Cause-Effect Knowledge Acquisition and Neural Association Model for Solving A Set of Winograd Schema Problems

Quan Liu[†], Hui Jiang[‡], Andrew Evdokimov[‡], Zhen-Hua Ling[†], Xiaodan Zhu[‡], Si Wei[§], Yu Hu[†§]
[†] NELSLIP, Dept. of EEIS, University of Science and Technology of China
[‡] Department of Electrical Engineering and Computer Science, York University, Canada
[§] iFLYTEK Research, Hefei, China
quanliu@mail.ustc.edu.cn, hj@cse.yorku.ca, ae2718@cse.yorku.ca, zhling@ustc.edu.cn
zhu2048@gmail.com, siwei@iflytek.com, yuhu@iflytek.com

Abstract

This paper focuses on the investigations in Winograd Schema (WS), a challenging problem which has been proposed for measuring progress in commonsense reasoning. Due to the lack of commonsense knowledge and training data, very little work has been reported on the WS problems. This paper addresses a set of WS problems by proposing a knowledge acquisition method and a general neural association model. To avoid the sparseness issue, the knowledge we aim to collect is the cause-effect relationships between a collection of commonly used words. The knowledge acquisition method supports us to extract hundreds of thousands of cause-effect pairs from text corpora automatically. Meanwhile, a neural association model (NAM) is proposed to encode the association relationships between any two discrete events. Based on the extracted knowledge and the NAM models, we successfully build a system for solving a causal subset of WS problems from scratch and achieve 70% accuracy. Most importantly, this paper provides a flexible framework to solve WS problems based on event association and neural network methods.

1 Introduction

In recent years, the rapid developments of machine learning, especially the deep learning and reinforcement learning techniques, have brought significant improvements in many research areas [Michalski *et al.*, 2013; LeCun *et al.*, 2015; Mnih *et al.*, 2015]. However, despite the successes in various artificial intelligence (AI) applications, the research filed of commonsense reasoning, a typical AI-complete problem, still remains to be carefully investigated [Mueller, 2014; Davis and Marcus, 2015]. In this paper, we pay our attentions to a recently proposed AI task, i.e. Winograd Schema (WS) [Levesque *et al.*, 2011]. WS is a task designed for measuring progress in commonsense reasoning, which has been suggested as an alternative to the Turing Test [Morgenstern and Ortiz Jr, 2015; Marcus *et al.*, 2016]. The core of the WS task is to answer some manually designed coreference resolution problems. A typical WS example was proposed in [Winograd, 1972], which defined a scenario "The city councilmen refused the demonstrators a permit because they feared violence." and a corresponding question "Who feared violence?". Based on the commonsense, the answer is obvious, i.e., the city councilmen since they tended to fear violence. Meanwhile, if we change verb "feared" in this scenario to "advocated", the answer changes correspondingly.

Although human beings could answer WS problems very easily, however, it is very difficult for machines to do that. To the best of our knowledge, there are few methods that have been proposed to work on the real WS problems[1]. In the works of [Sharma, 2014; Sharma *et al.*, 2015], they identified the knowledge needed to answer a challenge question, hunted down that knowledge from text repositories, and then made logic reasoning with them to output the answer. However, this approach relied on the WS test set to extract corresponding knowledge, which had very poor scalability. In [Schüller, 2014], they proposed to tackle WS problems by formalizing relevance theory in knowledge graphs. Their experiments were performed using Answer Set Programming. However, due to the complexity of the proposed method, they just examined how to answer four WS questions in that work.

In this paper, we work on the real WS task in a typical machine learning paradigm. Solving WS requires commonsense knowledge, however, it is very hard to collect all the commonsense knowledge in our daily life. Moreover, there is no training data provided in the WS task. Therefore, this paper proposes a method to extract commonsense knowledge. We construct a vocabulary with a collection of most commonly used words and phrases. Based on it, the knowledge acquisition method reads a large amount of texts, conducts query searching and dependency parsing, and finally outputs the cause-effect pairs. Meanwhile, this paper proposes a neural association model (NAM) to encode the association relationships between discrete events. In NAM, all symbolic events are represented in vector spaces. Deep neural networks are used to model the association between any two events, taking one event as input to compute a conditional probability of the other event. By combining the extracted knowledge and the NAM model, we successfully build a system to solve WS problems from scratch. Experiments made on a causal

[1] http://www.cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml.

subset of WS problems have proved the effectiveness of this work.

2 The Research Motivation

2.1 Winograd Schemas

Winograd schema evaluates a system's commonsense reasoning ability based on a traditional natural language processing (NLP) task: coreference resolution. All WS problems are carefully designed to be a task that cannot be easily solved without commonsense knowledge. The WS problems could be disambiguated by human readers. Here are some typical WS examples [Levesque *et al.*, 2011].

- **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
- My **meeting** started at 4:00 and I needed to catch the **train** at 4:30, so there wasn't much time. Luckily, it was short, so it worked out. Which was short?
  - Answer A: The meeting
  - Answer B: The train
  - Correct Answer: A

A human who answers these questions correctly typically uses various types of commonsense knowledge, including his abilities in morality, temporal, and his knowledge about meetings, trains, to determine the correct answers. In the first example, if we change verb "given" to "received", the answer changes to Joan. The commonsense is that a person who receives help should thanks the person who provides help to him. Similarly, if we change the word "short" to "delayed" in the second example, the answer changes as well.

An open WS test set is made available to the AI community for research purposes [Davis *et al.*, 2016]. As described in [Morgenstern *et al.*, 2016], creating Winograd schemas is difficult, requiring creativity and inspiration, and too burdensome to do on a yearly or biennial basis. That's why there just have two hundreds of WS test problems in this dataset. Due to the difficulties of solving WS problems, only a small number of approaches have been proposed for the WS task [Schüller, 2014; Bailey *et al.*, 2015; Sharma *et al.*, 2015]. These approaches use logic reasoning methods and just solve quite few WS examples, which are out of the scope of this paper.

There are two similar datasets in the NLP and AI community. The first is a collection of pronoun disambiguation problems (PDP), which is used in the first round of WS challenge [Davis *et al.*, 2016]. The best performances on this dataset is 66.7% [Liu *et al.*, 2016]. The second is a definite pronoun resolution dataset released by [Rahman and Ng, 2012]. Correspondingly, some efforts have been made on this dataset [Kruengkrai *et al.*, 2014; Peng *et al.*, 2015]. However, these two datasets are more like coreference resolution tasks, all the related works tend to use linguistic and coreference resolution methods. Therefore, based on our motivation, we would not use them and only use the real WS dataset.

2.2 Motivation

In this paper, we think some WS problems could be solved by modeling the association relationships between key events. Figure 1 shows a typical example. Given the sentence "The man couldn't lift his son because he was so heavy.", the corresponding question is "Who was heavy?". The answer is the son. This question could be solved by employing the knowledge: "a person who is heavy could not be lifted very easily". If we can model the association between events *heavy* and *not lift*, *not be lifted* correctly, i.e., $\Pr(\text{not lift}|\text{heavy}) < \Pr(\text{not be lifted}|\text{heavy})$, then we can solve this WS problem.

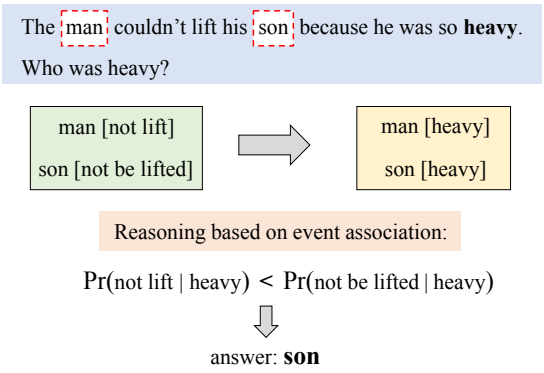

Figure 1: The main motivation of this paper. A typical example to indicate that some WS problems could be solved based on the comparisons between event association probabilities.

Solving WS problems under this motivation is straightforward. We will first extract the key events for all the pronoun and candidates. It is easy to do this work based on dependency parsing results. To focus our attentions on knowledge acquisition and model training, we will manually label all the key events in the WS test set for experiments. The main work of this paper is then straightforward, i.e., to collect commonsense knowledge and design models to solve WS problems. Figure 2 shows the main procedures to build systems to solve WS problems. Ultilzing a set of text corpora, the knowledge acquisition method extracts a so-called *CauseCom* knowledge base. Based on it, a neural association model is used to train models for finally answering WS problems.

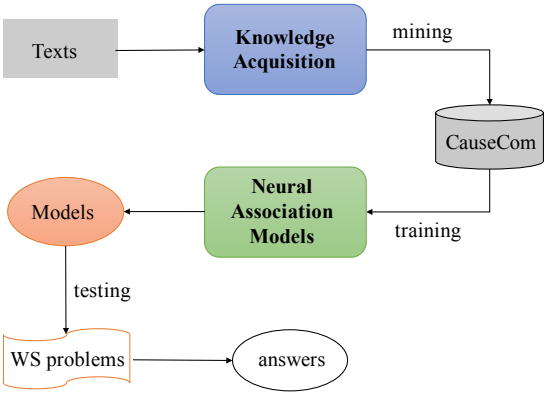

Figure 2: The overall system procedure of this work.

3 Commonsense Knowledge Acquisition

There are some commonsense knowledge bases (KBs) in the AI community, e.g. Cyc [Lenat, 1995] and ConcepNet [Liu and Singh, 2004]. Cyc is a knowledge base contains everyday commonsense knowledge. Typical pieces of knowledge represented in Cyc 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. It is built from *nodes* representing words or short phrases of natural language, and *relationships* are labeled between them. For example, the triple (*learn*, *MotivatedByGoal*, *knowledge*) indicates that "we would learn because we want knowledge".

These commonsense KBs are well constructed, however, the event space they defined is too large. In this paper, we aim to extract commonsense knowledge by restricting the event space. Figure 3 shows the framework of the proposed knowledge acquisition method, which includes vocabulary construction, query searching, dependency parsing and subject-object matching modules.

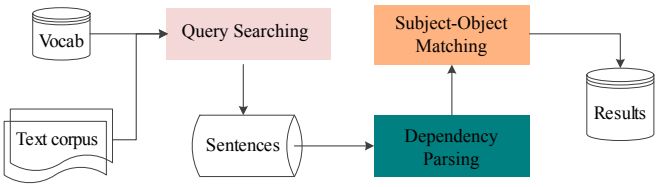

Figure 3: Automatic knowledge acquisition method.

3.1 Vocabulary Construction

This paper constructs a vocabulary with 12,500 most commonly used words and phrases, which contains 5000 verbs, 5000 nouns and 2500 adjectives. The vocabulary construction process starts by conducting part-of-speech tagging on a large Wikipedia corpus. After it, we count the term frequencies for all the words and phrases in different part-of-speech categories. Then, all the corpus based words and phrases would be left after filtered by the vocabulary of WordNet [Miller, 1995] and ConceptNet [Liu and Singh, 2004].

3.2 Query Searching

We generate queries by pairing any two phrases. Typical queries are "(rob, arrest)" and "(eat food, happy)". In this work, we define 4 patterns for each phrase based on two semantic dimensions, i.e., *positive-negative* and *active-passive* [Osgood, 1952]. Using word *rob* for example, it contains active-positive pattern (*rob*), active-negative pattern (*not rob*), passive-positive pattern (*be robbed*), and passive-negative pattern (*not be robbed*). Therefore, each query has 16 dimensions (see Figure 4). The task to mine the cause-effect pairs is to get the occurrence numbers for all the possible links.

The goal of query searching is to find all the possible sentences that may contain the input queries. Since the number of queries is very large, we structure all the queries as a hashmap and conduct string matching during text scanning. In detail, the searching program starts by conducting lemmatizing, part-of-speech tagging and dependency parsing on the source corpus. After it, we scan the corpus from the begining to end. When dealing with each sentence, we will try to find the matched phrases using the hashmap. This strategy help us to reduce the complexity to be linear with the corpus size.

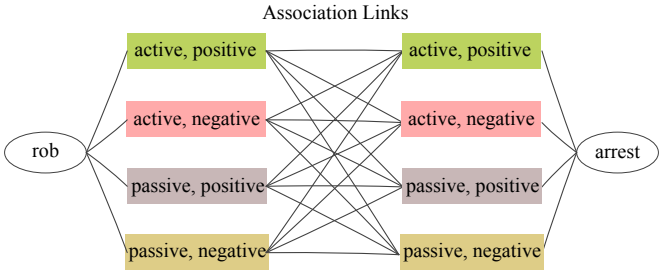

Figure 4: The 16 dimensions for a search query.

3.3 Subject-Object Matching

By conducting dependency parsing on the found sentences, once we detect one phrase of a query, we check whether that phrase is associated with a subject or an object. We also record whether the phrase is positive or negative as well as active or passive. To decide the cause-effect directions, we check whether the phrase is linked with connectives (e.g., *because*) or not. To extract the cause-effect pairs, we design a *subject-object matching* rule: 1) if the two phrases in one query share the same *subject*, the relationship between them is then straightforward; 2) if the *subject* of one phrase is the *object* of the other phrase, then we apply the *passive* pattern to the phrase related to *object*.

Using the query (*arrest*, *rob*) as an example. It appears in "Tom was arrested because Tom robbed the man" (Figure 5). Since *arrest* and *rob* share same subject, and the pattern for *arrest* is passive, we will add the number of the specific association link, i.e. link from the (active,positive) pattern of *rob* to the (passive,positive) pattern of *arrest*, by 1.

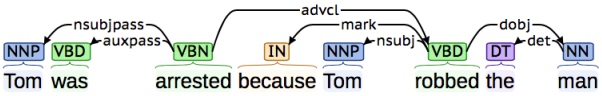

Figure 5: An example of the dependency parsing result.

4 Neural Association Model (NAM)

4.1 NAM in general

Figure 6 presents the framework of NAM for associating two events, E_1 and E_2. In the NAM framework, events are projected into low-dimensional vector spaces. Deep neural networks with multi-layer nonlinearity are used to model how likely these two events are to be associated. Neural networks take the embedding of one event E_1 as input and compute a conditional probability $\Pr(E_2|E_1)$ of the other event E_2. If the event E_2 is binary (true or false), the NAM models may use a *sigmoid* node to compute $\Pr(E_2|E_1)$. If E_2 takes multiple mutually exclusive values, we use a few *softmax* nodes for $\Pr(E_2|E_1)$, where it may need to use multiple embeddings for E_2 (one per value).

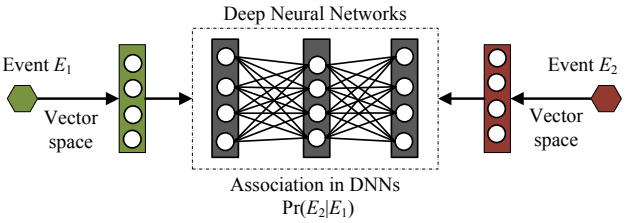

Figure 6: The NAM framework in general.

Learning NAMs

Assume we have a set of N_d observed examples (event pairs $\{E_1, E_2\}$), $\mathcal{D}$, each of which is denoted as x_n. This training set normally includes both positive and negative samples. We denote all positive samples ($E_2 = true$) as $\mathcal{D}^+$ and all negative samples ($E_2 = false$) as $\mathcal{D}^-$. The likelihood function of a NAM model is defined to be:

$$\mathcal{L}(\mathbf{\Theta}) = \sum_{x_n^+ \in \mathcal{D}^+} \ln f(x_n^+; \mathbf{\Theta}) + \sum_{x_n^- \in \mathcal{D}^-} \ln(1 - f(x_n^-; \mathbf{\Theta})) \tag{1}$$

where $f(x_n; \mathbf{\Theta})$ denotes a logistic score function derived by the NAM for each x_n. Stochastic gradient descent (SGD) methods could be used to make a maximum likelihood estimation (MLE). We design two NAM structures with a finite number of output nodes to model $\Pr(E_2|E_1)$ for any pair of events, where we have only a finite number of E_2. The first model is a typical DNN that associates antecedent event (E_1) at input and consequent event (E_2) at output. The second model is a relation-modulated neural network. In this work, all the event pairs have 16 possible cause-effect relationships.

4.2 DNN for NAMs

The first NAM structure is a typical DNN shown in Figure 7. Given a triple $x_n = (e_i, r_k, e_j)$ and its corresponding label y_n (true or false), we cast $E_1 = (e_i, r_k)$ and $E_2 = e_j$ to compute $\Pr(E_2|E_1)$ as follows.

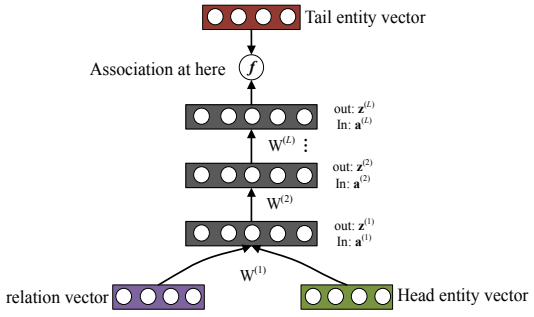

Figure 7: The DNN structure for NAMs.

Firstly, we represent head entity phrase e_i and tail entity phrase e_j by two embedding vectors $\mathbf{v}_i^{(1)} (\in \mathbf{V}^{(1)})$ and $\mathbf{v}_j^{(2)} (\in \mathbf{V}^{(2)})$. Similarly, relation r_k is also represented by a low-dimensional vector $\mathbf{c}_k \in \mathbf{C}$, which we call a *relation code* hereafter. Secondly, we combine the embeddings of the head entity e_i and the relation r_k to feed into an $(L+1)$-layer DNN as input. The DNN consists of L rectified linear (ReLU) hidden layers [Nair and Hinton, 2010]. The input is $\mathbf{z}^{(0)} = [\mathbf{v}_i^{(1)}, \mathbf{c}_k]$. During the feedforward process, we have

$$\mathbf{a}^{(\ell)} = \mathbf{W}^{(\ell)} \mathbf{z}^{(\ell-1)} + \mathbf{b}^{\ell} \quad (\ell = 1, \cdots, L) \tag{2}$$

$$\mathbf{z}^{(\ell)} = h\left(\mathbf{a}^{(\ell)}\right) = \max\left(0, \mathbf{a}^{(\ell)}\right) \quad (\ell = 1, \cdots, L) \tag{3}$$

where $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{\ell}$ represent the weight matrix and bias for layer ℓ respectively.

Finally, we propose to calculate a sigmoid score for each triple $x_n = (e_i, r_k, e_j)$ as the association probability using the last hidden layer's output and the tail entity vector $\mathbf{v}_j^{(2)}$:

$$f(x_n; \mathbf{\Theta}) = \sigma\left(\mathbf{z}^{(L)} \cdot \mathbf{v}_j^{(2)}\right) \tag{4}$$

where $\sigma(\cdot)$ is the *sigmoid* function, i.e., $\sigma(x) = 1/(1+e^{-x})$.

All network parameters of this NAM structure, represented as $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{C}\}$, are jointly learned.

4.3 Relation-modulated Neural Networks (RMNN)

The framework of relation-modulated neural nets (RMNN), which is very similar to the work of [Xue *et al.*, 2014], is shown in Figure 8.

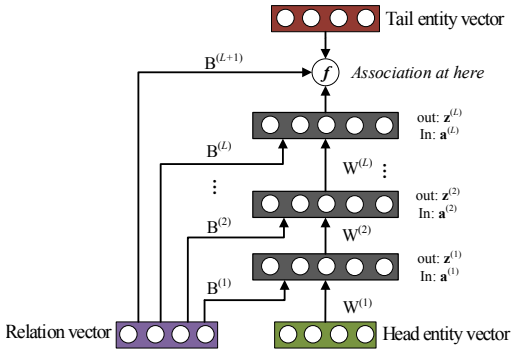

Figure 8: The relation-modulated neural networks (RMNN).

The RMNN uses the same operations as DNNs to project all entities and relations into low-dimensional continuous space. In Figure 8, we connect the knowledge-specific relation code $\mathbf{c}^{(k)}$ to all hidden layers in the network. As shown later, this structure is superior in knowledge transfer learning task. Therefore, for each layer of RMNNs, instead of using eq.(2), its linear activation signal is computed from the previous layer $\mathbf{z}^{(\ell-1)}$ and the relation code $\mathbf{c}^{(k)}$ as follows:

$$\mathbf{a}^{(\ell)} = \mathbf{W}^{(\ell)} \mathbf{z}^{(\ell-1)} + \mathbf{B}^{(\ell)} \mathbf{c}^{(k)}, \quad (\ell = 1 \cdots L) \tag{5}$$

where $\mathbf{W}^{(\ell)}$ and $\mathbf{B}^{\ell}$ represent the normal weight matrix and the relation-specific weight matrix for layer ℓ. At the topmost layer, we calculate the final score for each triple $x_n = (e_i, r_k, e_j)$ using the relation code as:

$$f(x_n; \mathbf{\Theta}) = \sigma\left(\mathbf{z}^{(L)} \cdot \mathbf{v}_j^{(2)} + \mathbf{B}^{(L+1)} \cdot \mathbf{c}^{(k)}\right). \tag{6}$$

In the same way, all RMNN parameters, including $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{B}, \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{C}\}$, can be jointly learned based on the above maximum likelihood estimation.

Schema texts	Verb/Adjective 1	Verb/Adjective 2	Verb/Adjective 3
The man couldn't lift his son because he was so weak	*weak*	*not lift*	*not be lifted*
The man couldn't lift his son because he was so heavy	*heavy*	*not lift*	*not be lifted*
The fish ate the worm. it was tasty	*tasty*	*eat*	*be eaten*
The fish ate the worm. it was hungry	*hungry*	*eat*	*be eaten*
Mary tucked her daughter Anne into bed, so that she could sleep	*tuck into bed*	*be tucked into bed*	*sleep*
Mary tucked her daughter Anne into bed, so that she could work	*tuck into bed*	*be tucked into bed*	*work*
Tom threw his schoolbag down to ray after he reached the top of the stairs	*reach top*	*throw down*	*be thrown down*
Tom threw his schoolbag down to ray after he reached the bottom of the stairs	*reach bottom*	*throw down*	*be thrown down*
Jackson was greatly influenced by Arnold, though he lived two centuries earlier	*live earlier*	*influence*	*be influenced*
Jackson was greatly influenced by Arnold, though he lived two centuries later	*live later*	*influence*	*be influenced*

Table 1: Examples of the cause-effect WS problems labelled from the whole official WS test set.

5 Experiments

5.1 Dataset

In this paper, we manually selected 70 WS problems that relies on cause-effect reasoning from the WS dataset[2]. Each WS problem contains a pronoun (to be resolved) and two main candidate noun phrases. We labeled all the possible events linked to all of them. For instanecs, in the sentence "*The man couldn't lift his son because he was so weak*", we will identify *weak*, *not lift* and *not be lifted* for *he*, *the man* and *son* respectively. The commonsense is that someone who is *weak* would more likely to be associated to *not lift* rather than *not be lifted*. Table 1 shows some more typical examples.

5.2 Knowledge Acquisition Results

Based on the knowledge acquisition method proposed in this work, we conduct experiments on a set of corpora. Table 2 presents all the knowledge acquisition results. In this paper, we named the extracted knowledge as **CauseCom**. We extract about 503,359 cause-effect pairs from different corpora. A typical example is "(rob, active-positive-Cause-passive-positive, arrest)", which indicates that a man who rob (i.e., active and positive pattern of event "rob") would cause a result: be robbed (passive and positive pattern of event "arrest"). To get better evidences for each associated cause-effect pairs, this paper calculates pointwise mutual information (PMI) for them. Figure 9 gives the typical PMI value distribution of the extracted cause-effect pairs.

Corpus	Size	#Result pairs
Gigaword [Graff *et al.*, 2003]	3.6B	283,430
Wikipedia	4.0B	105,071
Novels [Zhu *et al.*, 2015]	984M	106,928
BNC [Consortium, 2007]	100M	7,930
CBTest [Hill *et al.*, 2015]	319M	915

Table 2: Knowledge acquisition results on different corpora.

Validation On the Extracted Knowledge

Based on the motivation to design WS questions, there is no obvious statistical test over text corpora that will reliably disambiguate WS problems correctly. Indeed, people would not speak commonsense explicitly. Therefore, to check whether can we directly extract the exact knowledge for the WS test set, we make a straightforward validation for the extracted knowledge. We try to search and compare the statistical values between the two event pairs of each WS, e.g., "(heavy, not lift)" and "(heavy, not be lifted)" to answer WS problems, however, the hit result is 0 out of 70 problems. This indicates that we cannot answer WS test problems by explicitly utilizing the extracted knowledge with searching strategy. Therefore, the NAM models proposed based on distributed representation and neural network methods are then used to answer these WS test set problems.

[2] http://www.cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html.

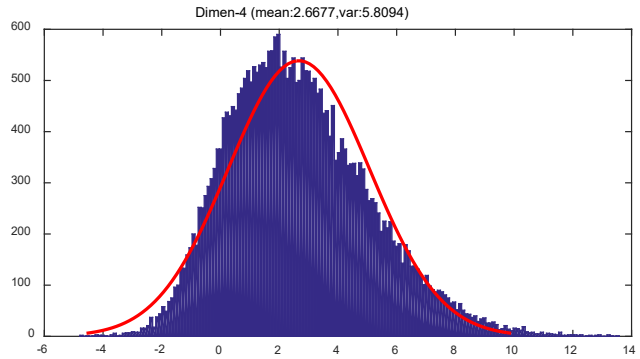

Figure 9: PMI value distribution of the extracted cause-effect pairs (in typical 4th dimension).

5.3 Neural Association Model Setup

This paper treats all the 16 dimensions shown in Figure 4 as distinct relations. So there are 16 relation vectors in the corresponding NAM models. Using the extracted cause-effect pairs, training the NAM models is straightforward. Since there doesn't have training and development set for the WS task, we propose to get the common experimental settings by making experiments on a self-built development set. We randomly split 10,000 cause-effect pairs from the extracted knowledge to be a development set. We then conduct a typical triple classification experiment, i.e., to answer whether there is a cause-effect relationship between two events or not.

Relying on the development experiments, this paper finds the common experimental settings as follows: 1) for entity representations, we represent entities by composing from their word vectors using bag-of-words method. The dimen-

sions for all word vectors are set to be 100. 2) the dimensions of all relation codes are set to be 50; 3) for network structures, we use 3 hidden layers and ReLU as the nonlinear activation function. Dropout [Hinton *et al.*, 2012] is adopted during training; 4) during the learning process of NAMs, negative samples are generated by randomly perturbing positive KB triples as $\mathcal{D}^- = \{(e_i, r_k, e_\ell)|e_\ell \neq e_j \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\}$. In our experiments, the number of negative samples is set to 5 for each positive instance; 5) all models are trained using the SGD algorithm while the learning rate is set to be 0.1.

5.4 Results

Based on the common parameter settings, all the NAM models, i.e., DNN and RMNN are trained using the extracted cause-effect pairs. After the training process is finished, we apply the trained models to answer WS questions directly. Due to the test set is very small, we make significance test and report the corresponding p-value for all the experimental results. Before reporting the results, we need to know that answer accuracy of random guessing is 50% for the WS dataset.

Overall Results

The overall results achieved in this paper are shown in Table 3. From the result, we find the DNN model achieves 65.71% accuracy with a p-value of 0.006, which is significantly better than random guessing. At the same time, the RMNN model achieves 70% accuracy. To the best of our knowledge, this is the first open result achieved on the WS task (with at least 70 WS questions).

Model	Accuracy (%)	p-value
Random guessing	50.00	0.284
DNN	**65.71**	0.006
RMNN	**70.00**	0.001

Table 3: WS answer accuracy with significance test results.

Detailed Results

To make clear the performances of the NAM models on solving WS problems, we list a set of detailed results in Table 4. For all the accuracy numbers of DNN and RMNN, we given the significance test results with respect to random guessing correspondingly. From the results, we find all the accuracies achieved in this paper are larger than 60%. Meanwhile, the p-values of significance test are all smaller than 0.05, which means that those results have significant differences with random guessing. Moreover, using 3 hidden layers helps the models perform best.

Network size	DNN		RMNN	
	Acc	p-val	Acc	p-val
100-[100*1]-100	60.00	0.021	61.43	0.015
100-[100*2]-100	62.86	0.013	65.71	0.007
100-[100*3]-100	**65.71**	0.006	**70.00**	0.001
100-[100*4]-100	61.43	0.017	64.29	0.010

Table 4: Results with different network sizes. For the network size, 100-[100*1]-100 represents that we set 1 hidden layer.

5.5 Final Remarks

Solving WS problems is challenging. We think this work is a reasonable start. Since the WS task is designed to measure the progresses in commonsense reasoning, it is possible to employ logic reasoning methods to build a problem solving system for it. However, those traditional methods have very poor scalability and would fail in many real situations. That's why all the existing works that employed logic reasoning methods to solve WS problems, just solved quite few WS examples [Schüller, 2014; Bailey *et al.*, 2015]. Meanwhile, it is too slow to use logic reasoning methods and the performance is poor. Besides, they are limited by the corresponding background commonsense knowledge. On the other hand, in most cases, we are not expected to extract knowledge online from the Internet, like the work of [Sharma *et al.*, 2015].

In this work, we relax the problem from complex logic reasoning to event association modeling. Modeling the association relationships between large number of discrete events is a fundamental work for AI. To support our motivation, this paper proposes a knowledge acquisition method to extract associated cause-effect pairs from text corpora. The knowledge we constructed in this paper covers a set of common words, which avoids the data sparseness problem. At the same time, the NAM model is built based on the distributed representation and neural network approaches, which has better scalability than traditional logic reasoning methods. Definitely, NAM is not the only choice for solving WS problems based on the extracted cause-effect knowledge. However, it provides us a flexible framework to model the association relationships between discrete events.

6 Conclusions

This paper focuses on the investigations in the novel Winograd Schema (WS) task. We have started this work to address the challenge that no usable commonsense knowledge or any training data exists for the WS task. We propose to address a set of WS problems by using a knowledge acquisition method and a general neural association model. To avoid the data sparseness issue, the knowledge we aim to collect is the cause-effect relationships between a collection of most commonly used words and phrases. Using the proposed method, we extract large number of cause-effect pairs from various text corpora. The extracted knowledge is then used to train a neural association model (NAM), which is proposed to encode the association relationships between any two discrete events. Based on these, we have successfully built a system for solving WS problem. This paper were supported by the Strategic Priority Research Program of the CAS (XDB02070006), the Science and Technology Development of Anhui Province (2014z02006), and the Fundamental Research Funds for the Central Universities (WK2350000001).

References

[Bailey *et al.*, 2015] Dan Bailey, Amelia Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. The winograd schema challenge and reasoning about correlation. In *In Working Notes of the Symposium on Logical Formalizations of Commonsense Reasoning*, 2015.

[Consortium, 2007] BNC Consortium. The British National Corpus, version 3 (bnc xml edition). http://www.natcorp.ox.ac.uk, 2007.

[Davis and Marcus, 2015] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[Davis *et al.*, 2016] Ernest Davis, Leora Morgenstern, and Charles Ortiz. The Winograd Schema Challenge. http://www.cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html, 2016.

[Graff *et al.*, 2003] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 2003.

[Hill *et al.*, 2015] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The Goldilocks Principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[Hinton *et al.*, 2012] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[Kruengkrai *et al.*, 2014] Canasai Kruengkrai, Naoya Inoue, Jun Sugiura, and Kentaro Inui. An example-based approach to difficult pronoun resolution. In *PACLIC*, pages 358–367, 2014.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

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

[Levesque *et al.*, 2011] Hector J Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

[Liu and Singh, 2004] Hugo Liu and Push Singh. ConceptNet: a practical commonsense reasoning toolkit. *BT technology journal*, 22(4):211–226, 2004.

[Liu *et al.*, 2016] Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. Combing context and commonsense knowledge through neural networks for solving winograd schema problems. *arXiv preprint arXiv:1611.04146*, 2016.

[Marcus *et al.*, 2016] Gary Marcus, Francesca Rossi, and Manuela Veloso. Beyond the turing test. *AI Magazine*, 37(1), 2016.

[Michalski *et al.*, 2013] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[Miller, 1995] George A Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Morgenstern and Ortiz Jr, 2015] Leora Morgenstern and Charles L Ortiz Jr. The winograd schema challenge: Evaluating progress in commonsense reasoning. In *AAAI*, pages 4024–4026, 2015.

[Morgenstern *et al.*, 2016] Leora Morgenstern, Ernest Davis, and Charles L Ortiz Jr. Planning, executing, and evaluating the Winograd Schema Challenge. *AI Magazine*, 37(1):50–54, 2016.

[Mueller, 2014] Erik T Mueller. *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann, 2014.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of ICML*, pages 807–814, 2010.

[Osgood, 1952] Charles E Osgood. The nature and measurement of meaning. *Psychological bulletin*, 49(3):197, 1952.

[Peng *et al.*, 2015] Haoruo Peng, Daniel Khashabi, and Dan Roth. Solving hard coreference problems. *Urbana*, 51:61801, 2015.

[Rahman and Ng, 2012] Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proceedings of EMNLP-CoNLL*, pages 777–789, 2012.

[Schüller, 2014] Peter Schüller. Tackling winograd schemas by formalizing relevance theory in knowledge graphs. In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 358–367. AAAI Press, 2014.

[Sharma *et al.*, 2015] Arpit Sharma, Nguyen Ha Vo, Somak Aditya, and Chitta Baral. Towards addressing the winograd schema challenge-building and using a semantic parser and a knowledge hunting module. In *IJCAI*, pages 1319–1325, 2015.

[Sharma, 2014] Arpit Sharma. *Solving winograd schema challenge: Using semantic parsing, automatic knowledge acquisition and logical reasoning*. PhD thesis, Arizona State University, 2014.

[Winograd, 1972] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

[Xue *et al.*, 2014] Shaofei Xue, Ossama Abdel-Hamid, Hui Jiang, Lirong Dai, and Qingfeng Liu. Fast adaptation of deep neural network based on discriminant codes for speech recognition. *Audio, Speech, and Language Processing, IEEE/ACM Trans. on*, 22(12):1713–1725, 2014.

[Zhu *et al.*, 2015] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of ICCV*, pages 19–27, 2015.