Commonsense Reasoning in a Deeper Way: By Discovering Relations between Predicates

Wenguan Huang and Xudong Luo[*]
Institute of Logic and Cognition, Department of Philosophy, Sun Yat-sen University, Guangzhou, China

Keywords: Knowledge Representation, Commonsense Reasoning, ConceptNet, Natural Language Processing.

Abstract: One of the biggest drawbacks of nowadays AI reasoning systems is their lack of commonsense. To address the issue, some commonsense knowledge bases and a bunch of reasoning mechanisms with them have been developed to tackle this problem. However, most of them concentrate on the relation between entities (*e.g.*, "cat" and "fish"), but few discuss the relation between predicates (*e.g.*, "angry" and "shout"), which fall into a deeper level of commonsense. To the end, in this paper, we develop a commonsense reasoning framework, which focuses on this type of commonsense knowledge. More specifically, first we give a formal definition of this kind of commonsense. Then we construct a set of knowledge by extending the predicate set of ConceptNet, and apply information extraction technique to capture them from corpus. Finally, to evaluate our framework, we conduct experiments against a part of the Winograd Schema Challenge, which, its author claimed, is an alternative of Turing Test. The result of our experiments confirms the effectiveness of our framework.

1 INTRODUCTION

Humans have an extremely powerful capability of inferring the meaning even it is expressed implicitly in a sentence. For example, we can understand the metaphor, sarcasm, or humor. Such a capability is built upon the huge scale of commonsense knowledge, which we human gather for ages. For example, by saying *that this smart phone is light but cannot put into my pocket"*, we can immediately know that this phone is too big. However, for an intelligent system that lacks of commonsense, it is a tough task to infer *"size is too big"* from *"cannot put into the pocket"*.

With the importance of commonsense reasoning in many AI tasks (from natural language understanding to computer vision, planning and reasoning), a lot of studies have been done to arm machines with commonsense knowledge bases, so that they can deal with commonsense reasoning (Davis and Marcus, 2015). For example, researchers have built up some commonsense knowledge bases such as Freebase (Bollacker et al., 2008), YAGO (Rebele et al., 2016), and ConceptNet (Speer and Havasi, 2013). Most of them are structured like an entity-relation graph. That is, they represent a commonsense fact as a relation among two entities (*e.g.*, HasProperty(phone, big)), and store it in a graph.

[*]The corresponding author.

Most studies focus on the relation between entities, but few studies are concerned with the relation of predicates. The former is important for forming the whole semantic network, while the latter is significant for umasking the causality (or correlativity) between predicates. For instance, in the above example, we can make such an inference with the commonsense knowledge of "*if A cannot put into B, then it may be the case that A is too large*". Note that it cannot be represented in any of the above bases, since it does not describe any relation between entities, but the relation between two predicates "*cannot put into*" and "*too big*". To this end, in this paper we study how to extract the relations between predicates.

The relation between predicates can have a leverage on commonsense reasoning, by discovering the causality and relativity between them. We can transfer from the former predicates to the latter predicate. On the other hand, finding out these relations can also generate more commonsense knowledge, by applying it to the existing commonsense knowledge bases. In this way, we can make the knowledge base more complete.

To tackle the problem, in this paper, we first introduce a set of predicates $\mathcal{P}$ to expand the expressiveness beyond ConceptNet's built-in predicates (Speer

Huang W. and Luo X.
Commonsense Reasoning in a Deeper Way: By Discovering Relations between Predicates.
DOI: 10.5220/0006120504070414
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 407-414
ISBN: 978-989-758-220-2

and Havasi, 2013). Then we present and discuss a manipulable definition of this kind of relation. Further, we propose two measures to find out these predicate-relation commonsense knowledge, which we termed as *rule patterns*, based on ConceptNet.

The rest of this paper is organised as follows. Section 2 reviews the related work. Sections 3 to 5 detail our framework. Section 6 evaluates our framework against Winograd Schema Challenge (Levesque et al., 2011). Finally, Section 7 concludes the paper with future work.

2 RELATED WORK

The most common methodology to deal with commonsense is collecting a huge number of commonsense facts to enable computer systems to have commonsense. A well-known commonsense knowledge base, which we are based on, is ConceptNet (Speer and Havasi, 2013). ConceptNet is a semantic network containing millions commonsense knowledge contributed by volunteers.

Other knowledge bases like YAGO (Rebele et al., 2016) and Freebase (Bollacker et al., 2008) have similar network-like structures, but ConceptNet exceeds them because it focuses on commonsense facts yet others just concern facts.

Several alternative approaches have been proposed to tackle commonsense reasoning concerning the relation between predicates. For example, Liu and Singh (2004) proposed a number of graph-based algorithms, which can turn reasoning problems into graph problems. Morover, Speer et al. (2008) proposed a novel reasoning method, called AnalogySpace, based on matrix representation and SVD technique. The key idea behind AnalogySpace is to imitate the analogy reasoning, which is a kind of induction inference. This work diversifies the inference method yet is limited as it mainly depends on the similarity between predicates.

Recently, Angeli and Manning (2014) proposed a natural logical inference system for inferring commonsense facts. By using natural logic, a reasoning problem can be embedded in a search framework, which can then be converted into a search tree problem. However, with the limitation of natural logic, the form of reasoning is bounded under just a few types of relation such as inheritance and transitivity.

In the field of reasoning on ontologies, Kazakov et al. (2009) proposed two role axioms of inclusion and transitivity to characterise relations between predicates. Nonetheless, they still only concern with inheritance (inclusion) and transitivity. There are more remain not mentioned, such as causality. Moreover, more flexible relation even between predicates and concepts, *e.g.*, the causality between "eat" and "hungry", are not considered.

On the other hand, relation extraction falls into the field of information extraction. Tandon et al. (2011) proposed a web-scale relation extraction method to extract commonsense knowledge based on seeds from a knowledge base. It uses the idea of pattern matching but makes progress upon it, which is similar to our extraction method. There are still other studies in the similar line such as (Soderland et al., 2010).

Another rule-mining approach proposed in (Berger-Wolf et al., 2013) tries to extract rule-like knowledge about relations in ConceptNet. For instance, $(AtLocation, PartOf, AtLocation)$ is a rule, since triple $(textbook, classroom, school)$ can be an instance of this rule, if the following is valid:

$$\begin{aligned} &AtLocation(textbook, classroom) \wedge \\ &PartOf(classroom, school) \\ \rightarrow &AtLocation(textbook, school). \end{aligned}$$

Our approach is inspired by this work (*i.e.*, we termed such a rule pattern as *syllogism rule pattern*), but in our work, the elements of such triple are not only relations but also predicates extracted from ConceptNet, which can make a dramatical improvement upon the reasoning breadth. Also we discover a more general definition of rule pattern and discuss more measures to extract them.

3 DISCOVERING MORE PREDICATES

In this section we present a method to discover more predicates from ConceptNet, by which we can construct more flexible and meaningful commonsense knowledge.

3.1 Predicates Expansion

ConceptNet represents assertions with pattern $P(a,b)$, where P stands for binary predicate modifying concepts a and b. Although ConceptNet has around 20 predicates (or relations) (e.g., *IsA, RelatedTo* and *AtLocation*), it is far from enough to do predicate reasoning, for example, from *"Mike reads a book"* to *"Mike learns something"*, since it does not have *read* and *learn* as predicates, so we need to put more predicates into account.

3.1.1 Predicate Set Constructing

ConceptNet has a great deal of original concepts that should be viewed as predicates, such as *"eat"*. The idea for finding out such concepts is that a word or phrase that can act as predicate normally has particular Part of Speech (PoS). For example, normally the verbs can be regarded as predicates. Hence, we construct a set of concepts, called predicate set $\mathcal{P}$, to include all the concepts that has predicate-required PoS.

The following are the PoS of the words and phrases that can be predicates:

PoSList:
VERB; VERB NOUN; ADV VERB; VERB PREP; VERB PREP NOUN; VERB NOUN PREP NOUN; ADJ; ADV ADJ

The construction of predicate set $\mathcal{P}$ is straightforward. In ConceptNet, many concepts have detailed informations expressed in URI. For example, concept *thank* is represented as

$$/c/en/thank/v/express\_gratitude\_or\_\cdots$$

where "*v*" after "*thank*" means that its PoS is verb. Hence, we can iterate and check the concepts' PoS denoted in URI, and add it into $\mathcal{P}$ if its PoS is in the PoS list. For those who do not show their PoS explicitly, we apply Stanford PoS tagger (Toutanova et al., 2003) to the original sentence of that concept in ConceptNet.

The previous syntactic rule of assertions can be expressed as

$$a \rightarrow r\ c\ c \tag{1}$$

where *a*, *r* and *c* represent assertion, relation and concept, respectively. This rule means an assertion can be represented as a sequence of a relation, a concept, and another concept.

After constructing predicate set $\mathcal{P}$, the syntactic rules are as follows:

$$a \rightarrow r\ c\ c \ \mid\ r\ c\ p \ \mid\ r\ p\ c \ \mid\ r\ p\ p \tag{2}$$

$$a \rightarrow p\ c^* \ \mid\ p\ c\ a \tag{3}$$

where *p* stands for concepts in $\mathcal{P}$, and c^* stands for any number of *c*.

Rule (2) is a variant of Rule (1), but Rule (3) does let our system become much more expressive. For example we can express sentences including clause like:

$$Think(people, Desire(monkey, eat\_banana))$$

3.1.2 Valuation

Here we present a method to valuate the truth of assertions.

It is easy to define the valuation of assertions with built-in predicates. The only thing that we need to do is to look up whether the assertion is in ConceptNet or not. If it is, return true; otherwise, return false. The hard part is the assertions with predicates in $\mathcal{P}$. For example, the modifier of assertion $help(police, kid)$ does not act as a relation in ConceptNet. The truth values of these assertions normally depend on context. That is, $help(police, kid)$ is not always true nor always false. However, for such predicates, we can still judge whether the assertion is *reasonable* or not. By saying reasonable, we mean the assertion may make sense in most common contexts. For instance, $help(police, kid)$ is much more reasonable than assertion like $help(police, pen)$ or $help(tiger, kid)$. This is because it is common for people to think police officers should help a kid, while the latter is rather nonsense. We claim that this kind of statements, though cannot be determined to be true or false, also contain implicit commonsense.

Putting these ideas together, we can formally give the definition of valuation as follows:

Definition 1. *A reasonable condition of predicate p, denoted as $v(p)$, is a set of pairs $\{(S, O)\}$. In every pair, S and O represent the subject equivalence set and object equivalence sets, respectively.*

Definition 2. *An assertion ϕ is reasonable if and only if, its subject, object pair $(s_\phi, o_\phi) \in v(p_\phi)$, where p_ϕ is the predicate of ϕ.*

For example, suppose $v(hold)$ is $\{(doctor, needle), (secretary, file)\}$. Then assertion *"A secretary holds a file"* is reasonable, while *"A secretary holds a needle"* is not.

3.2 Subject-object Set

The remained procedure is to construct subject-object set $v(p)$ for every predicate p in $\mathcal{P}$. The idea of constructing these sets is as follows: firstly find an original set of objects and subjects from corpus, then add their synonyms, superior and siblings (*e.g.*, *"apple"* and *"banana"*) into the set. Thus, it is converted into an information extraction problem: first match out the sentence from corpus containing the specific predicate, then find the subject and object by Ollie (Schmitz et al., 2012).

To extend the set, we also need add their synonyms, superiors and siblings. This can be done by ConceptNet, since synonyms and superiors have corresponding relations *Synonym* and *IsA*, and siblings correspond to those who owns the same superiors. Note that we may get pairs containing pronoun, and named entities, which are not desired, and so should be excluded.

We find out that novels may contain lots of pronoun and person name, while newspapers contain lots of named entities. So we decide to choose short stories such as fables from Project Gutenberg (Hart, 1971) as corpus. Also, the example sentences of dictionaries are good resource too.

4 RELATION OF PREDICATES

In this section, we will discuss the definition of relation between predicates.

The commonsense knowledge is normally in a form of $P(a,b)$, *e.g.*, "IsA(Labrador, dog)" and "HasA(dog, four legs)". In order to imply "HasA(Labrador, four legs)" from the above two pieces of commonsense, an implicit causal relation between *IsA* and *HasA* is required. That is, if a concept *A* belongs to the other concept *B*, then *A* may also has the properties that *B* has. In this piece of rule-like knowledge, *A* and *B* are viewed as variables, which can apply any concept to it. Note that the premise is unnecessarily commonsense knowledge, it could be other context depended statement, and this rule still holds. We believe that this causality or correlativity between predicates is also another kind of commonsense knowledge. However, the form of this commonsense remains unclear, so in this section we try to give it a formal definition, terming it as *rule patterns*.

4.1 Rule Patterns

In (Berger-Wolf et al., 2013), a syllogism like rule pattern is defined as a triple of relations (ρ_1, ρ_2, γ).

One of the shortages is that they only cover relations that are predefined in ConceptNet. After carefully defining the predicates in $\mathcal{P}$, we can extend its scope to $\mathcal{P}$, and term such a rule pattern as a *syllogism rule pattern*. It will extend the scope of application considerably, but this may reduce the reliability of the rule pattern. That is because in the pure relation version, its premises are always true, while the predicate version concerns only with "reasonable", which may lead to a more general but less true situation.

We define the syllogism rule pattern formally as follows:

Definition 3. *A* ***syllogism rule pattern*** *is a tuple* (ρ_1, ρ_2, γ)*, satisfying*

1. ρ_1*,* ρ_2 *and* γ *are predicates in* $\mathcal{P}$ *or ConceptNet, and*
2. *normally for any concepts* $r,s,t \in \mathcal{C}$*, if* $\rho_1(r,s)$ *and* $\rho_2(s,t)$ *hold, then* $\gamma(r,t)$ *holds,*

where $\mathcal{C}$ *represents the concepts set of ConceptNet.*

Similarly, we can also define another kind of rule pattern termed associative rule pattern with only two predicates as follows:

Definition 4. *An* ***associative rule pattern*** *is a tuple* (ρ, γ)*, satisfying*

1. ρ *and* γ *are predicates in* $\mathcal{P}$ *or ConceptNet, and*
2. *normally for any concepts* $r,s \in C$*, if* $\rho(r,s)$ *holds, then* $\gamma(r,s)$ *holds.*

For example, $(eat, Desires)$ is an associative rule pattern, since for concept tuple $(cat, fish)$, if *"Cat eats fish"*, then *"Cat desires fish"*. Note that we use *normally* in the definition to make the inference fuzzier, since it is hard and unnecessary to find the absolute valid and justified rules in commonsense reasoning.

The negative version of rule pattern is straightforward, *e.g.*, $\neg HasProperty(r,s)$ is true if and only if $HasProperty(r,s)$ is not true.

4.2 Extended Rule Patterns

The rule patterns we defined above are expressive, yet they still cannot deal with the following case. Consider an intuitive commonsense knowledge: *if r is upset then r will yell at s*. It cannot be represented by the rule patterns we defined above. Since the predicates *HasProperty* and *Yell* do not have the associative relation. That is, we cannot say that if someone has a property of something, then she/he yells at it. Rather, *upset* and *yell* seem to have the implicit relation. More concretely, $HasProperty(upset)$ and *yell* have the causal relation (*i.e.*, if someone is upset, she/he may yell at someone else"). In order to cover this kind of knowledge, we extend the definition to make it more flexible as follows:

Definition 5. *An* ***extended associative rule pattern*** *is a tuple* (ρ, γ)*, where* ρ *and* γ *are either predicates in* $\mathcal{P}$ *or ConceptNet, or simple concepts in ConceptNet, satisfying:*

rule	*label*	*rule*	*label*
$\frac{\rho(r,s)}{\gamma(r,s)}$	*pp*	$\frac{\sigma(r,\rho)}{\gamma(r,s)}$	*cp*
$\frac{\rho(r,s)}{\sigma(r,\gamma)}$	*pc*	$\frac{\sigma(r,\rho)}{\sigma(r,\gamma)}$	*cc*

where $\sigma \in \{HasProperty, IsA, CapableOf\}$*;* $r,s \in \mathcal{C}$*; and* $\frac{A}{B}$ *denote the causal relation that "if A then B".*

In the above definition, the label column describes the types of ρ and γ. If it is a common concept in ConceptNet, we label it as '*c*', while if it is a predicate in $\mathcal{P}$ or ConceptNet, we label it as '*p*'. So, (*upset*, *yell*) is labeled as '*cp*'. And the associative rule pattern is

Table 1: Definition of Extended Syllogism Rule Pattern.

rule	label	rule	label	rule	label	rule	label
$\frac{\rho_1(r,s),\rho_2(s,t)}{\gamma(r,t)}$	ppp	$\frac{\sigma(r,\rho_1),\rho_2(r,s)}{\gamma(r,s)}$	cpp	$\frac{\sigma(s,\rho_2),\rho_1(r,s)}{\gamma(r,s)}$	pcp	$\frac{\rho_1(r,s),\rho_2(s,t)}{\sigma(r,\gamma)}$	ppc
$\frac{\sigma(r,\rho_1),\sigma(s,\rho_2)}{\gamma(r,s)}$	ccp	$\frac{\sigma(r,\rho_1),\rho_2(r,s)}{\sigma(r,\gamma)}$	cpc	$\frac{\sigma(s,\rho_2),\rho_1(r,s)}{\sigma(r,\gamma)}$	pcc	$\frac{\sigma(r,\rho_1),\sigma(r,\rho_2)}{\sigma(r,\gamma)}$	ccc

actually a special case whose label is '*pp*', *i.e.*, ρ and γ are always predicates.

And σ appears only when ρ or γ is concept. It is decided by the PoS of this concept. The corresponding transformations are:

PoS		Relation of σ
noun	$\rightarrow$	IsA
adjective	$\rightarrow$	HasProperty
verb	$\rightarrow$	CapableOf

For simplicity, we only consider the above three straightforward and commonly used cases.

Here is an example illustrating the extended associative rule pattern: (*hungry, eat*) is a rule pattern, with label '*cp*', where *hungry* is a concept (labeled with *c*) and *eat* is a predicate (labeled with *p*). σ in this case is default as *HasProperty*. Thus, if we know that *"Mike is hungry"*, we can imply that *"Mike eats something"*.

Similarly, we also extend the syllogism rule pattern to include the above interior relation between verbs and properties:

Definition 6. *A* ***extended syllogism rule pattern*** *is a triple* (ρ_1, ρ_2, γ)*, where* ρ_1*,* ρ_2 *and* γ *are predicates in* $\mathcal{P}$ *or ConceptNet, or simple concepts in ConceptNet, satisfying the rules as shown in Table 1*

4.3 Reverse Rule

An important variation of rule pattern is the reverse rule. Every rule can have a reverse version, which changes the role of subject and object of the conclusion. This is necessary if we consider a commonsense rule (*eat, tasty*), which we want to express is: "If *A* eats *B*, then *B* is tasty." However, according to the definition, it is explained as: "If *A* eats *B*, then *A* is tasty." To differentiate them we use *r*(*eat, tasty*) to denote it instead, which reverses the role of subject and object in conclusion. Formally, the reversed version of rule (ρ, γ) is defined as follows:

Definition 7. *A* ***reversed associative rule pattern*** *is a tuple* $r(\rho, \gamma)$ *with a prefix r, satisfying*

1. ρ *and* γ *are predicates in* $\mathcal{P}$ *or ConceptNet, and*
2. *normally for any concepts* $r, s \in C$*, if* $\rho(r, s)$ *holds, then* $\gamma(s, r)$ *holds.*

Note that the syllogism rule pattern and their extension also have the corresponding reverse version rules, but we omit it for the lack of space.

5 EXTRACTION OF RULE PATTERNS

After the discussion of rule patterns, in this section we will present an efficient approach for extracting rule patterns.

The extraction can be divided into two steps: First we discover a set of potential rule patterns, and then use conditional probabilities to denote their confidence computing by Bayesian method. For those who reach a certain threshold, we regard them as valid rules; otherwise, we discard them.

In a sentence, connectives almost always occur, with a certain relation between the predicates, before and behind it. Our approach focuses on six connectives: *if, because, so, but, though*, and *and*. Each of them has the corresponding regular expression based on text pattern. So, we can use these text patterns to extract potential rules from corpus.

More specifically, we find out all the sentences that contain connective firstly, then extract the predicates from two clauses of it. The extraction process is implemented by Stanford Dependency Parser (Socher et al., 2013). Note that when extracting the predicates, if the predicates are *IsA, HasProperty* or *CapableOf*, we also need to consider the extended situation of rule pattern mentioned above, and take the objects of these predicates into account (*e.g.*, *IsA(pet)* instead of just *IsA*).

5.1 Verification

Once the potential rule patterns are constructed, we need a method to judge whether they are reasonable or not. The intuition of verification is computing a confidence for each rule pattern according to ConceptNet, then its validness can be determined by a predefined threshold.

We use Bayesian formula to calculate confidence

c_r for an associative rule pattern r:

$$c_r = p(\gamma \mid \rho) = \frac{p(\gamma, \rho)}{p(\rho)}. \tag{4}$$

Assume that there are n possible pairs of concept, and $n(\rho)$ represents the number of concept pairs that can be modified by ρ. Then $p(\rho) = \frac{n(\rho)}{n}$, and $p(\gamma, \rho) = \frac{n(\gamma,\rho)}{n}$ denote the probabilities that both premises are true. So, we have:

$$c_r = p(\gamma \mid \rho) = \frac{n(\gamma, \rho)}{n(\rho)}. \tag{5}$$

This equation is to denote how probable γ happens if ρ happens.

To determine whether a rule pattern is reasonable, a threshold of confidence ε is needed. According to Berger-Wolf et al. (2013), we select ε as 5%. So, for every potential rule patterns, if its confidence $c_r > 0.05$, it can be regraded as a rule pattern that is normally valid, or make sense. Note that the confidence rate is low even the rule is valid because of the sparsity of ConceptNet.

The verification of syllogism rule is similar. The confidence c_r of a rule pattern is defined as follows:

$$c_r = p(\gamma \mid \rho_1, \rho_2) = \frac{p(\gamma, \rho_1, \rho_2)}{p(\rho_1, \rho_2)}, \tag{6}$$

where $p(\rho_1, \rho_2)$ presents the probability that both premises are true, and $p(\gamma, \rho_1, \rho_2)$ presents the probability of truth of γ, ρ_1 and ρ_2 at the same time.

5.2 Bias Analysis

In order to analyse the bias induced through out the process, we randomly pick 15 short stories from "Fifty Famous People" (Hart, 1971), and annotate the commonsense knowledge as test set manually. There are 57 pieces of commonsense knowledge in it. Among them, 39 are concerned with the relations between predicates, and others are facts. Our system finds out 24 rules, while 15 are valid, *i.e.*, we reach the precision of 0.625 and recall of 0.385. As far as we know, there may be following main biases:

1. *Subclauses*. When the dependency parser encounters a sentence with subclauses, it works less efficiently, which also lead to a wrong predicate extraction.
2. *Double objects*. The predicates with double objects (*e.g.*, "bring me home") suffer from the difficulty on constructing the subject-object set. It leads to a low quality of verification of these predicates.
3. *Connectives*. Some of the commonsense knowledge does not appear with connectives, or ambiguous one like "as".

6 EVALUATION

For the evaluation part, we apply our system to the Winograd Schema Challenge (WSC) (Levesque et al., 2011). WSC is suggested as an alternative to Turing Test (Turing, 1950) because of its practical advantages.

An original example of WSC is as follows:

Joan made sure to thank Susan for all the help she had given. Who given help?

- *Answer 0: Susan*
- *Answer 1: Joan*

For adults who speak English, the answer to this question is obvious; while for computers, it is a really hard question. WSC is technically a pronoun resolution task, but a tough one, since one cannot get the correct answer simply by syntactic analysis (a deep analysis on semantics or even pragmatics is needed). Therefore, most state-of-art pronoun resolvers perform not very well when facing this challenge.

Some studies have tried to tackle this problem. For example, Sharma et al. (2015) proposed a method to search for the needed commonsense knowledge from web to answer the given question. Rahman and Ng (2012) proposed a hybrid method by integrating 8 techniques with machine learning algorithms, and it reaches the correctness of 73%. The error analysis in their work shows that one of the most contributed component, *Google*, is not good at handling schema that requires a deep understanding of the connection between two clauses. Another most important technique, narrative chains, can capture the relationship between the verb events in the two clauses. However, it cannot capture the relationship between the clauses that are not only verbs, *e.g.*, phrases or cases that are with adjective. So, narrative chains can be regraded as special case of the associative rule pattern. However, our method can capture the relationship between two clauses (predicates), which is difficult for the above systems.

Thus, we select 49 out of 273 schemas, which are hard for the above systems, to deal with (*i.e.,* the schemas that one of or both of the events are not described by verbs). And apply our system to this set of schemas, 33 of them can be successfully answered, and an accuracy of 67.35% is achieved.

The overall process is as follows. Given a schema, firstly extract predicate set S used in the text part and query predicate p_c in the question; then construct a set of potential rules, and verify them. In a potential rule, the last predicate (conclusion's predicate) should be p_c, because we want to lead to a conclusion with p_c, to answer the question. Other predicates are from S.

Table 2: Results of some schema.

schema	valid rule	confidence
fish-worm case	$(HasProperty(hungry), eat)$	6.04%
man-son case	$(HasProperty(weak),\ lift)$	7.20%
pay-generous case	$(pay, HasProperty(generous))$	5.87%
woman-daughter case	$(give\_birth, IsA(woman))$	8.07%
hire-take-care case	$r(hire, take\_care\_of)$	5.14%

For the valid rules, we apply them to the text, and see what kind of answer the rules would lead to. If both kinds of answer can be led to, we compare the confidence of the rules and choose the highest one; otherwise, we fail to answer this question and guess an answer instead. For example, if the predicate set S_t of the above schema is $\{thank, give\_help\}$, and the question's predicate p_c is $give\_help$, then there are only two potential rules: $(thank,\ give\_help)$ and $r(thank,\ give\_help)$. After verifying, we find out their confidence are 2.92% and 7.45% respectively (the later one has a higher confidence). As a result, we can apply it to $thank(Joan, Susan)$ and get $give\_help(Susan, Joan)$. Hence, the answer is Susan.

The rest of the schemas, especially those who require commonsense facts or complicated cases (*e.g.*, with subclauses), remain not easy to be dealt with by our system. Therefore, in future it is interesting to extend our system so that it can deal with such commonsense facts.

Table 2 shows five example schemas that are difficult for other systems, but are easily tackled using the corresponding valid rule pattern used in our system. All of the cases require an understanding of the relationship between two clauses, while in each schema one of the clauses contains non-verb predicate. So in this bunch of schemas, other systems can only answer them randomly.

7 CONCLUSION

It has been a tough struggle for researchers to arm AI systems with commonsense. Most commonsense reasoning approaches proposed till now focus on the relation between entities. Instead, in this paper we make another attempt and develop a commonsense reasoning approach, which aim to extract the relation between predicates. We argue that the correlative relation between two predicates (*e.g.*, *thank* and *help*) also hold an interior commonsense knowledge.

More specifically, we first discuss the definition of these kinds of commonsense, then we show a pattern matching based approach to find out the relation from corpus, and finally we apply our system to tackle a part of the Winograd Schema Challenge, which result shows that our system can successfully answer the daily questions that are hard for other systems.

In the future, it is interesting to identify other rule patterns. In this paper, we define two rule patterns (*i.e.*, associative rule pattern and syllogism rule pattern). Maybe they are not enough to exhaust all the possibilities of conclusions, as we discussed in the evaluation section. Hence, it is necessary to identify more of them, to tackle the subordinate clause case, and the double object case, and so on.

ACKNOWLEDGMENTS

This research was partially supported by the Bairen Plan of Sun Yat-sen University, the Natural Science Foundation of Guangdong Province, China (No. 2016A030313231) and the National Fund of Social Science (No. 13BZX066).

REFERENCES

Angeli, G. and Manning, C. D. (2014). Naturalli: Natural logic inference for common sense reasoning. In *EMNLP*, pages 534–545.

Berger-Wolf, T., Diochnos, D. I., London, A., Pluhár, A., Sloan, R. H., and Turán, G. (2013). Commonsense knowledge bases and network analysis. *Commonsense*.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of data*, pages 1247–1250. ACM.

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

Hart, M. (1971). *Project gutenberg*. Project Gutenberg.

Kazakov, Y. et al. (2009). Consequence-driven reasoning for horn shiq ontologies. In *IJCAI*, volume 9, pages 2040–2045.

Levesque, H. J., Davis, E., and Morgenstern, L. (2011). The Winograd schema challenge. In *AAAI Spring Sympo-*

sium: Logical Formalizations of Commonsense Reasoning.

Liu, H. and Singh, P. (2004). Commonsense reasoning in and over natural language. volume 3215 of *Lecture Notes in Computer Science*, pages 293–306.

Rahman, A. and Ng, V. (2012). Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789.

Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., and Weikum, G. (2016). Yago: A multilingual knowledge base from Wikipedia, Wordnet, and Geonames. In *The Semantic Web ISWC 2016*, volume 9982 of *Lecture Notes in Computer Science*, pages 177–185.

Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

Sharma, A., Vo, N. H., Gaur, S., and Baral, C. (2015). An approach to solve winograd schema challenge using automatically extracted commonsense knowledge. In *2015 AAAI Spring Symposium Series*. Citeseer.

Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y. (2013). Parsing with compositional vector grammars. In *the Association for Computational Linguistics*, pages 455–465.

Soderland, S., Roof, B., Qin, B., Xu, S., Etzioni, O., et al. (2010). Adapting open information extraction to domain-specific relations. *AI Magazine*, 31(3):93–102.

Speer, R. and Havasi, C. (2013). ConceptNet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer.

Speer, R., Havasi, C., and Lieberman, H. (2008). Analogyspace: Reducing the dimensionality of common sense knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 1, pages 548–553.

Tandon, N., De Melo, G., and Weikum, G. (2011). Deriving a web-scale common sense fact database. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 152–157.

Toutanova, K., Klein, D., and Manning, C. D. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 252–259.

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