Combing Context and Commonsense Knowledge Through Neural Networks for Solving Winograd Schema Problems

Quan Liu[†], Hui Jiang[‡], Zhen-Hua Ling[†], Xiaodan Zhu[ℓ], Si Wei[§], Yu Hu[†§]

[†] National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, China

[‡] Department of Electrical Engineering and Computer Science, York University, Canada

[ℓ] National Research Council Canada, Ottawa, Canada

§ iFLYTEK Research, Hefei, China

emails: quanliu@mail.ustc.edu.cn, hj@cse.yorku.ca, zhling@ustc.edu.cn, zhu2048@gmail.com
siwei@iflytek.com, yuhu@iflytek.com

Abstract

This paper proposes a general framework to combine context and commonsense knowledge for solving the Winograd Schema (WS) and Pronoun Disambiguation Problems (PDP). In the proposed framework, commonsense knowledge bases (e.g. cause-effect word pairs) are quantized as knowledge constraints. The constraints guide us to learn knowledge enhanced embeddings (KEE) from large text corpus. Based on the pre-trained KEE models, this paper proposes two methods to solve the WS and PDP problems. The first method is an unsupervised method, which represents all the pronouns and candidate mentions in continuous vector spaces based on their contexts and calculates the semantic similarities between all the possible word pairs. The pronoun disambiguation procedure could then be implemented by comparing the *semantic similarities* between the pronoun (to be resolved) and all the candidate mentions. The second method is a supervised method, which extracts features for all the pronouns and candidate mentions and solves the WS problems by training a typical mention pair classification model. Similar to the first method, the features used in the second method are also extracted based on the KEE models. Experiments conducted on the available PDP and WS test sets show that, these two methods both achieve consistent improvements over the baseline systems. The best performance reaches 62% in accuracy on the PDP test set of the first Winograd Schema Challenge.

Introduction

In recent years, many AI challenges or competitions have been proposed to help evaluate the cognitive levels of machines (Levesque, Davis, and Morgenstern 2011; Weston et al. 2015; Clark 2015). Among those challenges, the Winograd Schema Challenge (WSC) has been proposed as an alternative to the Turing Test (Levesque, Davis, and Morgenstern 2011). Turing first introduced the notion of testing a computer system's intelligence by assessing whether it could make a human judge think that she was conversing with a human rather a computer (Turing 1950). However, some recent efforts have merely on engaged surface-level conversation tricks to fool humans who do not delve deeply enough into a conversation, and make them think they are speaking to another human being (Veselov, Demchenko, and Ulasen ; Warwick and Shah 2014). To fix this issue, WSC is claimed to be a more suitable task which does not rely on human's subjective assessment. A Winograd schema (WS) question is a pair of sentences that differ only in one or two words which results in a different resolution of coreference. A well-known example is "The city council refused the demonstrators a permit because they feared violence," where "they" refers to the council. But if one changes the verb "feared" to "advocated", the computer needs to know that "they" refers to "demonstrator", if it possesses real intelligence. To solve the problem, common sense knowledge is essential. Towards solving the final solution for WS problem, a similar test set, called pronoun disambiguation problem (PDP) is designed as the first round of the Winograd Schema Challenge (Morgenstern, Davis, and Ortiz Jr 2016). A typical example is "Mrs. March gave the mother tea and gruel, while she dressed the little baby as tenderly as if it had been her own." One way to reason that she in *she dressed* refers to Mrs. March and not the mother, is to realize that the phrase "as if it were her own" implies that it (the baby) is not actually her own; that is, she is not the mother and must, by process of elimination, be Mrs. March. Similar to the Winograd schemas, a substantial amount of commonsense knowledge appears to be necessary to disambiguate pronouns.

This paper proposes neural network models to combine context and commonsense knowledge to solve both the WS and PDP problem. The main ideas are two-fold: 1) The first is to leverage *context* effectively. We believe that since context is a key information for learning word meanings, it should be useful for disambiguating the Winograd schema problems. In general, modeling context and learning word meanings could be very efficient through unsupervised learning that leverages large amounts of free texts. 2) In the common sense respect, we describe a simple but effective method utilizing commonsense knowledge. To jointly consider those two aspects, this paper proposes to combine context and commonsense knowledge through neural networks through a knowledge enhanced embeddings (KEE) framework. This paper further proposes two methods: 1) The first is the an unsupervised semantic similarity method (USSM), which represents all the pronouns and candidate mentions by composing their contexts based on the pre-trained knowledge enhanced embeddings. We then calculate the semantic similarities between the embedding vectors of the pronoun under concern and all candidate mentions. The candidate with largest semantic similarity with respect to the

pronoun will be predicted as the answer. 2) The second method is a neural knowledge activated method (NKAM), which extracts features based on the KEE models and trains a mention pair classifier with neural networks. For the experiment section, this paper conducts experiments on the official datasets of the first WSC challenge, including the PDP test set provided by `commonsensereasoning.org`, as well as a set of Winograd schemas manually created by (Levesque, Davis, and Morgenstern 2011). Experimental results all indicate that the proposed KEE method (combing context and knowledge) performs better comparing with the baseline models.

The remainder of the paper will start with introducing the main motivation. After that, we introduce the main methods proposed to solve the WS and PDP problems. We then present all the experiments, including setup, datasets, and results, before we conclude this work.

Motivation

In this section, we introduce the main motivation of this work. We will firstly present the main problems we aim to solve, i.e., the Winograd schema (WS) and the Pronoun Disambiguation Problem (PDP). After that, detailed descriptions would be given to illustrate our motivation.

Winograd Schema (WS)

The Winograd schema (WS) evaluates a system's commonsense reasoning ability based on a traditional, very specific natural language processing task: coreference resolution (Saba 2015). The WS problems are carefully designed to be a task that cannot be easily solved without commonsense knowledge. In fact, even the solution of traditional coreference resolution problems rely on semantics or world knowledge (Strube 2016). As described in (Levesque, Davis, and Morgenstern 2011), a WS is a small reading comprehension test involving a single binary question.

- ***Joan*** *made sure to thank* ***Susan*** *for all the help she had given. Who had given the help?*
  - *Answer A: Joan*
  - *Answer B: Susan*
  - *Correct Answer: B*

The correct answers to the above question are obvious for human beings. In each of the questions, the corresponding WS has the following four features:

1. Two parties are mentioned in a sentence by noun phrases. They can be two males, two females, two inanimate objects or two groups of people or objects.
2. A pronoun or possessive adjective is used in the sentence in reference to one of the parties, but is also of the right sort for the second party.
3. The question involves determining the referent of the pronoun or possessive adjective. Answer A is always the first party mentioned in the sentence (but repeated from the sentence for clarity), and Answer B is the second party.
4. There is a word (called the *special* word) that appears in the sentence and possibly the question. When it is replaced by another word (called the *alternate* word), everything still makes perfect sense, but the answer changes.

In this example, if we change the word *given* to *received*, the answer changes to *A* (i.e., *Joan*) since in our commonsense knowledge, we think that a person who receives help should makes sure to thank the person who provides help to him.

Pronoun Disambiguation Problems (PDP)

The pronoun disambiguation problems (PDP) are complex coreference resolution problems, which are taken directly or modified from examples found in literature, biographies, autobiographies, essays, news analyses, and news stories; they may need some manual processing (Morgenstern, Davis, and Ortiz Jr 2016). Here is one typical PDP example:

- ***The Dakota prairie*** *lay so warm and bright under* ***the shining sun*** *that it did not seem possible that* ***it*** *had ever been swept by the winds and snows of that hard winter.*
  - *Snippet:* ***it*** *had ever been swept*
  - *Answer A: the prairie*
  - *Answer B: the sun*
  - *Correct Answer: A*

In the PDP problem, the pronoun to be resolved is highlighted in bold. It is repeated again, with a snippet of context, and with several candidate answers, in the line following the passage. In the example shown here, we know that the *prairie* (rather than the *sun*) would be more likely to be *swept* by the winds. A difference between PDP and WS problems is that, the number of candidate noun phrases in each PDP problem would not always be two, but can be three, four, or even more. Therefore, the random-guess accuracy in the PDP problems will be less than 50% while the accuracy of a random-guess for WS is 50%.

Motivation: Knowledge Enhanced Embeddings

Solving WS or PDP problems is not easy since it requires commonsense knowledge. In the paper, we propose to combine ***context*** and commonsense ***knowledge*** through neural networks for solving both problems. The main motivation is described as follows. First, since context is key for learning word meaning (Harris 1954; Miller and Charles 1991), we represent the pronoun and all the candidate mentions by their contexts. For instance, in the aforementioned PDP example, we represent the word *prairie* by *The Dakota* and *lay so warm and bright*. Meanwhile, the word *sun* could be represented by *and bright under the shining* and *that it did not seem*. Based on this, the pronoun disambiguation problems is solved by calculating the semantic similarities between the representations of the pronoun and all the corresponding candidates. Second, relying only on context is not good enough for tackling the WS and PDP problems. It is essential to find an effective strategy to combine context and commonsense knowledge. Therefore, in this paper, we propose a knowledge enhanced embedding (KEE) model. As shown in Figure 1, the KEE model combines the context and commonsense knowledge, in the *feature level*. By training KEE

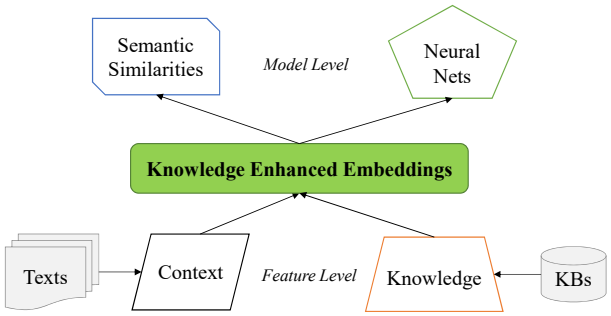

Figure 1: Based on Knowledge Enhanced Embeddings, two levels considered to be useful for solving WS and PDP.

model using large text corpora and commonsense KBs, we obtain useful distributed word representations. Based on the pre-trained word representations, we propose two effective methods in the *model level*, for finally solve the WS and PDP problems. The proposed two methods are semantic similarity method and neural network method. In the semantic similarity method, we represent all the pronouns and candidate mentions by composing their contexts from words. By further calculating the semantic similarities between the representations of each pronoun and the corresponding candidate mentions, the procedure to answer PDP or WS questions could then be implemented by finding the most similar candidate for the pronoun. In the neural network methods, the pre-trained KEE models are also used for extracting embedding features for all the pronouns and candidates. However, we do not calculate the semantic similarities. On the contrary, we train a typical neural mention pair classifier with supervised coreference training dataset. No matter how large the differences between these two methods, both of them are all implemented based on the KEE model.

The Proposed Methods

Based on the motivation, we introduce the KEE model and two methods to solve WS and PDP problems. Before introducing those methods, we describe the commonsense knowledge used in this paper.

The Commonsense Knowledge

There have been open commonsense knowledge bases in the artificial intelligence community, e.g. Cyc (Lenat 1995), ThoughtTreasure (Mueller 1998) and ConcepNet (Liu and Singh 2004). Cyc is an artificial intelligence project that attempts to assemble a comprehensive ontology and knowledge base of everyday common sense knowledge, with the goal to enable AI applications to perform human-like reasoning. Typical pieces of knowledge represented in the Cyc database are "every tree is a plant" and "plants die eventually". ConceptNet is a semantic network containing lots of things computers should know about the world, especially when understanding text written by people. It is built from *nodes* representing words or short phrases of natural language, and labeled *relationships* between them. For example, the triple (*learn*, *MotivatedByGoal*, *knowledge*) indicates that "we would learn because we want knowledge". Those existing commonsense KBs are well constructed; however, in this paper, we aim to find commonsense knowledge for solving the WS and PDP problems by the following requirements: 1) to avoid data sparseness problem, the vocabulary of the KB covers common words (not phrases) in daily life, e.g., common verbs, adjectives, etc. 2) The commonsense relationships between the nodes in the vocabulary cover common relations, e.g. cause-effect, entailment, etc.

Based on these two requirements, this paper proposes to use the KB constructed by a recent work (Liu et al. 2016), which collects word pairs with cause-effect relationships automatically. Figure 2 shows the typical formula of the corresponding KB.

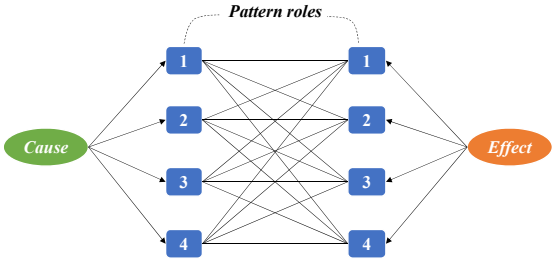

Figure 2: The typical formula of the commonsense KB: automatically constructed cause-effect word pairs.

The KB contains a large number of cause-effect word and phrase pairs constructed from large text corpora. The vocabulary covered by the KB contains thousands of common verbs and adjectives. At shown in Figure 2, there are four pattern roles for both the *cause* and *effect* phrases. The four roles include (active,positive), (active,negative), (passive,positive) and (passive,negative). Table 1 shows some examples. In this paper, all the word pairs (rather than phrase pairs) of this KB are used.

No	Pairs	Meaning
1	(win, happy)	sb. *win* → *happy*.
2	(rob, be arrested)	sb. *rob* → *be arrested*.
3	(confident, not afraid)	sb. *confident* → *not afraid*
4	(be restricted, unable)	sb. *be restricted* → *unable*

Table 1: Typical examples of the cause-effect pairs.

Knowledge Enhanced Embedding

To combine context and commonsense knowledge for solving the WS and PDP problems, this paper proposes to treat the commonsense knowledge as semantic constraints and learn knowledge enhanced embedding (KEE) based on the generated constraints. The idea to learn word embedding based on constraints is similar to the work of (Liu et al. 2015). The main difference is the way we generate the knowledge constraints. In this paper, we propose to create constraints as follows:

Knowledge constraints Since all the cause-effect pairs used in this paper contain the corresponding confidence weights, i.e. PMI values, we propose to generate semantic inequalities by randomly sampling two cause-effect pairs. More specifically, for each cause-effect pair, we will randomly sample 5 different pairs from the whole KB set and construct the inequalities by comparing their PMI values respectively. For instance, once we generate cause-effect pair (w_i, w_j) and (w_k, w_g) and the PMI value of pair (w_i, w_j) is larger than pair (w_k, w_g), we can make the inequality as:

$$\text{sim}(w_i, w_j) > \text{sim}(w_k, w_g) \tag{1}$$

The idea to generate such inequality is similar to the physical meaning of lexical entailment (Geffet and Dagan 2005; Turney and Mohammad 2015). This paper assumes if a word tends to be the effect of another word, they should have similar context patterns. Note that the commonsense knowledge base used in this paper covers all the common verbs and adjectives, and the knowledge constraints would not influence the learning for the remaining words in the whole large vocabulary. This is important because the used verbs and adjectives play a central role in commonsense reasoning. Currently, incorporating more knowledge of other types of words, e.g., nouns, adverbs and prepositions is beyond our concern.

The main framework The main framework for learning knowledge enhanced embedding is shown in Figure 3.

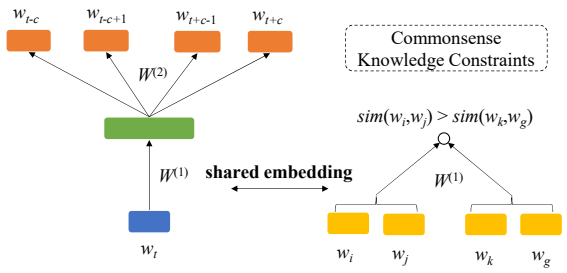

Figure 3: The framework for knowledge enhanced embeddings (KEE). The training process combines the text corpus and commonsense knowledge.

The left part in this framework is the typical skip-gram model, which learns continuous word vectors from text corpora based on the aforementioned distributional hypothesis (Mikolov et al. 2013). Each word in vocabulary (size of V) is mapped to a continuous embedding space by looking up an embedding matrix $\mathbf{W}^{(1)}$. And $\mathbf{W}^{(1)}$ is learned by maximizing the prediction probability, calculated by another prediction matrix $\mathbf{W}^{(2)}$, of its neighbouring words within a context window. Given a sequence of training data, denoted as $w_1, w_2, w_3, ..., w_T$ with T words, the skip-gram model aims to maximize the objective function:

$$\mathcal{Q} = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t) \tag{2}$$

where c is the size of context windows, w_t denotes the input central word and w_{t+j} for its neighbouring word. The skip-gram model computes the above conditional probability $p(w_{t+j}|w_t)$ using the following softmax function:

$$p(w_{t+j}|w_t) = \frac{\exp(\mathbf{w}_{t+j}^{(2)} \cdot \mathbf{w}_t^{(1)})}{\sum_{k=1}^{V} \exp(\mathbf{w}_k^{(2)} \cdot \mathbf{w}_t^{(1)})} \tag{3}$$

where $\mathbf{w}_t^{(1)}$ and $\mathbf{w}_k^{(2)}$ denotes row vectors in matrices $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$, corresponding to word w_t and w_k respectively.

In this paper, we proposed to incorporate the commonsense knowledge as constraints into the word embedding training process. Assume the knowledge is represented by a large number of inequalities, denoted as the set S. This paper denotes $s_{ij} = \text{sim}(\mathbf{w}_i^{(1)}, \mathbf{w}_j^{(1)})$ as the semantic similarity hereafter. The final objective function becomes:

$$\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\} = \arg\max_{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}} \mathcal{Q}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}) \tag{4}$$

subject to

$$s_{ij} > s_{kg} \quad \forall (i, j, k, g) \in S. \tag{5}$$

In this work, we formulate the above constrained optimization problem into an unconstrained one by casting all the constraints as a penalty term in the objective function:

$$\begin{aligned} \mathcal{Q}' &= \mathcal{Q} - \beta \cdot \mathcal{D} \\ \mathcal{D} &= \sum_{(i,j,k,g)\in S} f(i, j, k, g) \end{aligned} \tag{6}$$

where β is a control parameter to balance the contribution of the penalty term in the optimization process. The function $f(\cdot)$ is a normalization function. This paper uses a hinge loss function like $f(i, j, k, g) = h(s_{kg} - s_{ij})$ where $h(x) = \max(0, x)$.

The objective function in eq. (6) could be optimized using the standard stochastic gradient descent (SGD) algorithm. Generally, as shown in Figure 4, the work to combine context and commonsense knowledge is implemented in the proposed KEE model. After that, embeddings trained by the KEE would serve for the two methods (USSM and NKAM) designed to solve the WS and PDP problems in this paper.

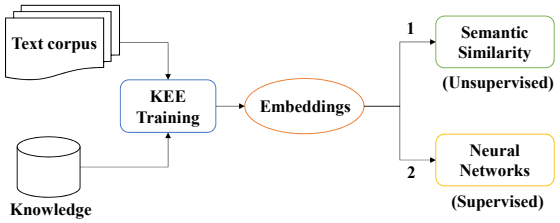

Figure 4: The main methods used in this paper.

Unsupervised Semantic Similarity Method

The first method proposed in this paper for answering the WS and PDP problems, shown in Figure 5, is an **unsupervised** method. We introduce a straightforward unsupervised semantic similarity method (USSM), which aims to

represent the pronoun and all the candidate mentions by composing from the pre-trained KEE embeddings. For the composition function, we design to use the recently proposed approach, named fixed-size ordinally-forgetting encoding (FOFE) (Zhang et al. 2015).

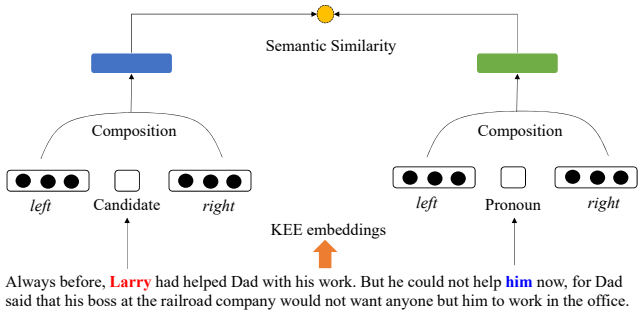

Figure 5: The unsupervised semantic similarity method.

For one sentence, the function FOFE works as follows. Given a sequence of words, $S = \{w_1, w_2, ..., w_T\}$, each word w_t is first represented by a 1-of-K representation $\mathbf{e}_t$, from the first word $t = 1$ to the end of the sequence $t = T$, FOFE encodes each partial sequence (history) based on a simple recursive formula (with $\mathbf{z}_0 = \mathbf{0}$) as:

$$\mathbf{z}_t = \alpha \cdot \mathbf{z}_{t-1} + \mathbf{e}_t, \ (1 \leq t \leq T) \quad (7)$$

where $\mathbf{z}_t$ denotes the FOFE code for the partial sequence up to w_t, and $\alpha, (0 < \alpha < 1)$ is a constant forgetting factor to control the influence of the history on the current position. Assume we have three symbols in vocabulary, e.g., A, B, C, whose 1-of-K codes are $[1, 0, 0]$, $[0, 1, 0]$ and $[0, 0, 1]$ respectively. In this case, the FOFE code for the sequence $\{ABC\}$ is $[\alpha^2, \alpha, 1]$, and that of $\{ABCBC\}$ is $[\alpha^4, \alpha + \alpha^3, 1 + \alpha^2]$. In this paper, we first use the FOFE method to encode the context of each word into a fixed-size code (of the vocabulary size). Then, we use the embedding matrix $W^{(1)}$, learned by KEE as above, to project into a low-dimension space. These low-dimension vectors are used to calculate cosine distances to select the answer from the candidates.

Neural Knowledge Activated Method

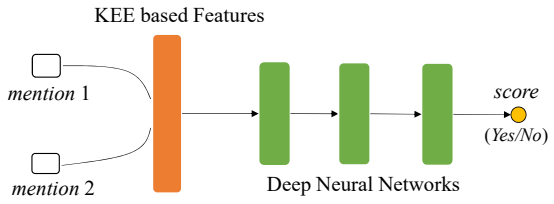

Figure 6: The neural knowledge activated method.

As shown in Figure 6, the second method proposed in this paper is an **supervised** method. The difference from the first semantic similarity method is that, it does not simply calculate the semantic similarities between the extracted embedding vectors of the pronoun (to be resolved) and all the candidate mentions, but instead uses the composed embedding vectors as input features and train a deep neural networks (DNN). The DNN model works as a mention pair classifier for judging whether two mentions are coreferent or not, which is a widely used technology in the coreference resolution community (Ng 2010). Since the features we extracted for training the DNN are composed from the knowledge enhanced embeddings, we call the method as neural knowledge activated method (NKAM) hereafter.

Experiments

In this section, we present all the experiments conducted to evaluate the effectiveness of the proposed methods. This section would be started by introducing the experimental datasets and experimental setups. After that, experimental results and analysis would be given correspondingly.

Datasets

For evaluating the effectiveness of the proposed methods and keep our methods comparable, all the experimental datasets investigated in this paper, including PDP test set[1] and WS test set[2], are from the Winograd Schema Challenge (Morgenstern, Davis, and Ortiz Jr 2016). Table 2 lists the sizes and the random-guess accuracies of all the datasets. The PDP test set contains 60 problems while the WS test set has 273 testing problems.

Dataset	Dataset size	Random-guess acc (%)
PDP test set	60	45.0
WS test set	273	50.0

Table 2: Experimental datasets used in this paper.

Experimental setup

To make clear all the settings for the proposed methods of this work, we describe the experimental setup as follows.

Setup for knowledge enhanced embeddings This paper uses the English Wikipedia corpora to train the knowledge enhanced embeddings. Particularly, we utilize two Wikipedia corpora with different sizes. The first corpus of a smaller size is a data set containing the first one billion characters from Wikipedia[3], named as *Wiki-Small* in our experiments. The second corpus of a relatively large size is a snapshot of the Wikipedia articles from (Shaoul 2010), named as *Wiki-Large* in our experiments. Both Wikipedia corpora have been pre-processed by removing all the HTML metadata and hyper-links and replacing the digit numbers with English words. After text normalization, the *Wiki-Small* corpus contains totally 130 million words, for which we create a lexicon of 225,909 distinct words appearing more than 5 times in the corpus. Similarly, the *Wiki-Large* corpus contains about 1 billion words, for which we create a lexicon of

[1]Available at `www.cs.nyu.edu/faculty/davise/papers/WinogradSchemas/PDPChallenge2016.xml`.

[2]Available at `www.cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml`.

[3]`http://mattmahoney.net/dc/enwik9.zip`.

Text Corpus	Answering Methods	KEE Settings	Answering Accuracy (%)	
			PDP test (size: 60)	WS test (size: 273)
Wiki-Small	USSM	Context	41.7	48.7
		Context + Knowledge	48.3 (+15.8)	50.9 (+4.5)
	NKAM	Context	46.7	49.1
		Context + Knowledge	53.3 (+14.1)	51.7 (+5.3)
	USSM + NKAM	Context	50.0	50.2
		Context + Knowledge	**58.3 (+16.6)**	**52.4 (+4.4)**
Wiki-Large	USSM	Context	48.3	49.8
		Context + Knowledge	55.0 (+13.8)	52.0 (+4.4)
	NKAM	Context	51.7	50.5
		Context + Knowledge	56.7 (+9.6)	52.4 (+3.8)
	USSM + NKAM	Context	53.3	50.6
		Context + Knowledge	**61.7 (+15.7)**	**52.8 (+4.3)**

Table 3: Overall experimental results. The random-guess accuracies for PDP test and WS test set are 45%, 50% respectively.

235,167 words, each appearing more than 60 times. In all the experiments of this paper, the settings for KEE are the same. The embedding dimension is set to be 100 while the context window size c in eq. (2) is set to be 5. The combination coefficient β in eq. (6) is set to be 0.01. The KEE models are trained by the stochastic gradient descents (SGD) algorithm. The initial learning rate is set as 0.025 and the learning rate is decreased linearly during the SGD model training process.

Setup for USSM and NKAM As for feature extraction in the USSM or NKAM methods, for both pronouns and candidate mentions, the context we utilize for feature extraction is the entire sentence. Meanwhile, the weight α in eq. (7) is set to 0.7 for context composition. In the USSM method, we use the popular Cosine similarity to evaluate the semantic similarity between any two mentions. On the other hand, for the NKAM method, this paper uses the popular coreference resolution datasets, i.e., OntoNotes (Weischedel et al. 2013), to extract labelled mention pairs for model training. Considering the sentence length in the WS or PDP questions is usually less than 3, in this paper, we extract all the labelled mention pairs for pronouns within three adjacent sentences. We finally extract 306,903 training mention pairs. Meanwhile, the corresponding neural network has 1 hidden layer with 300 units. The non-linear activation function is rectified linear unit (ReLU) (Nair and Hinton 2010).

Results

Table 3 shows the overall results. We divide the results by the text corpus we used for KEE training. In addition to experimenting the proposed two methods, i.e., USSM and NKAM, we also construct a system by combing the USSM and NKAM methods. For each pronoun and its candidate mentions, the system combination procedure is implemented by interpolating the scores calculated by the USSM and NKAM method (the interpolation coefficient is 70% for NKAM and 30% for USSM). From the results, we find the USSM method (as the unsupervised method) achieves a 41.7% and 48.3% accuracy on the PDP test set when the KEE models are only trained on texts (no commonsense knowledge combined, which is equal to the skip-gram models) from *Wiki-Small* and *Wiki-Large*. When we use the pre-trained knowledge enhanced embeddings (context+knowledge), the corresponding performance improves to 48.3% and 55.0%. Similar performances are achieved in NKAM and the combined system as well. In the combined system, we obtain a 61.7% accuracy on the PDP test set, which is significant better than the system (53.3%) constructed solely based on the context (without combining with commonsense knowledge). Meanwhile, since the WS test set is carefully designed by human beings, the performance of the baseline systems, i.e., all the results shown in the rows of "KEE settings = Context", are poor. The best performance is 50.6% when using *Wiki-Large* as text corpus, in the combined system. This suggests that only relying on context is clearly not good enough. After applying KEE to the USSM and NKAM methods, we achieve a 52.8% accuracy on the whole Winograd schemas test set, which is 4.4% better than the corresponding system.

Conclusions

This paper proposes a general knowledge enhanced embedding (KEE) framework to combine context and commonsense knowledge for solving the Winograd Schema (WS) and Pronoun Disambiguation Problems (PDP). The KEE is a flexible framework to learn distributed representations under the supervision of commonsense knowledge from large text corpus. Using the pre-trained KEE representations, we further proposes two methods, i.e. unsupervised semantic similarity method and neural knowledge activated method, to solve the WS and PDP problems. Experiments conducted on the official datasets show that these two methods achieve consistent improvements over the baseline systems. Furthermore, investigations conducted in this paper also provide some insights on the Winograd Schema Challenge.

References

[Clark 2015] Clark, P. 2015. Elementary school science and math tests as a driver for ai: take the Aristo challenge! In *AAAI*, 4019–4021.

[Geffet and Dagan 2005] Geffet, M., and Dagan, I. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of ACL*, 107–114. Association for Computational Linguistics.

[Harris 1954] Harris, Z. S. 1954. Distributional structure. *Word* 10(2-3):146–162.

[Lenat 1995] Lenat, D. B. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

[Levesque, Davis, and Morgenstern 2011] Levesque, H. J.; Davis, E.; and Morgenstern, L. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, 47.

[Liu and Singh 2004] Liu, H., and Singh, P. 2004. Conceptnetła practical commonsense reasoning tool-kit. *BT technology journal* 22(4):211–226.

[Liu et al. 2015] Liu, Q.; Jiang, H.; Wei, S.; Ling, Z.-H.; and Hu, Y. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of ACL*, 1501–1511.

[Liu et al. 2016] Liu, Q.; Jiang, H.; Evdokimov, A.; Ling, Z.-H.; Zhu, X.; Wei, S.; and Hu, Y. 2016. Probabilistic reasoning via deep learning: Neural association models. *arXiv preprint arXiv:1603.07704*.

[Mikolov et al. 2013] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Miller and Charles 1991] Miller, G. A., and Charles, W. G. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes* 6(1):1–28.

[Morgenstern, Davis, and Ortiz Jr 2016] Morgenstern, L.; Davis, E.; and Ortiz Jr, C. L. 2016. Planning, executing, and evaluating the winograd schema challenge. *AI Magazine* 37(1):50–54.

[Mueller 1998] Mueller, E. T. 1998. *Natural language processing with Thought Treasure*. Signiform New York.

[Nair and Hinton 2010] Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 807–814.

[Ng 2010] Ng, V. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, 1396–1411. Association for Computational Linguistics.

[Saba 2015] Saba, W. 2015. On the winograd schema challenge.

[Shaoul 2010] Shaoul, C. 2010. The westbury lab wikipedia corpus. *Edmonton, AB: University of Alberta*.

[Strube 2016] Strube, M. 2016. The (non) utility of semantics for coreference resolution (corbon remix). In *NAACL 2016 workshop on Coreference Resolution Beyond OntoNotes*.

[Turing 1950] Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59(236):433–460.

[Turney and Mohammad 2015] Turney, P. D., and Mohammad, S. M. 2015. Experiments with three approaches to recognizing lexical entailment. *Natural Language Engineering* 21(03):437–476.

[Veselov, Demchenko, and Ulasen] Veselov, V.; Demchenko, E.; and Ulasen, S. Eugene goostman (2014).

[Warwick and Shah 2014] Warwick, K., and Shah, H. 2014. Good machine performance in turing's imitation game. *IEEE Transactions on Computational Intelligence and AI in Games* 6(3):289–299.

[Weischedel et al. 2013] Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; Franchini, M.; et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.

[Weston et al. 2015] Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

[Zhang et al. 2015] Zhang, S.; Jiang, H.; Xu, M.; Hou, J.; and Dai, L. 2015. The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of ACL*.