Knowledge and Cross-Pair Pattern Guided Semantic Matching for Question Answering

Zihan Xu,[1,2] **Hai-Tao Zheng,**[1,2]* **Shaopeng Zhai,**[3] **Dong Wang**[1,2]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University
{xu-zh17, wangd18}@mails.tsinghua.edu.cn, zheng.haitao@sz.tsinghua.edu.cn,
zsp1197@163.com

Abstract

Semantic matching is a basic problem in natural language processing, but it is far from solved because of the differences between the pairs for matching. In question answering (QA), answer selection (AS) is a popular semantic matching task, usually reformulated as a paraphrase identification (PI) problem. However, QA is different from PI because the question and the answer are not synonymous sentences and not strictly comparable. In this work, a novel knowledge and cross-pair pattern guided semantic matching system (KCG) is proposed, which considers both knowledge and pattern conditions for QA. We apply explicit cross-pair matching based on Graph Convolutional Network (GCN) to help KCG recognize general domain-independent Q-to-A patterns better. And with the incorporation of domain-specific information from knowledge bases (KB), KCG is able to capture and explore various relations within Q-A pairs. Experiments show that KCG is robust against the diversity of Q-A pairs and outperforms the state-of-the-art systems on different answer selection tasks.

Introduction

Semantic matching is a basic problem in natural language processing. Many tasks are essentially a semantic matching problem, such as information retrieval (IR), question answering (QA) and paraphrase identification (PI) (Li, Xu, and others 2014). In QA, the matching of the question with the most proper answer from a set of candidates, which is known as answer selection (AS), remains challenging due to the diversity of Q-A pairs.

Usually, AS is reformulated as a PI problem. Methods can be divided into three categories based on the general model structures: Siamese networks (Feng et al. 2015; Yang, Yih, and Meek 2015), attentive networks (Santos et al. 2016; Yin et al. 2016) and Compare-Aggregate networks (Wang and Jiang 2017; Bian et al. 2017). However, these methods are all based on the comparing framework. It appears that by optimizing the likelihood of two text sequences being a matched pair based on their similarity, neural models assign high probability to those with the same words, phrases or other patterns. For example, considering Q_1 in Figure 1, it is difficult to pick out the true answer based on Q-A similarity, since the comparing framework is likely to be misled by words with identical attributes (marked by color).

In fact, QA is different from PI because the question and the answer are not synonymous sentences and not strictly comparable. Instead of being semantically equivalent, Q-A pairs usually form a continuity in meaning (and generally fall into certain Q-to-A patterns). Therefore, the basic semantic matching strategy for answer selection has room for improvement. In this work, we study the answer selection problem based on fundamental characteristics of QA itself. We observe that, a sentence will be considered as a proper answer only if it meets two basic conditions. First, the information in the sentence must be relevant to that in the question (knowledge condition). Second, the structure of the sentence should correspond to the question structure (pattern condition). Knowledge condition ensures that the sentence is "telling the truth", while pattern condition checks whether the sentence is "responding to" the question.

There is growing interest in the study of the knowledge condition. Some recent work leverages Wikipedia (Chen et al. 2017), knowledge bases (Shen et al. 2018) or other external resources to provide background information for QA. However, content correlation with the question is not sufficient for an answer. As can be seen from Figure 1, the listed candidate answers to Q_1 are more reliable than others because they are all statements of fact, and are all about the entity "8 track" in the question. However, among these candidates, A_{1-1} and A_{1-2} are giving irrelevant answers, but are also likely to be assigned with high probability, unless extra semantic parsing or information extraction methods (entity linking and relation detection) are conducted as in KBQA (Yao and Van Durme 2014).

However, if the pattern condition is followed when designing AS models, irrelevant answers will be avoided with few auxiliary tasks. As in the example, Q_1 and Q_2 both contain **what year was ... invented**. Meanwhile, A_{1-3} shares the same pattern **... was created in (year) ...** with A_{2-1}. These pairs are in a common Q-to-A pattern. If the pattern is explicitly learned to guide the answer selection process of Q_1, the correct answer will be picked out more easily.

*Corresponding Author

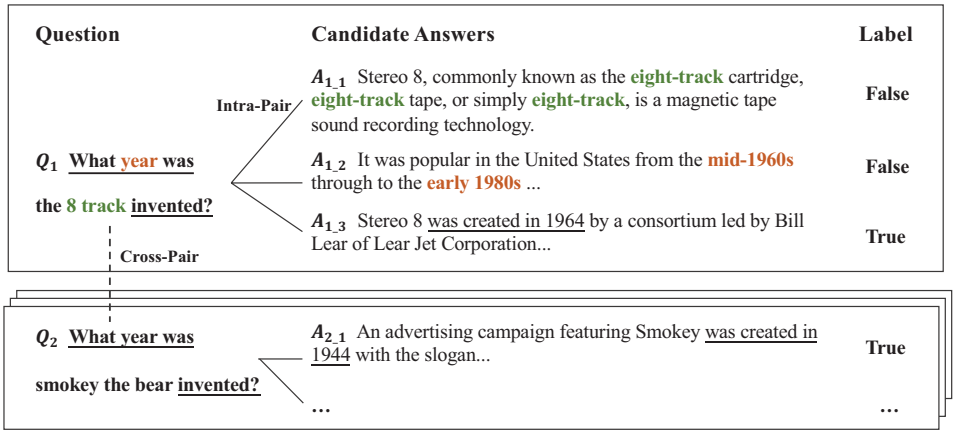

Figure 1: QA pairs from the WikiQA corpus. Intra-pair and cross-pair matches are denoted by colors and underlines respectively.

Taking both the pattern and knowledge conditions into consideration, we propose a novel knowledge and cross-pair pattern guided system (KCG) for answer selection. First, the idea of cross-pair similarity (Zanzotto and Moschitti 2006) is applied to learning general Q-to-A patterns. To be specific, for each candidate answer A of the question Q, in addition to the common intra-pair comparison between Q and A, matching is conducted between this pair $P = (Q, A)$ to other Q-A pairs. Therefore, global information is incorporated into each single matching pair.

To explicitly model cross-pair dependencies, we regard Q-A pairs as nodes and build a graph around the idea that similar Q-A pairs are close to each other, and turn AS into a node classification problem based on Graph Convolutional Network (GCN). GCN has been demonstrated as one of the most effective approaches for semi-supervised learning (Kipf and Welling 2017) because of its ability to exploit connectivity patterns between labeled and unlabeled data. Therefore, we find it a good fit for capturing global correlations and learning cross-pair patterns. With the correlation matrix which guides information propagation among nodes, the classification process retains semantic structures in the embedding space, where related concepts are neighbors.

In order to meet the knowledge condition, multi-view attention is utilized to capture interactive features within the Q-A pair (intra-pair matching part). To be specific, we adopt both textual attention from words and knowledge-based attention from entities to enhance the representation learning of the Q-A pair with the Compare-Aggregate network. Therefore, the model implements comparison on word, sentence and knowledge levels, thus learning more comprehensive intra-pair information.

Our main contributions include:

- We propose a universal semantic matching strategy for question answering. Different from the traditional comparing framework, we apply both intra-pair and cross-pair matching, thus enabling our system to learn not only multi-view information between the question and the answer, but also global Q-to-A pattern information.
- In order to learn cross-pair patterns, we propose the Q-A pair graph, and conduct node classification with GCN to capture global correlations. To the best of our knowledge, this is the first study to model the QA corpus as a graph to perform a GCN-based post-procedure, which may expand the application of graph neural networks on textual data.
- The proposed system considers both knowledge and pattern conditions for QA, and outperforms the state-of-the-art results on different answer selection tasks.

Related Work

Deep Semantic Matching Semantic matching is usually solved with the score of semantic recall (similarity computation) based on the comparing framework. Deep semantic matching starts with Siamese networks (Feng et al. 2015; Yang, Yih, and Meek 2015). These models use the same structure to encode the semantic sequences separately for matching. Then more interaction between sequences has been introduced by soft-attention (Santos et al. 2016; Yin et al. 2016). Further, some interaction-based networks (Hu et al. 2014; Pang et al. 2016; Wang and Jiang 2017) are proposed, most of which conduct the matching process before further representation learning.

However, these methods are all based on the intra-pair comparing framework. Cross-pair similarity was proposed by (Zanzotto and Moschitti 2006) in the textual entailment task. They devised the tree kernel based on cross-pair similarity for Support Vector Machines (SVM). Recently, (Tymoshenko and Moschitti 2018) combine the tree kernels with word-based kernels for AS. In this paper, we introduce GCN to model cross-pair dependencies, since GCN is naturally good at exploiting connectivity patterns through incorporating neighborhood information.

Application of GCN on NLP GCN is a simplified graph neural network (GNN), first introduced by (Kipf and Welling 2017) to perform semi-supervised classification. In NLP, GCN is mainly explored in tasks such as semantic role labeling (Marcheggiani and Titov 2017), machine translation (Bastings et al. 2017) and relation classification (Li, Jin, and Luo 2018) to encode syntactic structures. (Lai et al.

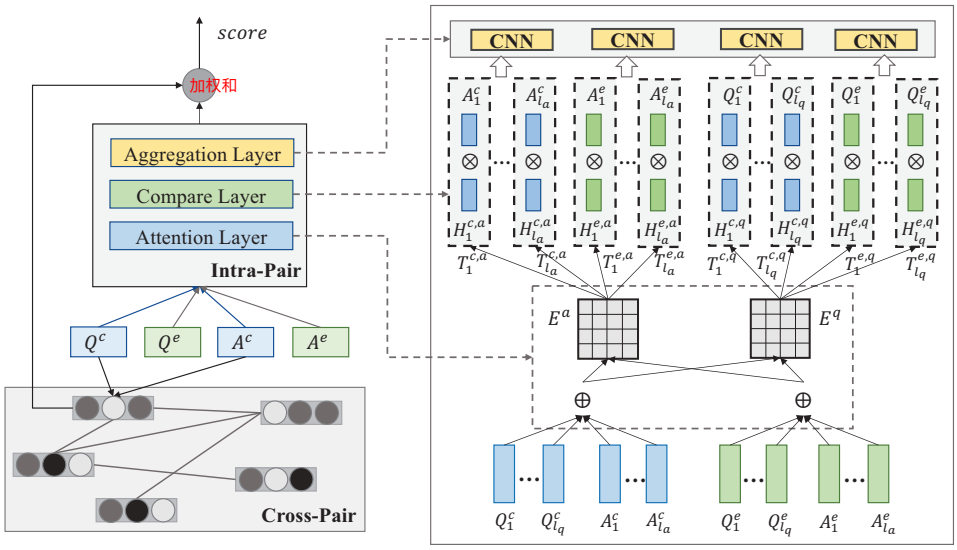

Figure 2: The proposed knowledge and cross-pair pattern guided semantic matching system (KCG).

2019) introduce word lattice (a directed graph) into Chinese QA, but the basic model relies on Siamese intra-pair matching. Besides, the above applications usually require that the data itself exhibit a natural graph structure, and mainly build the graphs inside sentences.

(Yao, Mao, and Luo 2019) first model a whole corpus as a graph where documents and words are regarded as nodes. However, the graph is built on traditional features like word co-occurrence, which may ignore word orders useful for text classification. Our graph based on sentence pair representations and correlations is easy to build and effective to model cross-pair dependencies. With the design and high-quality node embeddings, the application of GCN on textual data without pre-defined graph structures can be extended.

Knowledge and Cross-Pair Pattern Guided Semantic Matching

In this part, we elaborate on KCG for semantic matching in question answering, as shown in Figure 2. The cross-pair and intra-pair parts are trained independently, and final decisions are simply made based on weighted sum of the predictions to reduce dependency on parameters.

Cross-Pair Learning

Graph Convolutional Networks The essential idea of GCN is to update node representations by propagating information among nodes. Formally, for a graph $G = (V, E)$, $V(|V| = n)$ and E are sets of nodes and edges respectively. Every node is assumed to be connected to itself, i.e., $(v, v) \in E$ for any v. $X \in \mathbb{R}^{n \times m}$ is the feature matrix containing the features of all n nodes, where m is the dimension of feature vectors. A is the adjacency matrix of G and D is the degree matrix, where $D_{ii} = \sum_j A_{ij}$. The layerwise propagation rule is defined as:

$$Z^{(j+1)} = \rho(\tilde{A} Z^{(j)} W^{(j)}), \tag{1}$$

where j denotes the layer number and $Z^{(0)} = X$. $\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix and $W^{(j)}$ is a trainable weight matrix. ρ is an activation function, e.g., ReLU. Higher order neighborhood information can be introduced by stacking multiple GCN layers.

GCN-Based Cross-Pair Learning For graph-based learning, the key challenge is to exploit graph structures and data features to improve learning performance. In fields where GCN is widely used (e.g., social network, citation network or knowledge graph), the data usually exhibits a natural graph structure. However, there is no pre-defined graph structure in textual QA. Thus, the building of the graph is a crucial problem. In order to model cross-pair dependencies, we build the graph over Q-A pairs (Figure 3), where each node is represented by the sentence pair embedding. The correlation matrix P is computed based on cosine similarity between embedding vectors. We binarize the matrix by a threshold τ and get:

$$A_{ij} = \begin{cases} 0, & if P_{ij} < \tau \\ 1, & if P_{ij} \geq \tau \end{cases}, \tag{2}$$

where A is the binary correlation matrix.

The built graph is fed into GCN, and the output of the penultimate layer is passed to a softmax layer:

$$Z = \text{softmax}(\tilde{A} X W), \tag{3}$$

The loss function is defined as the cross-entropy error over all labeled Q-A pairs:

$$\mathcal{L} = -\sum_{p \in \mathcal{Y}_P} \sum_{f=1}^{F} Y_{pf} ln Z_{pf}, \tag{4}$$

where $\mathcal{Y}_P$ is the indices of labeled Q-A pairs, Y is the label indicator matrix, and F is the dimension of the output, which is equal to the number of classes.

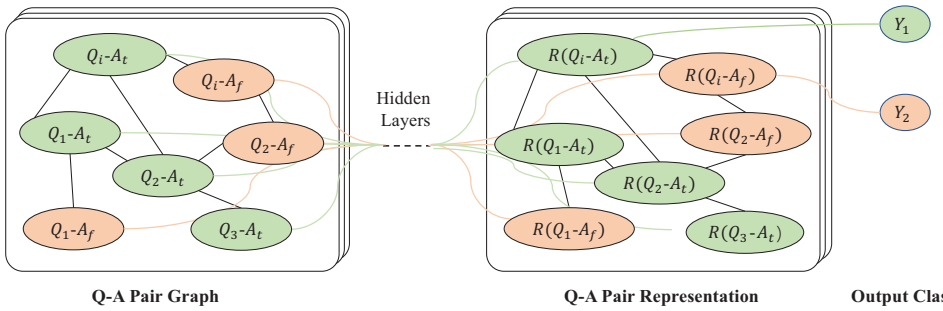

Figure 3: Cross-pair learning with the Q-A pair graph and GCN. Colors denote different classes.

In order to increase flexibility and label efficiency, we apply Auto-Regressive (AR) filter to get an improved GCN (IGCN) (Li et al. 2019), which replaces $\tilde{A}$ with $\tilde{A}'$:

$$Z^{(1)} = \rho(\tilde{A}' X W^{(0)}), \tag{5}$$

where $\tilde{A}' = p_{ar}(L) = (I + \alpha L)^{-1}$ is the AR filter and is approximated with polynomial expansion:

$$(I + \alpha L)^{-1} = \frac{1}{1+\alpha} \sum_{i=0}^{+\infty} [\frac{\alpha}{1+\alpha} \tilde{A}]^i, (\alpha > 0), \tag{6}$$

where $L = D - A$ is the graph Laplacian. $\tilde{A}' X = \frac{1}{1+\alpha} T^{(k)}$ is computed iteratively with:

$$T^{(0)} = \mathbf{O}, T^{(1)} = X, ..., T^{(i+1)} = X + \frac{\alpha}{1+\alpha} \tilde{A} T^{(i)}, \tag{7}$$

and $k = \lceil 4\alpha \rceil$ is enough according to (Li et al. 2019).

IGCN can achieve label efficiency by using the exponent k to conveniently adjust the filter strength. In this way, it can maintain a shallow structure with a reasonable number of trainable parameters to avoid overfitting.

Intra-Pair Learning

The intra-pair matching part is under the Compare-Aggregate framework, where vector representations of small units (such as words) of sentences are compared to capture interactive features, and then aggregated to calculate the final relevance score. In order to learn more comprehensive intra-pair information, we compute both textual attention E^c and knowledge-based attention E^e for the model:

$$\begin{aligned} E^c_{ij} &= \text{score}(Q^c_i, A^c_j) = Q^c_i \cdot A^c_j, \\ E^e_{ij} &= \text{score}(Q^e_i, A^e_j) = \tanh({Q^e_i}^\top U A^e_j), \end{aligned} \tag{8}$$

where Q^c and A^c are pre-trained word embedding matrixes of the question and the answer respectively, and Q^e and A^e are knowledge embedding matrixes based on entity linking results and knowledge graph (KG) entity embeddings. U is a parameter matrix to be learned.

Then with the softmax computation along the dimension j, we get $E^{c,q}_{ij}$ and $E^{e,q}_{ij}$. Similarly, $E^{c,a}_{ij}$ and $E^{e,a}_{ij}$ are computed along i. We merge textual and knowledge-based attention, and obtain the final attention vectors:

$$\begin{aligned} E^q_{ij} &= \frac{\exp(E^{c,q}_{ij} + E^{e,q}_{ij})}{\sum_{k=1}^{l_a} (\exp E^{c,q}_{ik} + E^{e,q}_{ik})}, \\ E^a_{ij} &= \frac{\exp(E^{c,a}_{ij} + E^{e,a}_{ij})}{\sum_{k=1}^{l_q} (\exp E^{c,a}_{kj} + E^{e,a}_{kj})}. \end{aligned} \tag{9}$$

Attention-weighted sums are computed as $H^{c,q}_i = \sum_{j=1}^{l_a} E^q_{ij} A^c_j$ and $H^{c,a}_j = \sum_{i=1}^{l_q} E^a_{ij} Q^c_i$ for textual representations. Analogously, knowledge-based representations are $H^{e,q}_i = \sum_{j=1}^{l_a} E^q_{ij} A^e_j$ and $H^{e,a}_j = \sum_{i=1}^{l_q} E^a_{ij} Q^e_i$.

Then unit-level comparisons match each unit of one sequence with a weighted version of its counterpart:

$$\begin{aligned} T^{c,q}_i &= \textbf{CMP}(Q^c_i, H^{c,q}_i) = Q^c_i \otimes H^{c,q}_i, \\ T^{c,a}_j &= \textbf{CMP}(A^c_j, H^{c,a}_j) = A^c_j \otimes H^{c,a}_j, \end{aligned} \tag{10}$$

where $\otimes$ is the element-wise multiplication. Analogously for knowledge-based representations, we get $T^{e,q}_i$ and $T^{e,a}_j$.

Further, the aggregation process is conducted based on CNN as suggested in (Bian et al. 2017):

$$\begin{aligned} R^{c,q} &= \textbf{AGG}([T^{c,q}_1, ..., T^{c,q}_{l_q}]) = \text{CNN}([T^{c,q}_1, ..., T^{c,q}_{l_q}]), \\ R^{c,a} &= \textbf{AGG}([T^{c,a}_1, ..., T^{c,a}_{l_a}]) = \text{CNN}([T^{c,a}_1, ..., T^{c,a}_{l_a}]). \end{aligned} \tag{11}$$

Similarly, we get final knowledge-based representations $R^{e,q}$ and $R^{e,a}$. Then outputs of the Aggregation Layer are concatenated to predicate the probability that Q and A form a pair by a full connection layer with sigmoid.

Experiments

Experimental Settings

Datasets We evaluate our model on two widely adopted QA benchmark datasets: WikiQA (Yang, Yih, and Meek 2015) and TrecQA (Wang, Smith, and Mitamura 2007). WikiQA is an open domain factoid answer selection benchmark. We adopt the standard setup (Yang, Yih, and Meek 2015) of only considering questions with correct answers for evaluation. TrecQA has clean and raw versions. The clean version removes questions that have only positive/negative answers or no answers. We evaluate on the clean version as noted by (Rao, He, and Lin 2016). The details of these two datasets are shown in Table 1. Evaluation measures are mean average precision (MAP) and mean reciprocal rank (MRR).

Table 1: Summary statistics of datasets.

Dataset	Type	Question	QA Pairs	%Correct	Nodes	Edges
WikiQA	Train	873	8672	12.0	12.2K	21.9M
	Dev	126	1130	12.4		
	Test	243	2351	12.5		
TrecQA (original/cleaned)	Train	1229/1160	53417/53313	12.0/11.8	12.5K	40.3M
	Dev	82/65	1148/1117	19.3/18.4		
	Test	100/68	1517/1442	18.7/17.2		

Table 2: Hyperparameters.

Hyperparameter		Method	
Name	*Definition*	*Intra-Pair*	*Cross-Pair*
λ	Learning rate	0.001	0.01
p	Dropout rate	0.2	0.5
L_2	L_2 normalization	0	0.0005
m	Batch size	4	1
w	Conv. size	[1,2,3,4,5]	1
h	Hidden layer size	300	(64)
τ	Edge threshold	-	0.95
r	Neg. rate	-	1:1

Common Training Setup For intra-pair learning, we use pre-trained GloVE embeddings (Pennington, Socher, and Manning 2014) for text and TransE (Bordes et al. 2013) embeddings for entities with a subset of Freebase (Bollacker et al. 2008): FB5M (4,904,397 entities, 7,523 relations and 22,441,880 facts) as KG following (Shen et al. 2018). We adopt listwise learning and use KL-divergence as the loss function as suggested in (Bian et al. 2017).

For the cross-pair part, graph node features are BERT (Devlin et al. 2019) embeddings of Q-A pairs, and the threshold τ is tuned to balance between quantity and quality of edges. A graph is built on a whole QA corpus (under-sampled on TrecQA) to capture cross-pair patterns (summarized in Table 1) with labels of the validation and testing sets masked following (Yao, Mao, and Luo 2019). Since negative answers are less useful for learning explicit Q-A patterns, and may introduce noise during propagation (if not randomly selected, typical wrong Q-A patterns may also help), we also apply sample masks on training data to tune the negative sampling rate r. We set the filter parameter $\alpha = 10$ for WikiQA and $\alpha = 1$ for TrecQA according to label rates (Li et al. 2019). GCN applies the first-order convolutional filter to integrate graph and feature information. For stacked GCN, the hidden layer size is set to 64. Adam (Kingma and Ba 2014) is adopted for training and the model with the lowest training loss in 400 steps is selected (Li et al. 2019). Other hyperparameters are shown in Table 2.

Results and Analysis

Comparison with the State of the Art Experimental results are summarized in Table 3 and nine baselines are adopted. Among which, CNN-Cnt (Yang, Yih, and Meek 2015) and HyperQA (Tay, Tuan, and Hui 2018a) are under the Siamese framework, AP-CNN (Santos et al. 2016), IWAN (Shen, Yang, and Deng 2017) and MCAN-FM (Tay, Tuan, and Hui 2018b) are attentive networks. KABLSTM (Shen et al. 2018) is a knowledge-aware attentive network. BiMPM (Wang, Hamza, and Florian 2017) and DCA (Bian et al. 2017) are built on the Compare-Aggregate architecture. SUM (Tymoshenko and Moschitti 2018) also applies both intra-pair and cross-pair learning, but in a more traditional way. It computes cross-pair relations with scalar products and ensembles different kernels-based SVM classifiers. For KCG, we implement it by using learned knowledge representations for aggregation (KCG_{eca}), or only applying knowledge to the Attention Layer and Comparison Layer (KCG_{ec}) to enhance Q-A pair representations.

We observe that KCG demonstrates significant gains over the baselines based on the intra-pair comparing frameworks, and outperforms the state-of-the-art systems on both WikiQA and TrecQA. The improvement on WikiQA is much more obvious than TrecQA. TrecQA includes editor-generated questions and candidate answer sentences selected by word matching (Wang, Smith, and Mitamura 2007), while WikiQA is constructed in a natural and realistic manner based on query logs and Wikipedia pages (Yang, Yih, and Meek 2015). Therefore, WikiQA is more lexically diverse and closer to real-world scenarios.

Ablation Study In order to analyze the effectiveness of different factors, we also report the ablation tests in terms of discarding cross-pair matching part (w/o IGCN) and knowledge graph information (w/o KG) respectively. The bottom of Table 3 shows ablation results on KCG_{ec}.

First, leaving out cross-pair matching (w/o IGCN) impacts the performance, and the drop is more significant on WikiQA (0.784 to 0.768 for MAP). It suggests that cross-pair matching helps to improve the performance on complex cases where intra-pair models may be insufficient. In fact, (Yih et al. 2013) found that simple word matching outperforms many sophisticated approaches on TrecQA. Therefore, intra-pair matching alone performs well on the dataset.

Second, note that KG is also a main contributor to the performance, which indicates the importance of background information for QA tasks. Knowledge-aware models enrich the representation learning of Q-A pairs with external knowledge. Aggregating learned knowledge representations with sentence representations in the end (KCG_{eca}), however, does not ensure further improvement. This may depend on the entity distributions in specific datasets.

Third, we have the hypothesis that KCG manages to meet both knowledge and pattern conditions, thus handling the AS problem based on the nature of QA. Experiments suggest that KCG substantially outperforms some well-designed models under PI comparing frameworks, and models only focusing on one condition (Shen et al. 2018; Tymoshenko and Moschitti 2018).

Comparison between Different Graph-Based Methods We also conduct experiments to compare effects of different graph-based methods for cross-pair learning. Results are presented in Table 4. The classic label propagation (Zhou et al. 2004) contributes little to the raw model. This method only makes predictions based on the graph structure, which is inadequate without representation learning of Q-A pairs. The model with GCN has achieved better results because of the first-order convolutional filter which integrates graph and feature information. However, GCN usually needs stacked layers to increase smoothness, and thus it is difficult to train with fewer labels due to high model complexity.

Table 3: Results on WikiQA and TrecQA datasets.

Framework	Method	WikiQA		TrecQA	
		MAP	***MRR***	***MAP***	***MRR***
Siamese	CNN-Cnt (Yang, Yih, and Meek 2015)	0.652	0.665	0.695	0.763
	HyperQA (Tay, Tuan, and Hui 2018a)	0.712	0.727	0.784	0.865
Attentive	AP-CNN (Santos et al. 2016)	0.689	0.696	0.753	0.851
	KABLSTM (Shen et al. 2018)	0.732	0.749	0.804	0.885
	IWAN (Shen, Yang, and Deng 2017)	0.733	0.750	0.822	0.889
	MCAN-FM (Tay, Tuan, and Hui 2018b)	-	-	0.838	0.904
Compare-Aggregate	BiMPM (Wang, Hamza, and Florian 2017)	0.718	0.731	0.802	0.875
	DCA (Bian et al. 2017)	0.754	0.764	0.821	0.899
Intra-Cross	SUM (Tymoshenko and Moschitti 2018)	0.762	0.776	0.777	0.869
	KCG_{eca}	0.772	0.786	**0.857**	**0.908**
	KCG_{ec}	**0.784**	**0.802**	0.841	0.902
	w/o IGCN	0.768	0.782	0.832	0.900
	w/o KG	0.763	0.778	0.828	0.889

Table 4: Results of replacing cross-pair learning part by different graph-based methods on WikiQA.

Cross-Pair Part	Layer	***MAP***	***MRR***
-	-	0.768	0.782
LP (Zhou et al. 2004)	1	0.770	0.785
	2	0.771	0.785
GCN (Kipf and Welling 2017)	1	0.774	0.787
	2	0.773	0.788
IGCN (Li et al. 2019)	1	**0.784**	**0.802**
	2	0.774	0.787

Table 5: The facilitation of cross-pair learning to various matching or binary classification models on WikiQA.

Intra-Pair	Cross-Pair	***MAP***	***MRR***
TextCNN (Kim 2014)	-	0.528	0.542
TextCNN	IGCN	0.664	0.683
Transformer (Vaswani et al. 2017)	-	0.641	0.644
Transformer	IGCN	0.668	0.681
Dilated CNN (Yu and Koltun 2016)	-	0.658	0.659
Dilated CNN	IGCN	0.677	0.690
ABCNN (Yin et al. 2016),	-	0.692	0.713
ABCNN	IGCN	**0.721**	**0.742**
ABCNN	Transformer	0.693	0.714

KCG with IGCN has achieved the best performance. IGCN improves GCN with low-pass graph convolutional filters to generate smooth and representative features for subsequent classification. Through flexibly adjusting the filter strength, it can significantly reduce trainable parameters and effectively prevent overfitting.

Note that two-layer models do not show better performance. Two layers allow exchange of information among nodes that are at maximum two steps away. However, noise can be introduced from some randomly selected negative samples. Constraints may be put on the negative-sampling of answers to further improve the performance, which we leave for future work.

Additional Analysis on GCN-Based Post-Procedure To further analyze the effectiveness and potential of the GCN-based post-procedure, we reimplement several classic intra-pair matching or binary classification models with the procedure on WikiQA, as shown in Table 5. Generally, it makes

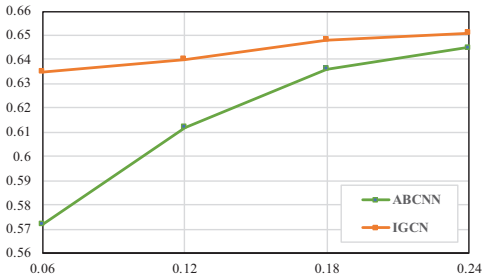

Figure 4: Test MAP results on WikiQA by varying training data proportions on ABCNN and IGCN.

performance boost to apply GCN-based cross-pair learning. In particular, the typical attentive network ABCNN (Yin et al. 2016) achieves competitive results with some strong baselines (Table 3). The attentive pooling mechanism in ABCNN considers local relations within pairs, but it does not cover global correlations as in cross-pair learning.

Considering that deep neural networks themselves may implicitly capture cross-pair similarity during training, we replace the cross-pair part with a classification model based on Transformer (Vaswani et al. 2017) (the last line). However, the change leads to a drop in the results, which further demonstrates the effectiveness of the immediate cross-pair modeling approach. The results also suggest that model integration is not the key factor of good performance here.

Further, in order to evaluate the label efficiency of the procedure, we test it alone with different proportions of training data. Figure 4 compares IGCN cross-pair learning with ABCNN on 6%, 12%, 18% and 24% of the WikiQA training set. Note that IGCN can achieve better MAP with limited training data, which is similar to the result in (Kipf and Welling 2017), where GCN performs well with low label rate. The results again suggest that our graph preserves global Q-to-A pattern information, and GCN can make better use of the corpus through propagating information among nodes.

The GCN-based post-procedure incorporates global information and brings progress over different basic models. In

Table 6: Examples of answer selection results.

ID	*Question*	*KCG*	*DCA*
1	What is the **color puce**?	**Puce** (often misspelled as "puse", "peuse" or "peuce") is defined in the United States as a brownish-purple **color**.	The **colors** in the boxes at right are two of the various shades and varieties of **puce**.
2	Who set the **world record** for **women** for **high jump**?	Stefka Kostadinova (Bulgaria) has held the **women's world record** since 1987, also the longest-held record in the event.	The **high jump** is a track and field athletics event in which competitors must **jump** over a horizontal bar placed at measured heights...
3	How many **numbers** are on a **credit card**?	An ISO/IEC 7812 **card number** is typically 16 digits in length.	**Bank card numbers** are allocated in accordance with isoiec 7812.
4	What is the **formula** for **calcium nitrate**?	**Calcium nitrate**, also called Norgessalpeter (Norwegian saltpeter), is an inorganic compound with the **formula** Ca(NO3)2.	Nitrocalcite is the name for a mineral which is a hydrated **calcium nitrate** that **forms** as an efflorescence...
5	Who are the members of the **climax blues band**?	The original members were guitarist/vocalist Peter Haycock, guitarist Derek Holt; keyboardist Arthur Wood; bassist Richard Jones...	The **Climax Blues Band** (originally known as the **Climax** Chicago **Blues Band**) were formed in Stafford, England in 1968.

the full-batch setup, the adjacency matrix in the sparse form has a linear relationship with non-zero elements, and the feature matrix grows linearly with the dataset size. To expand to large datasets, graph partitioning (Abbas et al. 2018) is usually adopted. Simple trade-off methods like adjusting the edge threshold and using fewer training samples are also effective due to the label efficiency of the method.

Case Study Considering the variety of Q-A pairs, some top ranked answers are demonstrated in Table 6 for further analysis. We reimplement DCA (Bian et al. 2017), a baseline model with the Compare-Aggregate architecture, to support comparisons in the case study. We use bold fonts to denote intra-pair matches and underlines to mark cross-pair ones.

We can see that DCA does not handle some cases well because of its sensitiveness to intra-pair key words, while KCG is generally more robust against the diversity of Q-A pairs. KCG considers both pattern and knowledge conditions. With the incorporation of global Q-to-A pattern information, KCG works more effectively on different question structures. Besides, knowledge information helps KCG to capture various relations within Q-A pairs, and prevents it from over-learning of some uncommon Q-to-A patterns.

Conclusion and Future Work

Answer selection is a basic semantic matching problem in QA. In this paper, we propose a novel system named KCG for AS, which considers both knowledge and pattern conditions. KCG applies both intra-pair and cross-pair learning. For intra-pair matching, it makes comparison on both textual and knowledge information within the Q-A pair. For cross-pair matching, it explores global information with the QA-pair graph and GCN, thus incorporating more general Q-to-A patterns and making better use of the corpus. Compared with multiple baselines, QURRD achieves state-of-the-art results on both WikiQA and TrecQA. For future work, we will construct heterogeneous graphs with both sentences and entities to better model cross-pair dependencies, and extend KCG to more general semantic matching scenarios.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 61773229 and 61972219), Shenzhen Giiso Information Technology Co. Ltd., the National Natural Science Foundation of Guangdong Province (Grant No. 2018A030313422), and Overseas Cooperation Research Fund of Graduate School at Shenzhen, Tsinghua University (Grant No. HW2018002).

References

Abbas, Z.; Kalavri, V.; Carbone, P.; and Vlassov, V. 2018. Streaming graph partitioning: an experimental study. *Proceedings of the VLDB Endowment* 11(11):1590–1603.

Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; and Simaan, K. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1957–1967.

Bian, W.; Li, S.; Yang, Z.; Chen, G.; and Lin, Z. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1987–1990. ACM.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. AcM.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1870–1879.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Feng, M.; Xiang, B.; Glass, M. R.; Wang, L.; and Zhou, B. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 813–820. IEEE.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, 2042–2050.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Lai, Y.; Feng, Y.; Yu, X.; Wang, Z.; Xu, K.; and Zhao, D. 2019. Lattice cnns for matching based chinese question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6634–6641.

Li, Q.; Wu, X.-M.; Liu, H.; Zhang, X.; and Guan, Z. 2019. Label efficient semi-supervised learning via graph filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9582–9591.

Li, Y.; Jin, R.; and Luo, Y. 2018. Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seg-gcrns). *Journal of the American Medical Informatics Association* 26(3):262–268.

Li, H.; Xu, J.; et al. 2014. Semantic matching in search. *Foundations and Trends® in Information Retrieval* 7(5):343–469.

Marcheggiani, D., and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1506–1515.

Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; and Cheng, X. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rao, J.; He, H.; and Lin, J. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1913–1916. ACM.

Santos, C. d.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Shen, Y.; Deng, Y.; Yang, M.; Li, Y.; Du, N.; Fan, W.; and Lei, K. 2018. Knowledge-aware attentive neural network for ranking question answer pairs. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 901–904. ACM.

Shen, G.; Yang, Y.; and Deng, Z.-H. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1179–1189.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018a. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 583–591. ACM.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018b. Multi-cast attention networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2299–2308. ACM.

Tymoshenko, K., and Moschitti, A. 2018. Cross-pair text representations for answer sentence selection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2162–2173.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. Curran Associates Inc.

Wang, S., and Jiang, J. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Wang, Z.; Hamza, W.; and Florian, R. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4144–4150. AAAI Press.

Wang, M.; Smith, N. A.; and Mitamura, T. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 22–32.

Yang, Y.; Yih, W.-t.; and Meek, C. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2013–2018.

Yao, X., and Van Durme, B. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 956–966.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7370–7377.

Yih, W.-t.; Chang, M.-W.; Meek, C.; and Pastusiak, A. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1744–1753.

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

Yu, F., and Koltun, V. 2016. Multi-scale context aggregation by dilated convolutions. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Zanzotto, F. M., and Moschitti, A. 2006. Automatic learning of textual entailments with cross-pair similarities. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 401–408.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *Advances in neural information processing systems*, 321–328.