Information Sciences 527 (2020) 241–252

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Cross-language question retrieval with multi-layer representation and layer-wise adversary

Bo Li*, Xiaodong Du, Meng Chen

*School of Computer Science, Central China Normal University, Wuhan, China*

ARTICLE INFO

*Article history:*
Received 22 October 2019
Revised 10 January 2020
Accepted 17 January 2020
Available online 21 March 2020

*Keywords:*
Cross-language question retrieval
Adversarial learning
Community question answering

ABSTRACT

In cross-language question retrieval (CLQR), users employ a new question in one language to search the community question answering (CQA) archives for similar questions in another language. In addition to the ranking problem in monolingual question retrieval, one needs to bridge the language gap in CLQR. The existing adversarial models for cross-language learning normally rely on a single adversarial component. Since natural languages consist of units of different abstract levels, we argue that crossing the language gap adaptatively on different levels with multiple adversarial components should lead to smoother text representation and better CLQR performance. To this end, we first encode questions into multi-layer representations of different abstract levels with a CNN based model which enhances conventional models with diverse kernel shapes and the corresponding pooling strategy so as to capture different aspects of a text segment. We then impose a set of adversarial components on different layers of question representation so as to decide the appropriate abstract levels and their role in performing cross-language mapping. Experimental results on two real-world datasets demonstrate that our model outperforms state-of-the-art models for CLQR, which is on par with the strong machine translation baselines and most monolingual baselines.



## 1. Introduction

In recent years, we have witnessed the great success of community question answering (CQA) services which range from the general Quora site[1] to the domain-specific StackOverflow site.[2] Users can ask questions and receive answers from other users in a crowd-source style in these CQA sites. Since a huge amount of questions and answers have accumulated in CQA sites, question retrieval (QR) has been proposed to fetch similar questions of a given question so that users can reuse the CQA content conveniently [1]. A lot of advances have been achieved on QR from the CQA research community. On top of traditional information retrieval (IR) models, researchers tried to overcome the lexical gap between questions in QR by introducing translation-based strategies [2] and topic model-based strategies [3]. More recently, deep models (e.g. [4–6]) were used in QR with superior performance. All the aforementioned studies deal with monolingual questions. One can note that there are substantially less CQA resources in languages other than English, especially within technical domains [7].

* Corresponding author.
*E-mail address:* libocs@qq.com (B. Li).

[1] https://www.quora.com.
[2] https://www.stackoverflow.com.

https://doi.org/10.1016/j.ins.2020.01.035
0020-0255/

Therefore, the non-English speakers (e.g. native Chinese speakers) can benefit a lot from cross-language question retrieval (CLQR) where the new question and candidate questions are written in different languages.

CLQR is a relatively novel topic in CQA research and we can find only a few studies on this task. In addition to the ranking problem in monolingual QR, one needs to cross the language gap in CLQR so as to match two questions in different languages, which can be practiced with either off-the-shelf machine translation systems or cross-language models specifically designed for CLQR. Joty et al. [8] investigated CLQR with a neural framework which used a simple feed-forward network to encode questions in two languages and an adversarial learning component to direct the learning of language invariant representation. Martino et al. [9] developed two models for CLQR, namely a cross-language tree kernel model and a simple neural model similar to that in [8]. They reported that kernel systems showed the better performance when training data were not sufficient to train a neural model. Rahman et al. [10] compared several query expansion strategies for CLQR that relied on resources such as word embeddings and DBpedia concept linking. Rücklé et al. [7] transformed CLQR into a monolingual QR problem via neural machine translation, which aimed to adapt the general machine translation model to technical domains so as to relieve translation errors.

Adversarial learning has been proven to be an effective and efficient technique for removing language specific characteristics, which helps to arrive at language invariance in a common semantic space. However, previous studies mostly make use of a single adversarial component (i.e. discriminator) to bridge the language gap (e.g. the work [8] on CLQR). Since natural languages are composed of units of different abstract levels (e.g. from words to phrases), we argue that crossing the language gap adaptatively according to intrinsic features on different abstract levels is superior, which can lead to smoother representation in the cross-language semantic space. Therefore, we follow recent advances on CLQR and propose a novel neural model with multi-layer representation and layer-wise adversarial learning. The model first encodes questions with a multi-layer feature extraction framework based on CNN in order to extract features of different abstract levels. Then layer-wise adversarial learning components are imposed on each feature extraction layer so as to bridge the language gap smoothly. Experimental results on real-world CQA datasets demonstrate that our model outperforms state-of-the-art models for CLQR and is comparable with the strong machine translation baselines and monolingual baselines.

The main contribution of this work lies in the neural model for CLQR based on layer-wise adversarial learning. The model encodes questions into multi-layer representations prior to imposing adaptative discriminators on each level. In the question encoding phase, we extend the conventional text encoding model with various kernel shapes and the corresponding pooling strategy so as to capture different aspects of a text segment. In the adversarial learning phase, we go beyond the typical usage of single discriminator in cross-language models and impose a set of discriminators on different representation levels so as to decide the role of each layer in cross-language mapping. Such a model leads to smoother representation of questions in the cross-language semantic space and improved CLQR performance. As far as we can tell, this is the first time that CLQR is modeled with multi-layer representation and layer-wise adversary.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the whole framework of our model and all the technical details. Section 4 describes the experiments designed to evaluate the model performance. Section 5 concludes this paper.

## 2. Related work

In this section, we will discuss the research work related to CLQR such as monolingual QR, cross-language question answering and adversarial learning. As we have discussed above, the studies specifically devoted to CLQR are still rare which have been introduced in Section 1. We will not discuss them again in this part.

### *2.1. Monolingual question retrieval*

QR has been attracting the continuous attention of the CQA research community, since it is an essential technique to make the CQA archive reusable to users. Early QR studies employed classic models in IR such as the language modeling approach. In order to solve the lexical gap problem of IR models applied in QR, researchers have developed the translation models and the topic models. The translation models (e.g. [2,11]) were inspired by the theory in statistical machine translation, which used the CQA question-answer pairs as bilingual parallel text to train the translation model. Then the model was able to capture the translation probability of aligned words. Such translation probability could be used to link literally different but semantically related words so as to cross the lexical gap. All the translation models showed improved performance compared with the original IR models. However, one can easily point out that question-answer pairs are far from being parallel in practice, which weakens the theoretical foundations of such models. The topic models for QR were directly the application of probabilistic topic models commonly used in text modeling. Similar with translation models, the topic models for QR also made strong assumptions for question-answer pairs. For instance, Ji et al. [12] assumed that questions and answers in a pair shared the same topic distribution. In addition to the topic distribution assumption, Zhang et al. [3] considered the answer quality signals and built a model performing robustly against noise in answers.

More recently, researchers resorted to deep models in QR which showed superior performance. For instance, Das et al. [4] employed CNN to encode CQA content (both questions and answers) and built a Siamese network framework with the contrastive loss. Wang et al. [5] made use of a standard CNN-based sentence encoding approach to take the general word

representation and the concept based representation as inputs. Uva et al. [6] combined artificial feature engineering with neural models in QR, which achieved improved performance compared with the purely neural models.

### 2.2. Cross-language question answering

Cross-language question answering (CLQA) that retrieves answers in a language different from that of questions is a highly related field of CLQR. CLQA has been practiced in several evaluations and benchmark datasets can be obtained from CLEF [13], NTCIR [14] and etc. Typical strategies transform CLQA into a monolingual problem via the translation of questions or answers with machine translation systems. For instance, Bouma et al. [15] employed Google Translate to translate questions prior to using their Joost system to perform monolingual QA. Similar strategies can be found in early studies such as [16,17]. Since general machine translation systems may bring in additional errors, there have been some studies trying to enhance the translation quality. For example, on one hand, Ture and Boschee [18] introduced a set of features that combined lexical and semantic similarities between a question-answer pair through multiple translation strategies. On the other hand, Ture and Boschee [18] noted that machine translation could be combined with other components of CLQA so as to perform joint training. By the way, some researchers built a unified semantic space among languages to deal with CLQA over knowledge bases like DBPedia [19]. In addition to the CLQA models, we note some other studies, for instance the ones on personalized cross-language search [20] and bilingual word embeddings with generative autoencoder [21], are relevant to both CLQA and CLQR tasks.

### 2.3. Adversarial learning

Deep neural networks (DNN) are first popular in computer vision (CV) before their broad usage in natural language processing and other fields. For instance, one can find that the recent advance in image recognition and retrieval [22] comes from the use of hierarchical deep word embedding model [23]. One can also refer to literature such as [24] and [25] for the advances of DNN in pose estimation and place recognition. Among various DNN techniques in CV, the generative adversarial networks (GAN) [26] and its variants such as WGAN [27] have been exploited broadly in topics such as image generation and image style transfer. GAN is able to learn arbitrarily complex data distribution via simultaneously training two components, namely a generative model G and a discriminative model D, in a *minmax* style.

We note that GAN is especially suitable for learning shared features among different domains or modals in CV [28], which is somehow related to our work in this paper. For instance, in domain adaptation, GAN has been used broadly which employs the generator to synthesize images or representations in different domains so as to learn the transferable features [29]. In terms of modals, GAN has been used successfully to deal with cross-modal retrieval which uses one modal (e.g. text) to retrieve another modal (e.g. image) [30]. Besides the aforementioned studies dealing with CV problems, adversarial learning has also been used successfully in natural language processing so as to learn the domain or language invariant representation. For instance, Liu et al. [31] combined adversarial learning into cross-domain text classification so as to enhance text representation by extracting domain common features and domain specific features. Joty et al. [8] applied adversarial learning on top of the simple feed-forward neural networks in order to learn language invariant representation and to cross the language gap in CLQR. Chen et al. [28] employed adversarial learning in monolingual text matching in order to learn the common features of a sentence pair, which was different from other work in that it modeled the discriminator and generator either in the collaborative way or in the adversarial way depending on the sentence pair similarity.

## 3. The layer-wise adversarial learning framework

### 3.1. General framework

The model in this paper relies on adversarial learning to learn language invariant representation so as to cross the language gap in CLQR. The general GAN consists of two main components, namely the generator G and the discriminator D, which play a *minmax* game [26]. When GAN is applied to representation learning in CLQR, for example in [8], the generator G performs the task of representation learning via mapping questions in different languages into a common subspace. G tries to confuse the discriminator D which is an adversarial component trying to figure out the language from which the representation is encoded. Different from the standard GAN, we impose multiple discriminators D in our model so as to cross the language gap smoothly via layer-wise adversarial learning. The idea is inspired by similar intuitions in CV [32] that different layers of image representation show different transferrability in domain adaptation. In our case, text representation on different abstract levels such as word, phrase and snippet has different characteristics. For instance, word and phrase representations are low-level features which are prone to be language specific. However, the upper-level features (e.g. a semantic block or a topic) are more abstract and tend to be language invariant on which we can cross the language gap more efficiently. We thus argue that language invariance in CLQR can be reached more efficiently and smoothly with multiple rather than only one discriminator.

Based on such considerations, we devise a cross-language learning framework consisting of two groups of components as illustrated in Fig. 1. The first one is the multi-layer text embedding networks (i.e. G) which encode questions (i.e. Questions 1 and 2) into multi-layer representation of different abstract levels (i.e. $\boldsymbol{f}^1, \boldsymbol{f}^2, \ldots, \boldsymbol{f}^m$). The second one is the layer-wise

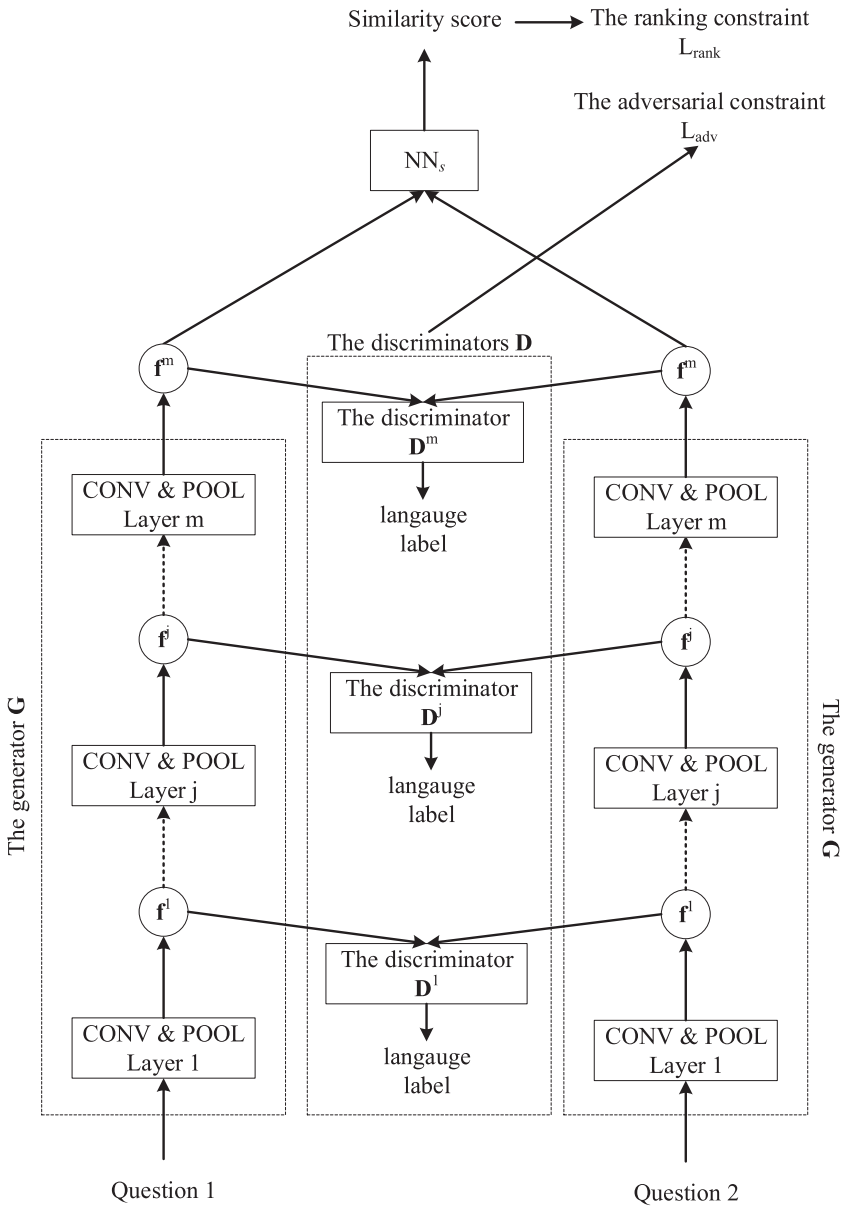


**Fig. 1.** The multi-layer representation and layer-wise adversarial learning framework.

adversarial components $D^1, D^2, \ldots, D^m$ on top of various levels of question representation to bridge the language gap adaptatively. Each discriminator $D^j$ works independently to predict the language label of a given feature vector $\boldsymbol{f}^j$ encoded from Question 1 or 2 by G. Finally, a neural network $NN_s$ is placed on top of the cross-language representation $\boldsymbol{f}^m$ of both questions to produce a similarity score for question ranking. The ranking constraint $L_{rank}$ (i.e. the generator's constraint) and the adversarial constraint $L_{adv}$ (i.e. the discriminator's constraint) are then defined based on the similarity score and the discriminators' output respectively. More details of the model will be given in the following sections.

### 3.2. Text encoding networks

As analyzed above, we first need to encode questions into multi-layer abstract levels. We note that CNN based text encoding approaches that treat word embedding sequence in a text segment as an *image* are an appropriate choice. There have been several studies in this field (e.g. [33,34]). We devise a structure similar to that in [35] for the sake of simplicity. The framework consists of several convolution and pooling layers (refer to Fig. 1) so as to extract multi-layer abstraction of questions.

*Input representation.* The input of text encoding network is the same as those in previous studies [33–35]. Let us assume that a text segment $s$ consisting of word sequence $w_1, w_2, \ldots, w_{l_s}$ where $l_s$ is the segment length. Each word $w$ is represented as the standard $d_w$-dimensional word embeddings. Then the segment $s$ can be denoted as a matrix $M_s$ of dimension $d_w \times l_s$ where each column corresponds to the vector representation of a word.

*Wide convolution layer.* The input matrix $M_s$ can be seen as an *image* on which the convolution and pooling operations can be performed, which resembles the standard practice in CV. We use here a wide convolution strategy similar to the one in [35] that combines the idea of *wide* in [34] and the idea of full-dimensional convolution in [33]. Wide convolution ensures the weights in the filter can reach words at the margins of the text segment. We denote a segment of $n$ word sequence starting from the index $i$ in $s$ as $s_{i,n} = (w_i, w_{i+1}, \ldots, w_{i+n-1})$. The representation of $s_{i,n}$ then corresponds to a subpart of $M_s$ that is denoted as $M_{s_{i,n}}$. Following the general idea of convolution, we can obtain the corresponding feature representation $\boldsymbol{c}_{i,n}$ of $s_{i,n}$ after convolution via:

$$\boldsymbol{c}_{i,n} = g(WM_{s_{i,n}} + \boldsymbol{b}) \tag{1}$$

where $g$ is a non-linear function, $W$ is a convolution weight matrix of size $d_f \times d_w n$ (i.e. $d_f$ kernels of the same shape) and $\boldsymbol{b}$ is a bias vector of length $d_f$. Since we are using the wide convolution, the index $i$ can be picked from $2-n$ to $l_s$. The resulting feature map matrix $C = (\boldsymbol{c}_{2-n,n}, \boldsymbol{c}_{3-n,n}, \ldots, \boldsymbol{c}_{l_s,n})$ then contains $d_f$ rows and $l_s + n - 1$ columns.

The Eq. (1) computes $d_f$ kernels of the same shape. Different from conventional models, we use here various kernels of different shapes so as to capture different aspects of a text segment, which resembles the idea in [36]. For instance, let us consider $d'_f$ kernels of a different size, then the corresponding feature representation for a segment $s_{i,n'} = (w_i, w_{i+1}, \ldots, w_{i+n'-1})$ can be written as $\boldsymbol{c}_{i,n'} = g(W'M_{s_{i,n'}} + \boldsymbol{b}')$ where the matrix $W'$ and the vector $\boldsymbol{b}'$ are convolution parameters. Similarly, we can obtain the feature map matrix $C'$ of $s_{i,n'}$ which has the size $d'_f \times (l_s + n' - 1)$. In this case, one can easily add more kernels of different shapes if necessary.

*Pooling layer.* We perform pooling on top of the convolution layer to extract robust features. We use here two kinds of pooling that is computed column-wise: one is the pooling applied on intermediate convolution layers for model derivation, the other is the pooling on the last convolution layer for text vector representation. On intermediate layers, since feature maps $C$ and $C'$ from different kernels are of different shapes, we need to use the appropriate pooling techniques (e.g. max pooling, average pooling) to extract robust features while guaranteeing that (a) the features after pooling have the same column number so that a number of convolution-pooling layers can be stacked to extract features of different abstract levels [35] and (b) features after pooling are compatible in terms of size from different feature maps (e.g. $C$ and $C'$) so that they can be combined into a comprehensive feature representation. For instance, let us assume we use the $k$-max pooling on the intermediate layers. After the pooling process, we concatenate the two feature maps into a unified feature map via:

$$F_{kmax} = POOL_{kmax}(C) \oplus POOL_{kmax}(C')$$

The representation $F_{kmax}$ then contains $d_f + d'_f$ rows meaning that we encode the input text with $d_f + d'_f$ feature extractors. Corresponding to the $k$-max pooling on intermediate layers, we perform a simple column-wise max pooling on the last convolution layer and obtain a single vector representation as the final representation of the input text. Furthermore, inspired by previous studies [35,37], we can also perform simple max pooling on intermediate layers to obtain a single vector representation which can be written as:

$$\boldsymbol{f}_{max} = POOL_{max}(C) \oplus POOL_{max}(C')$$

Note that the vector $\boldsymbol{f}_{\max}$ of length $d_f + d'_f$ will be used as the intermediate vector representation of the text on each abstract level, since it is cheaper to compute.

### 3.3. Layer-wise adversarial learning

Based on the analysis in above sections, text representation on different levels shows different properties when crossing the language gap. We prefer to applying adversarial learning on upper-level features to learn language invariant representation since these levels correspond to more abstract features. In order to realize this idea in a flexible way, we bring in the layer-wise adversarial learning strategy that is able to decide the layers on which adversarial learning should be imposed. We note that the idea resembles the collaborative and adversarial learning idea that has been used successfully in sentence similarity modeling [28] and in domain adaptation in CV [32]. However, our work differs significantly from previous studies in that (a) we consider the learning patterns on various abstract levels of text in different languages and (b) we model with the Wasserstein distance so as to guarantee optimization stability.

Before introducing the layer-wise adversarial learning framework, let us first recall the general idea of GAN [26]. GAN consists of a generative model G and a discriminative model D that are trained in a competing way. G produces samples based on a noisy distribution and aims to capture the real data distribution $P_r$. D tries to figure out whether a given data

point is generated by G satisfying a distribution $P_g$ or directly sampled from $P_r$. The objective according to which G and D play the *minmax* game can be written as:

$$\min_G \max_D V(D, G) = E_{x\sim P_r}[\log D(x)] + E_{\tilde{x}\sim P_g}[\log(1 - D(\tilde{x}))] \tag{2}$$

where maximizing $V(D, G)$ with respect to $D$ approximates the Jensen-Shannon divergence between $P_r$ and $P_g$.

According to the analysis in Section 3.1, when GAN is applied to cross-language learning, G and D compete in order to produce language invariant representations in Eq. (2). However, as we have analyzed above, on the lower levels of text representation, features are prone to be language specific and it is not efficient to cross the language gap on these layers. Based on such considerations, we change the competing learning manner in Eq. (2) for lower levels such that language specific rather than invariant features can be learned via these levels. To this end, we simply change the *minmax* game in Eq. (2) into a maximization problem that is:

$$\max_G \max_D V(D, G) = E_{x\sim P_r}[\log D(x)] + E_{\tilde{x}\sim P_g}[\log(1 - D(\tilde{x}))] \tag{3}$$

which corresponds to the expectation that the representation generated by G contains much language specific information and can be predicted correctly by the discriminator D. In this case, G and D are not trained in an adversarial way any more.

According to previous studies [27,38], it is unstable to train the standard GAN as in Eq. (2) since the Jensen-Shannon divergence that GAN aims to minimize is potentially not continuous. We thus follow the improvement in [27] and make use of the Wasserstein distance (i.e. Earth Mover's Distance) that is continuous under mild assumptions as the distance measure between two probability distributions. The Wasserstein-1 distance between two distributions $P_r$ and $P_g$ is formally defined in the infimum format as:

$$W(P_r, P_g) = \inf_{\gamma\in\Gamma(P_r,P_g)} E_{(x,y)\sim\gamma}\|x - y\| \tag{4}$$

where $\Gamma(P_r, P_g)$ is the set of joint distributions $\gamma(x, y)$ whose marginals satisfy $P_r$ and $P_g$. The original Wasserstein distance in Eq. (4) is hard to compute in practice. Based on appropriate approximation and the Kantorovich–Rubinstein theorem [39], the Wasserstein distance can be obtained through:

$$W(P_r, P_g) = \sup_{\|f\|_L\le 1} E_{x\sim P_r}[f(x)] - E_{\tilde{x}\sim P_g}[f(\tilde{x})]$$

Furthermore, the Wasserstein distance $W(P_r, P_g)$ can be obtained by solving the maximization problem which is:

$$\max_{D\in\mathbb{D}} E_{x\sim P_r}[D(x)] - E_{\tilde{x}\sim P_g}[D(\tilde{x})] \tag{5}$$

where D is the discriminator component (i.e. critic) in the set of 1-Lipschitz functions $\mathbb{D}$. Lastly, one can obtain the new objective to optimize in adversarial learning based on Eq. (5), which is[3]:

$$\min_G \max_{D\in\mathbb{D}} V(D, G) = E_{x\sim P_r}[D(x)] - E_{\tilde{x}\sim P_g}[D(\tilde{x})]$$

Since different layers of text representation show varied characteristics, two adversarial components $\min_G\max_D$ and $\max_G\max_D$ should be imposed on different feature extraction layers adaptively. To achieve the goal, we compare the two adversarial components and find out that the only difference exists in the G part. Indeed, one can rewrite the $\min_G\max_D$ operation in another format:

$$\begin{aligned}(G, D) &= arg\min_G \max_D V(D, G) \\ &= arg\max_G[-\max_D V(D, G)]\end{aligned} \tag{6}$$

In this case, we can see that Eqs. (6) and (3) are in the same format in the outer $\max_G$ operation, which can be combined into a uniform formula:

$$\max_G[\alpha \max_D V(D, G)] \tag{7}$$

where $\alpha \in \{+1, -1\}$ controls the optimization objectives. We can then combine two adversarial components conveniently into the text encoding network in order to build a layer-wise adversarial framework.

Let us then plug the adversarial components developed in Eq. (7) into the whole model in Fig. 1. To be precise, let us assume there are $m$ convolution-pooling layers in the text encoding network $NN_g$. Each layer consists of a set of CONV&POOL elements. The text vector representation after each layer $j(j = 1, 2, \ldots, m)$ is denoted as $NN_g^j(M_s)$ where $NN_g^j$ (i.e. a generator $G^j$) is the first $j$ convolution-pooling layers of the text encoding network and $M_s$ is the input text matrix. Then on top of the representation $NN_g^j(M_s)$ at each layer, we impose an adversarial component $NN_d^j$ (i.e. a discriminator $D^j$). Note that different from the conventional GAN model, we have a group of m G-D pairs (i.e. $NN_g^j - NN_d^j$) in our model. Based on above analysis,

[3] For more detailed explanations and derivations, one can refer to the original work [27].

the layer-wise adversarial learning objective on layer $j$ for the input $s$ can be written in a uniform formula which is:

$$\begin{aligned} L_j(s) &= \max_G[\alpha_j \max_D V(D,G)] \\ &= \max_{G^j}[\alpha_j \max_{D^j} V(NN_d^j, NN_g^j)] \end{aligned} \tag{8}$$

where $\alpha_j \in \mathbb{R}$ extends the weight $\alpha$ in Eq. (7) to allow for more flexibility. In our model, the parameter $\alpha_j$ controls the role of each layer in cross-language mapping. To be precise, when $\alpha_j$ is positive, the model aims to learn language invariant features on the corresponding layer $j$. Otherwise, the model tries to learn language specific features on the layer $j$. Furthermore, since we always expect to learn language invariant features on the last layer, the $\alpha_j$ for the last layer is fixed to the positive value 1. The overall adversarial loss $L_{adv}$ (in Fig. 1) can be obtained by directly summing $L_j$ for all layers which is $L_{adv}(s) = \sum_{j=1}^{m} L_j(s)$.

### 3.4. Model training

Neural models usually rely on a large amount of data for training. However, as one can find from monolingual QR literature (e.g. [4,6]), annotated question pairs are still rare in the research community which constraints the efficiency of neural models. We thus follow similar studies [3,4] and make use of question-answer pairs as the training data which are cheaper to obtain in large volume. We collect monolingual question-answer pairs from CQA sites and translate them into another language using machine translation so as to build the cross-language training set. More details of the dataset will be given in the experiment section. Based on such dataset, we employ the pairwise hinge loss to constraint the ranking activities of our model. Given a triple of question-answer pair denoted as $(q, a+, a-)$ where $a+$ is the positive (i.e. relevant) answer for $q$ and $a-$ is the negative (i.e. irrelevant) answer, the pairwise ranking loss $L_{rank}$ on $(q, a+, a-)$ can be written as:

$$L_{rank}(q, a+, a-) = max[0, 1 - score(q, a+) + score(q, a-)] \tag{9}$$

where the similarity *score* is computed with a feed-forward neural network $NN_s$ for a question and an answer in different languages (refer to Fig. 1).

Lastly, we combine the pairwise ranking loss $L_{rank}$ in Eq. (9) and the layer-wise adversarial learning loss $L_{adv}$ that is the sum of $L_j$ in Eq. (8) to obtain the final objective function for optimization, which is:

$$L(QR) = \sum_{(q,a+,a-)\in QR} L_{rank} + \beta \sum_{(q,a+,a-)\in QR} L_{adv} \tag{10}$$

where $\beta$ is the hyperparameter, $QR$ is the set of question-answer triples and the second element $\sum_{(q,a+,a-)\in QR} L_{adv}$ is computed by summing up the adversarial loss $L_{adv}$ on $(q, a+, a-)$. According to the conventional practice in adversarial learning, one can optimize the loss function $L$ by back-propagation with the mini-batch based gradient descent algorithm.

## 4. Experiments and results

### 4.1. Datasets

As we have discussed above, our model makes use of bilingual question-answer pairs in large volume for training, which do not exist in current studies. We thus build two datasets by fetching question-answer pairs from the popular Yahoo Answers website[4] and Baidu Zhidao website.[5] We obtain 12M English CQA items from Yahoo and 7M Chinese CQA items from Baidu. Considering the computational cost, we select a subset of 1.2M question-answer pairs from the categories such as *computers&Internet, Food&Drink* and *Health* in Yahoo as well as 1M question-answer pairs from the categories such as *Computers&Internet, Electronics&Digital* and *Health&Life* in Baidu. Lastly, we translate the question in each pair from English (resp. Chinese) into Chinese (resp. English) so as to construct the bilingual training data.

For evaluating CLQR performance, we randomly sample 500 English questions from the above Yahoo dataset and 500 Chinese questions from the above Baidu dataset. We make sure that these questions have been excluded from the above training set. For each of the 500 Yahoo (resp. Baidu) questions, we employ a retrieval model (e.g. BM25) to retrieve the whole Yahoo (resp. Baidu) question collection and retain the top 20 results. For each of the retrieved results, we follow previous studies (e.g. [40]) and ask two human annotators to label it as relevant or irrelevant. If a conflict happens, the third annotator is asked to make the final judgment. In order to construct the cross-language set, we ask translators to translate each of the 500 Yahoo (resp. Baidu) questions from English (resp. Chinese) to Chinese (resp. English). In addition to the evaluation set, for each of the Yahoo or Baidu dataset, we build another validation set consisting of 500 questions with the similar approach for building the test set. We again make sure that there is no intersection between the training set, validation set and the evaluation set.

[4] https://answers.yahoo.com.
[5] https://zhidao.baidu.com.

### 4.2. Experimental setup

The input to the text encoding network $NN_g$ is the 300D word embeddings trained with the skip-gram model [41] on the Wikipedia dump corpora.[7] The text encoding network consists of several convolution and pooling layers and we choose the stacking levels from {1, 2, 3, 4, 5} (i.e. $m$). The convolution filter widths are chosen from {2, 3, 4, 5} (i.e. $n$ and $n'$) and for each kernel type we use 100 kernels (i.e. $d_f$). The pooling strategy is selected from {max pooling, average pooling}. The hyperparameter $\beta$ in Eq. (10) is selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. The discriminator network $NN_d$ is implemented as a fully connected neural network with three layers of which the network structure is $input \rightarrow 64 \rightarrow output$. The batch size is fixed to 128. The learning rate is chosen from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. All the variable hyper parameters are tuned on the validation set. We implement the whole model with PyTorch.[8]

For performance estimation, we employ the measures frequently used in QR research which are mean average precision (MAP) and precision@10 (P@10). The significance test when cited is performed with the $t$-test at $p = 0.05$.

### 4.3. Results and analysis

In order to demonstrate the efficiency of our model, we design systematic experiments to compare our model with state-of-the-art models for CLQR. We will also inspect the model performance under different experimental settings.

#### 4.3.1. Comparisons with State-of-the-art Models

In this part, experiments are performed to compare our layer-wise adversarial learning framework with several other models in CLQR. We follow recent studies [8,9] and make use of the following baselines:

- Monolingual baselines. Similar to other fields such as cross-language information retrieval (CLIR), monolingual systems are strong baselines in CLQR. We choose as baselines several models ranging from the traditional unsupervised models to the recent neural models for monolingual QR. To be precise, we use BM25, translation-based model TRANS [11], topic model TOPIC [3] and the neural matching models DSSM [42], CNTN [40], AI-CNN [43], Match-SRNN [44] as the monolingual baselines. For a fair comparison, we train these monolingual baselines (except the unsupervised BM25 model) on the CQA items that are constructed in Section 4.1.
- Machine translation baselines. These baselines rely on machine translation systems to translate questions from other languages (e.g. Chinese) into the target language of answers prior to performing monolingual QR. We then choose to apply the best-performing monolingual QR models which are CNTN [40] and Match-SRNN [44].
- CLQR baselines. There have been only a few studies specialized in CLQR. We make use of the adversarial neural model CLANN in [8] and the cross-language tree kernel model CLTK in [9]. We train CLQR models on the same CQA items developed above.

The results of our model as well as all the baseline models are listed in Table 1. From the results one can observe that:

(a) Various monolingual baselines show significantly different performance on the CLQR task. Not supervising, the unsupervised BM25 model shows the worst performance on both datasets. The early models TRANS and TOPIC trying to solve the lexical gap problem of traditional IR models outperform the BM25 model with significance. The four deep matching models show different performance on the two datasets where DSSM is the weakest model and Match-SRNN is the best one. Indeed, all deep models except DSSM are always significantly better than any of the other monolingual baselines. Such results are coincident with previous studies (e.g. [5]), which demonstrate the efficiency of deep models on QR given enough training data.

(b) Both machine translation baselines show slightly worse but similar performance as their corresponding monolingual models. The performance degradation that is significant only appears on CNTN-MT with P@10 on Yahoo dataset. The comparisons demonstrate the efficiency of modern machine translation systems. However, as indicated by [7], using machine translation system in CLQR is not suitable for many practical reasons, which is similar with the case in CLIR [45].

(c) Two CLQR baselines show better performance than that of the early monolingual models (i.e. BM25, TRANS and TOPIC). They are on par with or slightly better than the deep model DSSM in most cases. However, both CLQR baselines are significantly worse than the other monolingual baselines in most cases, which indicates that crossing the language gap with both CLQR baselines removes certain information useful for question ranking. We further note that both CLQR baselines are worse than the machine translation baselines although CLANN approaches the performance of CNTN-MT in most cases. Such a conclusion is also partially coincident with the findings in CLIR literature. Lastly, we note that CLANN outperforms CLTK which is different from the conclusion in [9]. The reason is probably that we use a larger corpus to train the neural model.

[7] https://dumps.wikimedia.org.
[8] https://pytorch.org.

**Table 1**
Experimental results of our model (the last row) with regard to various baselines on Yahoo dataset and Baidu dataset. The significance comparisons are given in the text.

| | Yahoo dataset | | Baidu dataset | |
|---|---|---|---|---|
| Model | MAP | P@10 | MAP | P@10 |
| Monolingual baselines | | | | |
| BM25 | 0.385 | 0.302 | 0.341 | 0.285 |
| TRANS | 0.472 | 0.349 | 0.410 | 0.324 |
| TOPIC | 0.478 | 0.350 | 0.413 | 0.326 |
| DSSM | 0.496 | 0.355 | 0.424 | 0.329 |
| AI-CNN | 0.530 | 0.369 | 0.445 | 0.338 |
| CNTN | 0.534 | 0.373 | 0.452 | 0.343 |
| Match-SRNN | 0.551 | 0.379 | 0.473 | 0.356 |
| Machine translation baselines | | | | |
| CNTN-MT | 0.526 | 0.364 | 0.449 | 0.337 |
| Match-SRNN-MT | 0.543 | 0.374 | 0.466 | 0.353 |
| CLQR baselines | | | | |
| CLANN | 0.521 | 0.360 | 0.439 | 0.335 |
| CLTK | 0.505 | 0.354 | 0.421 | 0.328 |
| **Our model** | ***0.539*** | ***0.372*** | ***0.460*** | ***0.349*** |

**Table 2**
Our model with different text encoding components. The significance comparisons are given in the text.

| | Yahoo dataset | | Baidu dataset | |
|---|---|---|---|---|
| Text encoding | MAP | P@10 | MAP | P@10 |
| ENC-K | 0.527 | 0.364 | 0.450 | 0.342 |
| ENC-P | 0.536 | 0.372 | 0.465 | 0.353 |
| **Our model** | ***0.539*** | ***0.372*** | ***0.460*** | ***0.349*** |

(d) Our model, by using a layer-wise adversarial learning framework, significantly outperforms both CLQR baselines proposed very recently. Furthermore, we find that our model is the only CLQR model that is on par with both machine translation baselines. Indeed, our model is better than CNTN-MT in all cases, which is slightly worse than Match-SRNN-MT with all differences being not significant. With respect to the monolingual baselines, our model is always significantly better than BM25, TRANS, TOPIC and DSSM. Our model is also on par with CNTN and AI-CNN. The strong baseline Match-SRNN significantly outperforms our model in most cases, which might indicate the importance of interaction signals in question matching.

To sum up, our model significantly outperforms the state-of-the-art CLQR models proposed very recently, which is on par with the strong machine translation baselines and most monolingual baselines.

*4.3.2. Analysis of text encoding component*

In the text encoding component in Section 3.2, we have used convolution kernels of various shapes and the corresponding pooling mechanism so as to capture different aspects of text while ensuring computational efficiency. Such a design is different from conventional work (e.g. [35]) using uniform kernel shape. In order to prove the effectiveness of our design, we perform CLQR experiments with different text encoding components, which are:

- ENC-K: the standard encoding component used in [35] that uses uniform kernel shape rather than diverse kernel shapes as in our model.
- ENC-P: the encoding component with the feature map from the original intermediate representation (e.g. from $k$-max pooling) rather than the simplified representation (e.g. from simple max pooling) as used in our model.

We replace our text encoding network with the above two variants and redo the CLQR experiments with other experimental settings unchanged. The results are then listed in Table 2. From the results one can observe that (a) using kernels of diversity can lead to better text representation and better retrieval performance on both datasets (comparing our model with ENC-K). The improvement that is significant appears on Baidu dataset with MAP and on Yahoo dataset with MAP. (b) using the more complex pooling strategy as in ENC-P rather than the simple one in our model leads to longer feature representation. However, the performance of ENC-P is never significantly better than our model, which indicates that using the simple pooling strategy maintains CLQR performance while reducing the computational cost.

**Table 3**
Comparisons of our model with regard to four variants with different adversarial components. The significance comparisons are given in the text.

| | Yahoo dataset | | Baidu dataset | |
|---|---|---|---|---|
| Model Variants | MAP | P10 | MAP | P10 |
| ADV-NON | 0.501 | 0.353 | 0.420 | 0.328 |
| ADV-ONE | 0.524 | 0.363 | 0.445 | 0.337 |
| ADV-ALL | 0.525 | 0.361 | 0.427 | 0.334 |
| ADV-TOP | 0.487 | 0.341 | 0.408 | 0.320 |
| **Our model** | ***0.539*** | ***0.372*** | ***0.460*** | ***0.349*** |

*4.3.3. Analysis of adversarial learning component*

The layer-wise adversarial learning components are the crucial part of our model which help to cross the language gap smoothly. According to the design in Section 3.3, we expect the model to learn language invariant features on upper levels through Wasserstein distance directed adversarial learning. In order to validate the role of layer-wise adversarial learning, we try to modify the adversarial components so as to validate their importance in the whole model. To be precise, we develop several variants of the current model which are:

- ADV-NON: the model without adversarial learning component. We remove the adversarial loss $L_{adv}$ from the total loss $L$ to optimize in Eq. (10). This is a strategy commonly used in question answering literature before adversarial learning is introduced.
- ADV-ONE: the model with only one adversarial component on the top-level text representation. That is to say we set $L_{adv} = L_m$ in Eq. (10) where $m$ is the number of convolution-pooling layers in the text encoding network (refer to Fig. 1). It is the typical way of using adversarial learning in cross-language representation learning.
- ADV-ALL: the model with fixed adversarial components on all the layers. That is to say we remove the *layer-wise* property from the original model by setting the parameters $\alpha_j (j = 1, 2, \ldots, m)$ in Eq. (8) to be positive. In this case, we explicitly aim to learn language invariant features on all the abstract levels.
- ADV-TOP: in the original model, we force the top-level representation to be language invariant by setting $\alpha_m$ to be positive in Eq. (8). We change such a constraint in ADV-TOP by setting $\alpha_m$ to be negative, which means that the top-level representation is forced to be language variant.

We compare our model with the above variants under the same experimental settings and list the results in Table 3. From the results one can conclude that our layer-wise adversarial model outperforms all the four variants with most differences being significant. It shows that layer-wise adversarial learning is important if one wants to cross the language gap effectively and efficiently for CLQR. We further compare these variants and find out that (a) trying to learn language invariant features on all layers as in ADV-ALL violates the layer-wise intuition in this paper and harms the performance (comparing ADV-ALL with our model). (b) following the convention in adversarial learning, it is beneficial to impose the adversarial component on the last layer of text representation to force the learned representation to be language invariant (comparing ADV-ONE with ADV-NON). However, a single adversarial component is not efficient enough for one to achieve language invariant (comparing ADV-ONE with our model). (c) it is important to explicitly ask the last layer to learn language invariant representation as in our layer-wise model. Otherwise, the model could not cross the language gap successfully (comparing ADV-TOP with our model). Indeed, if the top layer is forced to be language variant as in ADV-TOP, the layer-wise model can not decide how to bridge the language gap, resulting in the worst performance in all the variants.

## 5. Conclusions

We propose in this paper a neural model for CLQR with multi-layer representation and layer-wise adversary. The model consists of a multi-layer text encoding network and several adversarial learning components on all the layers. The text encoding network is developed based on CNN and contains stacking of convolution and pooling layers so as to extract text features of different abstract levels. The adversarial learning components are imposed on each layer with the aim of learning language invariant features so as to cross the language gap smoothly. Lastly, the layer-wise adversarial constraints and the pairwise ranking loss are combined as the objective to optimize. Experimental results on two real-world datasets demonstrate that our model outperforms state-of-the-art models for CLQR, which is also on par with the machine translation baselines and most monolingual baselines. Furthermore, we prove by experiments the importance of kernel shape diversity and layer-wise adversarial components in our model.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Bo Li:** Conceptualization, Methodology, Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Xiaodong Du:** Methodology, Software, Writing - original draft, Writing - review & editing. **Meng Chen:** Software, Validation, Writing - original draft, Writing - review & editing.

## Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work was co-supported by Humanity and Social Science Youth Foundation of Ministry of Education of China (no. 19YJC870012), Guangdong Province Key Laboratory of Cyber-Physical System, National Natural Science Foundation of China (no. 61977032), Research Planning Project of National Language Committee (no. YB135-40), as well as Fundamental Research Funds for the Central Universities (nos. CCNU19ZN010, CCNU19TS019).

## References

[1] I. Srba, M. Bielikova, A comprehensive survey and classification of approaches for community question answering, ACM Trans. Web 10 (3) (2016) 18:1–18:63.
[2] J.-T. Lee, S.-B. Kim, Y.-I. Song, H.-C. Rim, Bridging lexical gaps between queries and questions on large online Q&A collections with compact translation models, in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, in: EMNLP, 2008, pp. 410–418.
[3] K. Zhang, W. Wu, H. Wu, Z. Li, M. Zhou, Question retrieval with high quality answers in community question answering, in: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, in: CIKM, 2014, pp. 371–380.
[4] A. Das, H. Yenala, M. Chinnakotla, M. Shrivastava, Together we stand: Siamese networks for similar question retrieval, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, in: ACL, 2016, pp. 378–387.
[5] P. Wang, Y. Zhang, L. Ji, J. Yan, L. Jin, Concept embedded convolutional semantic model for question retrieval, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, in: WSDM, 2017, pp. 395–403.
[6] A. Uva, D. Bonadiman, A. Moschitti, Injecting relational structural representation in neural networks for question similarity, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, in: ACL, 2018, pp. 285–291.
[7] A. Rücklé, K. Swarnkar, I. Gurevych, Improved cross-lingual question retrieval for community question answering, in: Proceedings of The World Wide Web Conference, in: WWW, 2019, pp. 3179–3186.
[8] S. Joty, P. Nakov, L. Marquez, I. Jaradat, Cross-language learning with adversarial neural networks: Application to community question answering, in: Proceedings of the 21st Conference on Computational Natural Language Learning, in: CoNLL, 2017, pp. 226–237.
[9] G.D.S. Martino, S. Romeo, A. Barroón-Cedeño, S. Joty, L. Maàrquez, A. Moschitti, P. Nakov, Cross-language question re-ranking, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR, 2017, pp. 1145–1148.
[10] M.M. Rahman, S. Hisamoto, K. Duh, Query expansion for cross-language question re-ranking, CoRR abs/1904.07982 (2019).
[11] X. Xue, J. Jeon, W.B. Croft, Retrieval models for question and answer archives, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR, 2008, pp. 475–482.
[12] Z. Ji, F. Xu, B. Wang, B. He, Question-answer topic model for question retrieval in community question answering, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, in: CIKM, 2012, pp. 2471–2474.
[13] P. Forner, A. Peñas, E. Agirre, I. Alegria, C. Forăscu, N. Moreau, P. Osenova, P. Prokopidis, P. Rocha, B. Sacaleanu, R. Sutcliffe, E. Tjong Kim Sang, Overview of the CLEF 2008 multilingual question answering track, in: Proceedings of Workshop of the Cross-Language Evaluation Forum for European Languages, 2009, pp. 262–295.
[14] Y. Sasaki, C.-J. Lin, K. hua Chen, H.-H. Chen, Overview of the NTCIR-6 cross-lingual question answering task, in: Proceedings of the 6th NTCIR Workshop on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, 2007.
[15] G. Bouma, J. Mur, G. van Noord, L. van der Plas, J. Tiedemann, Question answering with joost at CLEF 2008, in: Proceedings of Workshop of the Cross-Language Evaluation Forum for European Language, 2008.
[16] S. Hartrumpf, I. Glockner, J. Leveling, Efficient question answering with question decomposition and multiple answer streams, in: Proceedings of Workshop of the Cross-Language Evaluation Forum for European Language, 2009.
[17] H. Shima, T. Mitamura, Bootstrap pattern learning for open-domain CLQA, in: Proceedings of NTCIR-8 Workshop, 2010.
[18] F. Ture, E. Boschee, Learning to translate for multilingual question answering, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, in: EMNLP, 2016, pp. 573–584.
[19] A. Pouran Ben Veyseh, Cross-lingual question answering using common semantic space, in: Proceedings of TextGraphs-10: the Workshop on Graph-based Methods for Natural Language Processing, 2016, pp. 15–19.
[20] D. Zhou, W. Zhao, X. Wu, S. Lawless, J. Liu, An iterative method for personalized results adaptation in cross-language search, Inf. Sci. (Ny) 430–431 (2018) 200–215.
[21] J. Su, S. Wu, B. Zhang, C. Wu, Y. Qin, D. Xiong, A neural generative autoencoder for bilingual word embeddings, Inf. Sci. (Ny) 424 (2018) 287–300.
[22] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2015) 767–779.
[23] J. Yu, M. Tan, H. Zhang, D. Tao, Y. Rui, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[24] C. Hong, J. Yu, J. Zhang, X. Jin, K. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE Trans. Ind. Inf. 15 (7) (2019) 3952–3961.
[25] J. Yu, C. Zhu, J. Zhang, Q. Huang, D. Tao, Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–14.
[26] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, in: NIPS, 2014, pp. 2672–2680.
[27] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning, in: ICML, 2017, pp. 214–223.
[28] Q. Chen, Q. Hu, J.X. Huang, L. He, Can: Enhancing sentence similarity modeling with collaborative and adversarial network, in: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR, 2018, pp. 815–824.
[29] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, D. Erhan, Domain separation networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, in: NIPS, 2016, pp. 343–351.
[30] B. Wang, Y. Yang, X. Xu, A. Hanjalic, H.T. Shen, Adversarial cross-modal retrieval, in: Proceedings of the 25th ACM international conference on Multimedia, in: MM, 2017, pp. 154–162.
[31] P. Liu, X. Qiu, X. Huang, Adversarial multi-task learning for text classification, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, in: ACL, 2017, pp. 1–10.

[32] W. Zhang, W. Ouyang, W. Li, D. Xu, Collaborative and adversarial network for unsupervised domain adaptation, in: Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition, in: CVPR, 2018.
[33] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, in: EMNLP, 2014, pp. 1746–1751.
[34] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, in: ACL, 2014, pp. 655–665.
[35] W. Yin, H. Schütze, B. Xiang, B. Zhou, ABCNN: Attention-based convolutional neural network for modeling sentence pairs, Trans. Assoc. Comput.Linguist. 4 (2016) 259–272.
[36] Y. Zhang, B.C. Wallace, A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, in: Proceedings of the 8th International Joint Conference on Natural Language Processing, in: IJCNLP, 2017, pp. 253–263.
[37] W. Yin, S. Ebert, H. Schütze, Attention-based convolutional neural network for machine comprehension, in: Proceedings of the 2016 NAACL Workshop on Human-Computer Question Answering, 2016, pp. 15–21.
[38] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, in: NIPS, 2017, pp. 5767–5777.
[39] C. Villani, Optimal Transport: Old and New, Springer-Verlag, 2008.
[40] X. Qiu, X. Huang, Convolutional neural tensor network architecture for community-based question answering, in: Proceedings of the 24th International Conference on Artificial Intelligence, in: IJCAI, 2015, pp. 1305–1311.
[41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, in: NIPS, 2013, pp. 3111–3119.
[42] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, Learning deep structured semantic models for web search using clickthrough data, in: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, in: CIKM, 2013, pp. 2333–2338.
[43] X. Zhang, S. Li, L. Sha, H. Wang, Attentive interactive neural networks for answer selection in community question answering, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, in: AAAI, 2017.
[44] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, X. Cheng, Match-SRNN: modeling the recursive matching structure with spatial rnn, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, in: IJCAI, 2016, pp. 2922–2928.
[45] D. Zhou, M. Truran, T. Brailsford, V. Wade, H. Ashman, Translation techniques in cross-language information retrieval, ACM Comput. Surv. 45 (1) (2012) 1:1–1:44.