–1363.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017b. Adversarial multi-task learning for text classification. In *ACL*. Association for Computational Linguistics, pages 1–10.

Lionel Martin and Pearl Pu. 2014. Prediction of Helpful Reviews Using Emotions Extraction. In *AAAI*. AAAI Press, pages 1551–1557.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. ACM, pages 165–172.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *EMNLP*. Association for Computational Linguistics, pages 479–489.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. Association for Computational Linguistics, pages 1532–1543.

Minghui Qiu, Peilin Zhao, Ke Zhang, Jun Huang, Xing Shi, Xiaoguang Wang, and Wei Chu. 2017. A short-term rainfall prediction model using multi-task convolutional neural networks. In *ICDM*. IEEE, pages 395–404.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2017. Unsupervised cross-domain image generation. *ICLR* .

Dong Wang and Thomas Fang Zheng. 2015. Transfer learning for speech and language processing. In *APSIPA*. IEEE, pages 1225–1237.

Yinfei Yang, Cen Chen, and Forrest Sheng Bao. 2016. Aspect-based helpfulness prediction for online product reviews. In *ICTAI*. IEEE, pages 836–843.

Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. 2015. Semantic analysis and helpfulness prediction of text for online product reviews. In *ACL-IJCNLP*. Association for Computational Linguistics, pages 38–44.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *ICLR* .

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *NIPS 2014*. MIT Press, pages 3320–3328.

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 *WSDM*. ACM, pages 682–690.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*. MIT Press, pages 649–657.