Exploiting WordNet Synset and Hypernym Representations for Answer Selection

Weikang Li and Yunfang Wu*
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{wavejkd, wuyf}@pku.edu.cn

Abstract

Answer selection (AS) is an important subtask of document-based question answering (DQA). In this task, the candidate answers come from the same document, and each answer sentence is semantically related to the given question, which makes it more challenging to select the true answer. WordNet provides powerful knowledge about concepts and their semantic relations, so we employ WordNet to enrich the abilities of paraphrasing and reasoning of the network-based question answering model. Specifically, we exploit the synset and hypernym concepts to enrich the word representation and incorporate the similarity scores of two concepts that share the synset or hypernym relations into the attention mechanism. The proposed WordNet-enhanced hierarchical model (WEHM) consists of four modules, including WordNet-enhanced word representation, sentence encoding, WordNet-enhanced attention mechanism, and hierarchical document encoding. Extensive experiments on the public WikiQA and SelQA datasets demonstrate that our proposed model significantly improves the baseline system and outperforms all existing state-of-the-art methods by a large margin.

1 Introduction

Answer selection (AS) is a challenging subtask of document-based question answering (DQA) in natural language processing (NLP). The AS task is to select a whole answer sentence from the document and can be regarded as a ranking problem, which is different from the machine reading comprehension (MRC) task on the SQuAD and MS-MARCO datasets. Compared with a single word or phrase, returning the full sentence often adds more value as the user can easily verify the correctness without reading a lengthy document (Yih et al., 2013). In this paper, we focus on the AS task of DQA. Table 1 gives a real example of this task.

Lots of fruits on answer selection have been achieved via deep learning models, including convolutional neural network (CNN) (Yang et al., 2015), recurrent neural network (RNN) (Tan et al., 2015), attention-way (Wang et al., 2016) and generative adversarial networks (GAN) (Wang et al., 2017a). Recently proposed models often consist of an embedding layer, an encoding layer, an interaction layer, and an answer layer (Weissenborn et al., 2017; Wang et al., 2017b; Hewlett et al., 2017).

Different from other question answering like community-based question answering, the candidate answers of DQA come from the same document, and each candidate answer is semantically related to the question. From the example in Table 1, we can see that almost every candidate answer contains the information related to the word "food" and "afghan" in the given question. As a result, it is difficult for the existing network-based models to choose the right answer, since the power generation ability of the networks may have transformed the sentences into similar meanings in the latent space.

To tackle this challenge, we propose to leverage WordNet knowledge base into the neural network model. Our hypothesis is that the ability of paraphrase and reasoning is essential to the question-answering task. WordNet is a semantic network (Fellbaum, 1998), where the words that are related in meanings are interlinked by means of pointers, which stand for different semantic relations. It organizes concepts mainly with the is-a relation, where a concept is a set of word senses (synset). On the one hand, we apply the synset information to enrich the sentence's paraphrase representation, which could distinguish the candidate answers in the latent semantic space to some degree. On the other hand, we apply the hypernym information to capture reasoning knowledge. The real case

* Corresponding author.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 106–115

Question: what food is in afghan ?
Document:
[1] A table setting of Afghan food in Kabul.
[2] Afghan cuisine is largely based upon the nation's chief crops; cereals like wheat, maize, barley and rice.
[3] .....
[4] Afghanistan's culinary specialties reflect its ethnic and geographic diversity.
[5] Though it has similarities with neighboring countries, Afghan cuisine is undeniably unique.
[6] .....
Reference Answer:
Afghan cuisine is largely based upon the nation's chief crops; cereals like wheat, maize, barley and rice.

Table 1: An example from the WikiQA data. The text is shown in its original form, which may contain errors in typing.

from the WikiQA dataset in table 1 shows that if our model has the ability of reasoning on common sense, like "wheat is a kind of food", "maize is a kind of food" and so on, it would be of great help for choosing the right answer with respect to the question "what food is in afghan ?".

The overall framework of our proposed model is shown in Figure 1, which mainly consists of four modules. First, we apply the synset and hypernym information to enrich the word representation. Second, we use an RNN module to encode the WordNet-enhanced word representation. Third, we propose to use the synset's and hypernym's relation score based on two senses' path in the WordNet to enrich the attention mechanism. Specifically, the attention similarity matrix is not only measured by a similarity score over hidden vectors produced by CNN or RNN networks but also measured based on the synset and hypernym relation scores of two concepts in Wordnet. And then following the compare-aggregate framework (Wang and Jiang, 2016), we combine the original representation with the attention representation. Finally, considering the strong relations among context sentences, we employ a hierarchical neural network for answer sentence selection.

We conduct extensive experiments on the public WikiQA and SelQA datasets. The results show that our proposed WordNet-enhanced hierarchical model outperforms the baseline models by a large margin and achieves state-of-the-art performance on both datasets. On the WikiQA data, it obtains a MAP of 77.02, which beats the existing best result by 1.62 points; on the SelQA data, it achieves a MAP of 91.71, which outperforms the previous best result by 2.57 points.

2 Model Description

Given a question q and the sentences $a_i, i = 1, 2, ..., S$ in a document d, our model aims to select the best sentence which could answer the question.

2.1 WordNet-Enhanced Word Representation

Firstly, we map each word into the vector space. Different from directly using word embedding or the concatenation of word embedding and sum of its character embeddings, we propose to exploit the word's hypernym and synset in the WorNet to enrich the word representation. Suppose w_j is the j_{th} word in a sequence, k_{sj} and k_{hj} represent the hypernym and synset in the WordNet with respect to the word w_j. The WordNet-enhanced word embedding is computed as follows:

$$k_j = [w_j; k_{sj}; k_{hj}] \tag{1}$$

$$k_{sj} = \frac{1}{|S|} \sum_{i=1}^{|S|} w_i^{k_{sj}} \tag{2}$$

$$k_{hj} = \frac{1}{|H|} \sum_{i=1}^{|H|} w_i^{k_{hj}} \tag{3}$$

where $w_i^{k_{sj}}$ and $w_i^{k_{hj}}$ represent word embeddings in the synset and hypernym concepts respectively; $|S|$ and $|H|$ denote the number of concepts in the synset and hypernym respectively. And ; means the concatenation operation.

We use k_j^q and $k_j^{a_i}$ to represent the j_{th} word's WordNet-enhanced embedding of the question and the i_{th} candidate answer sentence respectively.

2.2 Sentcene Encoding

We encode the question and each sentence in the document into latent vectors using a Bi-directional Gated Recurrent Unit (Bi-GRU) network. The formulas of a GRU (Cho et al., 2014) are as follows:

$$r_j = \sigma(W_r k_j + U_r h_{j-1} + b_r) \tag{4}$$

$$z_j = \sigma(W_z k_j + U_z h_{j-1} + b_z) \tag{5}$$

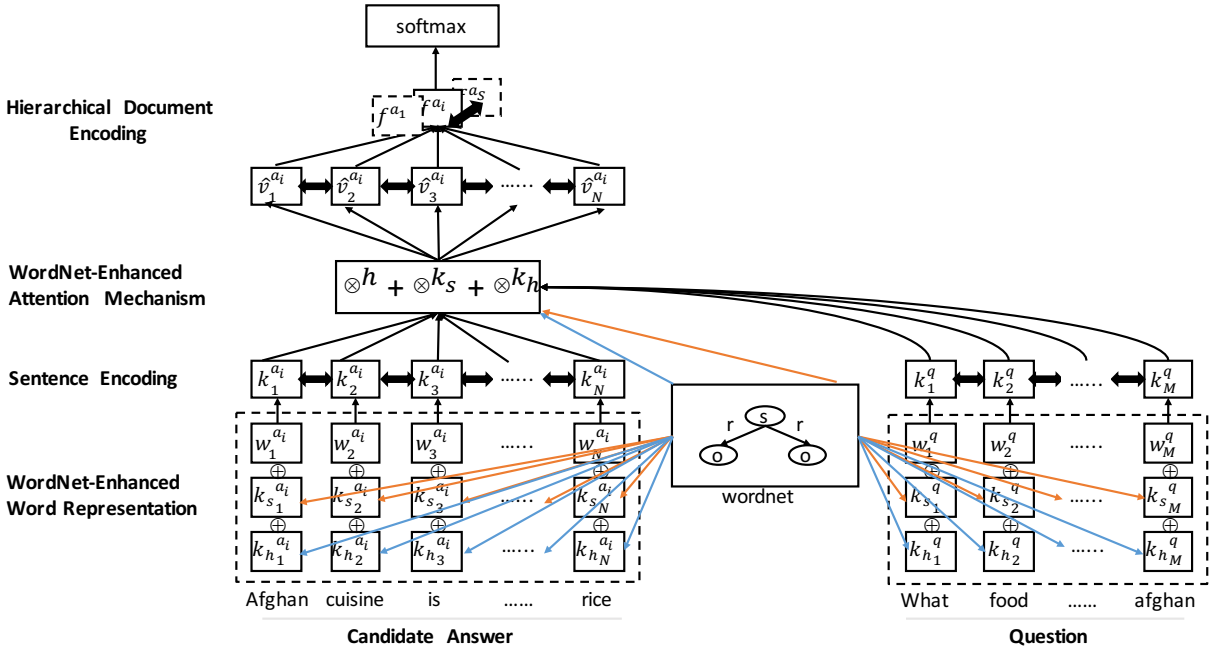


Figure 1: Framework of our proposed WordNet-enhanced hierarchical model (WEHM).

$$\widetilde{h}_j = \tanh(W_h k_j + U_h(r_j \odot h_{j-1}) + b_h) \quad (6)$$

$$h_j = (1 - z_j) \odot h_{j-1} + z_j \odot \widetilde{h}_j \quad (7)$$

where $\odot$ is element-wise multiplication. r_j and z_j are the reset and update gates respectively. And $W_r, W_z, W_h \in R^{H \times E}$, $U_r, U_z, U_h \in R^{H \times H}$ and $b_r, b_z, b_h \in R^{H \times 1}$ are parameters to be learned. A Bi-GRU processes the sequence in both forward and backward directions to produce two sequences $[h_1^f, h_2^f, ..., h_S^f]$ and $[h_1^b, h_2^b, ..., h_S^b]$. The final output of h_j is the concatenation of h_j^f and h_j^b.

We use h_j^q and $h_j^{a_i}$ to represent j_{th} word's hidden vector produced by sentence encoding in the question and in the i_{th} candidate answer sentence respectively.

2.3 WordNet-Enhanced Attention Mechanism

Different from the vanilla attention mechanism, where the attention score is only measured by hidden vectors, we propose to employ the synset and hypernym relation scores of two concepts in WordNet to enhance the attention mechanism, which can capture more rich interaction information between two sequences. The sketch of our proposed WordNet-enhanced attention mechanism is shown in Figure 2, which consists of three parts: the standard attention score, the synset relation score, and the hypernym relation score.

As for the standard attention mechanism, we adopt the Luong attention (also known as bilinear function attention mechanism) (Luong et al., 2015), which is widely used in NLP. In our model, $M^h_{|a_i|,|q|}$ represents the attention score between the question and one of its candidate answers. The formulas of computing each element are as follows:

$$M^h_{n,m} = h_n^{a_i} W h_m^{q\,\mathrm{T}} \quad (8)$$

$$M^h_{n,m} = \exp(M^h_{n,m}) / \sum_{k=1}^{|q|} \exp(M^h_{n,k}) \quad (9)$$

where $h_n^{a_i}$ and h_m^q represent the n_{th} and m_{th} word hidden vector in the candidate answer and the question respectively, $|a_i|$ and $|q|$ are the candidate answer's length and the question's length respectively.

Besides the standard attention, we employ two kinds of WordNet-enhanced mechanism to measure the attention score.

Lots of studies have been done on computing lexical similarity based on WordNet (Pedersen et al., 2004). Wu-Palmer Similarity (Wu and Palmer, 1994) denotes how similar two words senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer. Leacock-Chodorow Similarity (Leacock and Chodorow, 1998) denotes how similar two-word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hyponym) taxonomy.

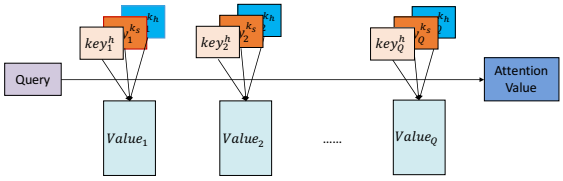


Figure 2: Sketch of our proposed WordNet-enhanced attention mechanism. Key_j^h means the attention score derived by two hidden vectors. $Key_j^{k_s}$ and $Key_j^{k_h}$ represent the attention score derived by synset relation and hypernym relation respectively. $Vlaue_j$ means the hidden vector of question, and $Query$ means the candidate answer.

We use Wu-Palmer Similarity to compute the attention score with the synset relation. $M^{k_s}_{|a_i||q|}$ represents the attention matrix between the question and one of its candidate answers, where each element $M^{k_s}_{n,m}$ is computed as:

$$M^{k_s}_{n,m} = 2 * N_c/(N_{a^i_n} + N_{q_m} + 2 * N_c) \quad (10)$$

$$M^{k_s}_{n,m} = \exp(M^{k_s}_{n,m})/\sum_{k=1}^{|q|} \exp(M^{k_s}_{n,k}) \quad (11)$$

where a^i_n and q_m represent the corresponding concepts of the nth word of the ith candidate answer and the mth word of the question respectively, c is the least common superconcept of a^i_n and q_m, $N_{a^i_n}$ is the number of nodes on the path from a^i_n to c, N_{q_m} is the number of nodes on the path from N_{q_m} to c, N_c is the number of nodes on the path from c to root.

We use Leacock-Chodorow Similarity to measure the attention score with hypernym relation. Let $M^{k_h}_{|a_i||q|}$ denote the attention matrix between the question and one of its candidate answers, where each element $M^{k_h}_{n,m}$ can be computed as:

$$M^{k_h}_{n,m} = -log(path(a^i_n, q_m)/2L) \quad (12)$$

$$M^{k_h}_{n,m} = \exp(M^{k_h}_{n,m})/\sum_{k=1}^{|q|} \exp(M^{k_h}_{n,m}) \quad (13)$$

where $path(a^i_n, q_m)$ is the shortest path length connecting two concepts and L is the whole taxonomy depth.

Finally, we combine all the three similarity matrixes. The formulas are as follows:

$$M_{n,m} = M^h_{n,m} + M^{k_s}_{n,m} + M^{k_h}_{n,m} \quad (14)$$

$$M_{n,m} = \exp(M_{n,m})/\sum_{k=1}^{|q|} \exp(M_{n,k}) \quad (15)$$

Equipped with the WordNet-enhanced similarity matrix M, we apply the attention mechanism between the question encoding h^q and the sentence encoding h^{a_i} to obtain a new sentence representation v^{a_i}, which is a weighted sum of hidden vectors of the question. We then aggregate the vectors of h^{a_i} and v^{a_i}. Formulas are as follows:

$$v^{a_i} = M \cdot h^q \quad (16)$$

$$\hat{v}^{a_i} = [v^{a_i}; h^{a_i}; v^{a_i} \odot h^{a_i}; v^{a_i} + h^{a_i}; v^{a_i} - h^{a_i}] \quad (17)$$

where ; is the concatenation operation, $+$ is element-wise addition, $-$ is element-wise subtraction and $\odot$ is element-wise multiplication.

2.4 Hierarchical Document Encoding

Inspired by the work (Bian et al., 2017), we also adopt a list-wise method to model the answer selection task. But different from their model, we employ a hierarchical Bi-GRU architecture to compare candidate sentences by ranking them with respect to a given question. Considering that candidate answers all come from a whole document, the hierarchical Bi-GRU architecture can capture contextual features among sentences and make the understanding of a document more coherent.

We first encode each candidate answer $\hat{v}^{a_i}$ and then extract features among sentences' hidden vectors. Then we again encode the document based on each candidate answer's extracted features. The Bi-GRU is the same to that mentioned in our sentence encoding section.

$$u^{a_i}_j = BiGRU\left(u^{a_i}_{j-1}, \hat{v}^{a_i}_j\right) \quad (18)$$

$$u^{a_i}_{avg} = \frac{1}{|a_i|}\sum_{j=1}^{|a_i|} u^{a_i}_j, u^{a_i}_{max} = \max_{j=1}^{|a_i|} u^{a_i}_j \quad (19)$$

$$f^{a_i} = \left[u^{a_i}_{avg}; u^{a_i}_{max}\right] \quad (20)$$

$$\hat{u}^d_i = BiGRU\left(\hat{u}^d_{i-1}, f^{a_i}\right) \quad (21)$$

where j is the j_{th} word in the i_{th} sentence in the candidate answers, f^{a_i} is the i_{th} sentence extracted features and $\hat{u}^d_i$ is the i_{th} sentence's hidden vector after the document encoding phase.

At last, we use a $softmax$ layer to choose the right answer among every step's output of the document's RNN layer. The model is trained to minimize the cross-entropy loss function:

$$\tilde{a}_i = \sigma(FC(\hat{u}^d_i)) \quad (22)$$

Dataset	Split	#Questions	#Pairs
WikiQA	TRAIN	873	8672
	DEV	126	1130
	TEST	243	2351
SelQA	TRAIN	5529	66438
	DEV	785	9377
	TEST	1590	19435

Table 2: Statistical distribution of two benchmark datasets.

$$C = -\frac{1}{|d|}\sum_{i\in|d|}[a_i \log \tilde{a}_i + (1-a_i)\log(1-\tilde{a}_i)] \quad (23)$$

where FC is a feed-forward neural network, i means the sentence index in the document, $|d|$ is the document's length in terms of sentences, a_i is the true label (0 or 1) from the training data and $\tilde{a}_i$ is the predicted probability score by our model. The sentence with the highest probability score is regarded as the right answer.

3 Experiments

3.1 Datasets and Baselines

We use two different datasets to conduct our answer selection experiments: WikiQA (Yang et al., 2015) and SelQA (Jurczyk et al., 2016). Both datasets contain open-domain questions whose answers were extracted from Wikipedia articles. In the AS task, it is assumed that there is at least one correct answer for a question. In the WikiQA, there are some questions which have no answer, we removed these questions, just like other researches do. Table 2 shows the statistical distribution of the two datasets.

As for the WikiQA dataset, it has been well studied by lots of literature. Baselines adopted are as follows:

- **CNN-Cnt:** this model combines sentence representations produced by a convolutional neural network with the logistic regression (Yang et al., 2015).
- **ABCNN:** this model is an attention-based convolutional neural network (Yin et al., 2015).
- **IARNN-Occam:** this model adds regularization on the attention weights (Wang et al., 2016).
- **IARNN-Gate:** this model uses the question representation to build GRU gates for each candidate answer (Wang et al., 2016).
- **CubeCNN:** this model builds a CNN on all pairs of word similarities (He and Lin, 2016).
- **CA-Network:** this model applies a compare-aggregate neural network to model question answering problem (Wang and Jiang, 2016).
- **IWAN-Skip**: this model measures the similarity of sentence pairs by focusing on the interaction information (Shen et al., 2017b).
- **Dynamic-Clip**: this model proposes a novel attention mechanism named Dynamic-Clip Attention, which is then directly integrated into the Compare-Aggregate framework. (Bian et al., 2017).

As for the SelQA dataset, besides the above mentioned CNN-Cnt model, Jurczyk et al. (2016) also re-implement CNN-Tree and two attention RNN models. Other baselines are as follows:

- **CNN-hinge:** this is a re-implemented CNN-based model with hinge loss function (dos Santos et al., 2017).
- **CNN-DAN:** dos Santos et al. (2017) propose a CNN-based model trained with a DAN framework, which is to learn loss functions for predictors and also implements semi-supervised learning.
- **AdaQA:** Shen et al. (2017a) propose an adaptive question answering (AdaQA) model, which consists of a novel two-way feature abstraction mechanism to encapsulate co-dependent sentence representations.

The answer selection task can be considered as a ranking problem, and so two evaluation metrics are used: mean average precision (MAP) and mean reciprocal rank (MRR).

3.2 Experiment Setup

The proposed models are implemented with TensorFlow. The dimension of word embeddings is set to 300. The word embeddings are initialized by *300D GloVe 840B* (Pennington et al., 2014), and out-of-vocabulary words are initialized randomly. We fix the embeddings during training. We train the model with the Adam optimization algorithm with

Model	MAP	MRR
CNN-Cnt (Yang et al., 2015)	65.20	66.52
ABCNN (Yin et al., 2015)	69.21	71.08
CubeCNN (He and Lin, 2016)	70.90	72.34
IARNN-Gate (Wang et al., 2016)	72.58	73.94
IARNN-Occam (Wang et al., 2016)	73.41	74.18
CA-Network (Wang and Jiang, 2016)	74.33	75.45
IWAN-Skip (Shen et al., 2017b)	73.30	75.00
Dynamic-Clip (Bian et al., 2017)	75.40	76.40
WEHM (Proposed)	**77.02**	**78.82**

Table 3: Experimental results on the WikiQA dataset

Model	MAP	MRR
CNN-Cnt (Jurczyk et al., 2016)	84.00	84.94
CNN-Tree (Jurczyk et al., 2016)	84.66	85.68
RNN: one-way (Jurczyk et al., 2016)	82.06	83.18
RNN: attn-pool (Jurczyk et al., 2016)	86.43	87.59
CNN-DAN (dos Santos et al., 2017)	86.55	87.30
CNN-hinge (dos Santos et al., 2017)	87.58	88.12
AdaQA (Shen et al., 2017a)	89.14	89.83
WEHM (Proposed)	**91.71**	**92.22**

Table 4: Experimental results on the SelQA dataset

a learning rate of 0.001. Our models are trained in mini-batches (with a batch size of 10). We fix the length of the question and each sentence in the document according to their sentence's max length in each mini-batch, and any sentences not enough to this range are padded. The hidden vector size is set to 150 for a single RNN. We conduct word sense disambiguation for ambiguous words via the nltk tool.

3.3 Results and Analysis

3.3.1 Performance

We compare our model with state-of-the-art methods on the WikiQA and SelQA dataset in Table 3 and Table 4, respectively. Our proposed model not only obtains state-of-the-art performance on two datasets but also makes a significant improvement. Compared with the existing best method Dynamic-Clip, our model yields nearly 1.6% improvement in MAP and 2.4% in MRR on the WikiQA dataset. On the SelQA dataset, our model improves 2.6% in MAP and 2.4% in MRR, compared with the previous best method AdaQA. It is a challenging task for answer selection, especially for the WikiQA dataset. As is shown in Table 3, the notable CA-network outperforms the IARNN-Occam approach only by 0.92 MAP points, and our best result (77.02) achieves a performance gain of 1.6 MAP points over the Dynamic-Clip. In this sense, the improvement of our model is valuable.

Model	MAP	$\Delta/\%$
without WordNet knowledge	**84.87**	-
(1) only hypernym token	85.35	0.48
(2) only synset token	85.17	0.30
(3) only hypernym&synset token	86.32	1.45
(4) only hypernym attention	90.21	5.34
(5) only synset attention	89.99	5.12
(6) only hypernym&synset attention	90.49	5.62
WEHM	**91.71**	**6.84**

Table 5: Ablation study on the SelQA dataset

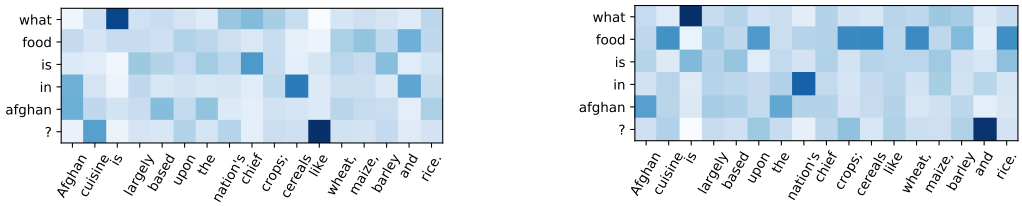
3.3.2 Ablation Study

We further conduct an ablation study to explore different WordNet-enhanced components in our model, including WordNet-enhanced word embedding and WordNet-enhanced Attention Mechanism. Table 5 reports the experimental results.

We first remove all knowledge components from our model, denoted as *without WordNet knowledge*, which can be regarded as the baseline model. In the baseline model, we only use the original word embeddings and the conventional Luong attention mechanism. Then we evaluate the WordNet-enhanced word embedding by adding the hypernym, synset, and the combination of both to the word embeddings, shown in (1)-(3) of Table 5. To evaluate the WordNet-enhanced attention mechanism, we also add the synset relation score, the hypernym score or its combination to the original hidden vectors' score based on the baseline model, shown in (4)-(6) of Table 5.

Compared with the baseline model, the WordNet knowledge brings consistent performance gain both for the WorNet-enhanced word embedding and WordNet-enhanced attention mechanism. As for the Knowledge-enhanced word embedding, the hypernym and synset improve 0.48% and 0.30% in MAP, respectively, and the combination of them improves 1.45% in MAP. As for the Knowledge-enhanced attention mechanism, the hypernym and synset improve 5.34% and 5.12% in MAP respectively, and the combination of them improves 5.62% in MAP. At the result, our full proposed model WEHM yields a significant performance gain of 6.84 MAP points.

We could find that the knowledge-enhanced attention mechanism is more effective than the simple knowledge-enriched word embedding, perhaps because computing the similarity scores of two concepts takes into account much information, like the shortest path between them and the depth of the concept in the taxonomy. Moreover, the combina-



(a) M_{vector} (b) $M_{wordnet}$

Figure 3: Attention score matrixes M_{vector} and $M_{wordnet}$ of a real case on the WikiQA dataset. The matrix $M_{wordnet}$ not only captures the paraphrase information like "food" and 'cuisine', but also enhances relations between the question's word "food" and some of the sentence's words, like "crops", "cereals", "wheat" and "rice".

tion of hypernym and synset is better than the single hypernym or synset information in both knowledge components because it captures more diverse information. Interestingly, the hypernym information is more effective than the synset information in the question-answering task.

3.3.3 Case Study

To make a detailed analysis of the effectiveness of our proposed model, we give a case study to visualize the different attention score matrix M_{vector} and $M_{wordnet}$, by a heatmap in Figure 3. M_{vector} is only computed by hidden vectors, and $M_{wordnet}$ is calculated by our proposed model. When answering the question, our proposed model not only captures the information of "food" and "afghan", but also pays more attention to the related meaning of "wheat - food", "rice - food" and so on , which brings vital information to the prediction, while the baseline method performs weakly on capturing this information.

3.3.4 Error Analysis

We further make an error analysis of our model for further improvements. Table 6 is a wrong prediction produced by our proposed model (WEHM). "Cardiovascular disease" is another name for "heart disease". However, "Cardiovascular disease" isn't mentioned in the given question. Although we have enriched the model with WordNet knowledge, it is still hard for the model to capture the lexical gap between these two words, for that their concepts are not the same in WordNet. From this analysis, we'd like to employ more fine-grained knowledge, like the clarification for proper nouns.

3.3.5 Comparison with other knowledge-enhanced models

To the best of our knowledge, we are the first to explore the WordNet knowledge to enhance the neural network model for the DQA problem. There are also some other knowledge-enhanced models designed for specific tasks, in which the natural language inference (NLI) task is somewhat similar to the QA task. In order to compare with our proposed WEHN model, we re-run the KEM model on the WikiQA dataset by using its public codes, which is designed for NLI task by Chen et al. (2018). ESIM (Chen et al., 2017) is the basic model of KEM without knowledge. KEM uses feature vectors of specific dimensions in WordNet, while our WEHM model directly employs synset and hypernym relation scores to enrich the attention score and also use their concepts to enrich the word representation. Table 7 shows the results of the WikiQA dataset. We could see that our proposed model outperforms the KEM model by a large margin. Besides, when comparing the improvements produced by the enriched knowledge, our proposed model is still better than KEM, with nearly 4% gain versus about 3% gain in MAP.

Question: what causes heart disease?
Document:
[1] Cardiovascular disease (also called heart disease) is a class of diseases that involve the heart or blood vessels (arteries, capillaries, and veins).
[2] .....
[3] The causes of cardiovascular disease are diverse but atherosclerosis and hypertension are the most common.
[4] .....
Reference Answer:
The causes of cardiovascular disease are diverse but atherosclerosis and hypertension are the most common.

Table 6: The error prediction of our proposed model. The text is shown in its original form, which may contain errors in typing. Our proposed model predict the first sentence is the right answer, however it is wrong.

Model	MAP	MRR
ESIM (Lan and Xu, 2018)	65.20	66.40
KEM (Chen et al., 2018)	68.03	69.58
WEHM (without knowledge)	73.17	74.63
WEHM (Proposed)	**77.02**	**78.82**

Table 7: Experimental results on the WikiQA dataset. We list the reported results of ESIM in the paper (Lan and Xu, 2018), and re-run the public code of KEM proposed in the paper (Chen et al., 2018) to produce its results.

4 Related Work

In the NLP field, many problems involve matching two or more sequences to make a decision. For the DQA task, most of the studies also consider this problem as text matching, and they compute the semantic similarity between the question and candidate answers to decide whether a sentence in the document could answer the question.

There have been various deep neural network models proposed to tackle sentence pairs matching. Two kinds of matching strategies have been considered: the first is to convert the whole source and target sentences into embedding vectors in the latent spaces respectively, and then calculate the similarity score between them; the second is to calculate the similarities among all possible local positions of the source and target sentences and then summarize the local scores into the final similarity value. As for works using the first strategy, Qiu and Huang (2015) apply a tensor transformation layer on CNN-based embeddings to capture the interactions between the question and answer. Tan et al. (2015) employ the long short-term memory (LSTM) network to address this problem. In the second strategy, Pang et al. (2016) build hierarchical convolution layers on the word similarity matrix between sentences, and Yin and Schütze (2015) propose MultiGranCNN to integrate multiple granularity levels of matching models.

For the DQA task, the notable work is the compare-aggregate structure, which is first proposed by Wang and Jiang (2016). Following this structure, Bian et al. (2017) propose the dynamic-clip way to compute the attention score. Our basic model also adopts this structure, but with a different implementation. What's more, we employ a hierarchical module to capture inter-sentence relations.

Exploiting the background knowledge and common sense to improve NLP tasks' performance has long been a heated research topic. To facilitate NLP tasks, various semantic knowledge bases (KBs) have been constructed, ranging from manually annotated semantic networks like WordNet (Fellbaum, 1998) to semi-automatically or automatically constructed knowledge graphs like Freebase (Bollacker et al., 2008). Recently, several approaches have been proposed to leverage the prior knowledge in neural networks on different tasks (Yang and Mitchell, 2017; Chen et al., 2018; Wu et al., 2018; Wang et al., 2019). Wu et al. (2018) fuse the prior knowledge into word representations with a knowledge gate by using question categories for the QA task and topics for the conversation task. Yang and Mitchell (2017) propose a KBLSTM network architecture, which incorporates the background knowledge into LSTM to improve machine reading. Unlike the two approaches, our model directly employs the synset and hypernym concepts information to enrich the word representation. Chen et al. (2018) use WordNet to measure the semantic relatedness of word pairs for the natural language inference task, including synonym, antonym, hypernym, and same hypernym. Each of these features is denoted as a real number and is incorporated into the neural networks. Compared to the feature vectors derived from the WordNet, our model directly employ the synset and hypernym relation scores to enrich the attention mechanism. Wang et al. (2019) present an entailment model for solving the Natural Language Inference (NLI) problem that utilizes ConceptNet as an external knowledge source, while our method mainly focus on the WordNet.

5 Conclusion

In this paper, we exploit a WordNet-enhanced hierarchical model to address the answer selection problem. Based on WordNet's prior knowledge, the proposed model applies the synset and hypernym concepts to enrich word representations and uses synset and hypernym relation scores between two concepts to enhance the traditional attention score. Extensive experiments conducted on two benchmark datasets demonstrate that our method significantly improves the baseline model and outperforms state-of-the-art results by a large margin. Our approach obtains 1.62% improvement and 2.57% improvement in MAP on the WikiQA and SelQA datasets, respectively, compared to the state-of-the-art results. In the future, we would like to

explore more knowledge in the neural networks to deal with different NLP tasks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61773026) and the Key Project of Natural Science Foundation of China (61936012).

References

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 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*, pages 1987–1990. ACM.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2406–2417.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Hua He and Jimmy J Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *HLT-NAACL*, pages 937–948.

Daniel Hewlett, Llion Jones, Alexandre Lacoste, et al. 2017. Accurate supervised and semi-supervised machine reading for long documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2010.

Tomasz Jurczyk, Michael Zhai, and Jinho D Choi. 2016. Selqa: A new benchmark for selection-based question answering. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 820–827. IEEE.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. *arXiv preprint arXiv:1806.04330*.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *AAAI*, pages 2793–2799.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics.

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

Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, pages 1305–1311.

Cicero Nogueira dos Santos, Kahini Wadhawan, and Bowen Zhou. 2017. Learning loss functions for semi-supervised learning via discriminative adversarial networks. arxiv preprint. *arXiv preprint arXiv:1707.02198*.

Dinghan Shen, Martin Renqiang Min, Yitong Li, and Lawrence Carin. 2017a. Adaptive convolutional filter generation for natural language understanding. *arXiv preprint arXiv:1709.08294*.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017b. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *ACL (1)*.

Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017a. Irgan: A minimax game for unifying generative and discriminative information retrieval models. *arXiv preprint arXiv:1705.10513*.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017b. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. 2019. Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7208–7215.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural qa as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280.

Yu Wu, Wei Wu, Can Xu, and Zhoujun Li. 2018. Knowledge enhanced hybrid neural network for text matching. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5586–5593.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1436–1446.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 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)*, volume 1, pages 1744–1753.

Wenpeng Yin and Hinrich Schütze. 2015. Multi-grancnn: An architecture for general matching of text chunks on multiple levels of granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 63–73.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.