Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference

Commonsense Knowledge Extraction Using Concepts Properties

Eduardo Blanco
HLT Research Institute
The University of Texas at Dallas
Richardson, TX 75080 USA
eduardo@hlt.utdallas.edu

Hakki C. Cankaya
Dept. of Computer Engineering
Izmir University of Economics
Balcova, Izmir 35330 Turkey
hakki.cankaya@ieu.edu.tr

Dan Moldovan
HLT Research Institute
The University of Texas at Dallas
Richardson, TX 75080 USA
moldovan@hlt.utdallas.edu

Abstract

This paper presents a semantically grounded method for extracting commonsense knowledge. First, commonsense rules are identified, e.g., *one cannot see imaginary objects*. Second, those rules are combined with a basic semantic representation in order to infer commonsense knowledge facts, e.g. *one cannot see a flying carpet*. Further combinations of semantic relations with inferred commonsense facts are proposed and analyzed. Results show that this novel method is able to extract thousands of commonsense facts with little human interaction and high accuracy.

Introduction

Commonsense knowledge encompasses facts that people know and use in their daily lives. It is assumed to be known by average people, therefore it is not verbally communicated most of the time. For example, when John says *I'm going to brush my teeth*, he is implicitly saying that most probably he will do that in the bathroom by the sink, will use a toothbrush and toothpaste, will not swallow but spit the toothpaste, will rinse his mouth, and so forth.

It is widely accepted that in order to make machines more intelligent they need to be aware of the vast amount of knowledge that humans have from their early experiences. If machines are to interact with humans in an intelligent way, they need to have commonsense knowledge (Minsky 2000). Armed with commonsense knowledge, in the context of the sentence above, a machine would know where John is, what utensils he is using, how long the activity takes, etc.

Commonsense knowledge is defeasible and context dependent, which complicates matters greatly. For example, *one can see through windows* is commonsense. However, *very dirty* or *dark tinted* windows do not allow you to see through. *Cameras* do not allow to see through clothes, but *x-ray cameras* do. A *sailboat* could return to port with a broken engine as long as there is wind. *Black clothes are appropriate for a funeral* holds in most Western cultures, but the opposite color, white, is appropriate according to Japanese tradition. *If German students knock on their desks after a lecture, it was a good lecture*, but the same action would be interpreted differently in other countries.

Commonsense knowledge is needed in numerous applications and its unavailability often hinders system performance. For example, applications requiring text understanding and inferences, like question answering, recognizing textual entailments, or extracting implicatures would benefit from commonsense knowledge. In AI, qualitative reasoning, the ability to reason without precise quantitative information (Iwasaki 1997), and analogical reasoning, the ability to solve problems based on past cases (Rissland 2006), are but a few examples of commonsense knowledge applications.

Previous Work

Commonsense knowledge is considered obvious and many researchers (Singh 2002; Lenat 1995; Ahn, Kedia, and Blum 2006) claim that it is unfeasible to obtain it from text or any other existing resource. They believe that humans, experts or non-experts, are needed to obtain commonsense facts.

Cyc is the biggest and oldest project aiming at building a commonsense knowledge base (Lenat 1995). The project started in 1984 and since then experts have introduced millions of commonsense facts using a formal language, CycL. Currently, the Cyc knowledge base contains nearly 500,000 terms, including about 15,000 types of relations, and about 5,000,000 facts (assertions) relating these terms[1].

ConceptNet is a semantic network for commonsense knowledge generated from the data collected by the Open Mind Common Sense Project (OMCSP) (Havasi, Speer, and Alonso 2007). It started in 2000 and it is based on a collaborative effort of thousands of anonymous non-expert users over the internet. The OMCSP project has collected over 700,000 pieces of common sense information in English from 15,000 contributors in 8 years (Speer, Havasi, and Lieberman 2008). In a similar fashion, the Verbosity project (Ahn, Kedia, and Blum 2006) implements an online game to grab commonsense facts from online users.

Systems described so far require human interaction and thus are slow by design. Minsky estimated that between 30 and 50 million commonsense facts are needed (Dreifus 1998); the OMCSP will need between 300 and 500 years to reach that amount of facts at the current acquisition rate. Surprisingly, human interaction cannot be justified by the need of high accuracy. The OMCSP reported that 12% of

[1]Source: http://www.cyc.com/

its contents is *garbage* (Singh et al. 2002). Discarding the *garbage*, 75% of the assertions are rated 4 or 5 on a scale ranging from 1 to 5, where 1 means *false*. Including all the assertions, only 66% are rated 4 or 5.

In opposition to the *humans are needed* assumption, Schubert and his team have shown that it is possible to extract commonsense facts from the Penn TreeBank (Schubert and Tong 2003), which consists of news articles syntactically parsed. Using rules, logic and generalization techniques, they extract commonsense facts (they call it *general knowledge*) such as *Children may live with relatives*, which are not explicitly stated, but can be inferred (Van Durme, Michalak, and Schubert 2009). They have also extracted facts from automatically parsed corpora, hence the only requirement is plain text. Humans judging the factoids they extract indicated that about 2 out of 3 factoids are *perceived as reasonable claims* (Van Durme, Michalak, and Schubert 2009), so the knowledge they extract is as good as the one manually extracted by the OMCSP. The web has also been proposed as a source of commonsense knowledge (Zhui et al. 2008), mainly due to its vast size.

We believe automatic extraction is possible and needed. There are numerous publicly available resources and working on top of them or combining them is a better approach than starting fresh from scratch.

Approach

We propose a semantically grounded method for commonsense knowledge extraction. Our approach is based on the following assumption: *concepts have properties which imply commonsense*, e.g., *edible concepts can be found in kitchens*. We claim that it is possible to infer commonsense knowledge given commonsense rules and a basic semantic representation of concepts. We first outline the basic method, later we give details and discuss refinements.

We identify *commonsense rules* (CS-R), that is, rules r that apply to a concept with a property p. For example, *[one can find in a kitchen]$_r$ [edible]$_p$ concepts* and *[one cannot see]$_r$ [imaginary]$_p$ concepts*. Formally, we denote CS-R(*can find in a kitchen*, *edible*) and CS-R(*cannot see*, *imaginary*). The same properties or rules may appear in more than one commonsense rule. For example, *one can find edible objects in a kitchen, refrigerator or supermarket*.

The most basic *semantic representation* consists of an object's properties, hypernymy and meronymy information. They are encoded by the semantic relations PROPERTY, HYPERNYM and PART-WHOLE (PRO, HYP and PW). For example, we have PRO(*edible*, *vegetables*), HYP(*vegetable*, *lentil*) and PW(*lentil*, *lentil soup*). The easiest way to read a semantic relation R(x, y) is *x is R of y*.

Both commonsense rules and semantic relations are combined through a *metarule* in order to yield a commonsense fact (CS-F). For example, given CS-R(*can find in a kitchen*, *edible*) and PRO(*edible*, *vegetables*), we can infer the commonsense fact *one can find vegetables in a kitchen*. Formally, CS-F(*one can find in a kitchen*, *vegetables*).

Using *extensions*, inferred commonsense facts are combined with other semantic relations in order to infer more facts. Applying the extensions detailed below, we can infer that one can find in a kitchen *lentils* and *lentil soup*. Formally, CS-F(*one can find in a kitchen*, *lentil*) and CS-F(*one can find in a kitchen*, *lentil soup*).

All combinations of CS-R and semantic relations are made within the framework of Composition of Semantic Relations (Blanco, Cankaya, and Moldovan 2010), especifically with the composition operator $\circ$.

Basic Metarule

The basic metarule takes as its premises a commonsense rule and a property, and yields a commonsense fact.

Object properties imply commonsense

Rationale Given a rule r that applies to property p (CS-R(r, p)) and an object x with property p (PRO(p, x)), we can infer that rule r applies to object x (CS-F(r, x)).

Example One [can find in a kitchen]$_r$ [edible]$_p$ objects. A [vegetable]$_x$ is an *[edible]$_p$ seeds or roots or stems [. . .]*[2]. Therefore, one [can find in a kitchen]$_r$ [vegetables]$_x$.

Formalism Formally, we denote $\text{CS-R}(r, p) \circ \text{PRO}(p, x) \rightarrow \text{CS-F}(r, x)$. Following with the example, we have:

$$\frac{\begin{array}{l}\text{CS-R}(\textit{can find in a kitchen}, \textit{edible})\\ \text{PRO}(\textit{edible}, \textit{vegetable})\end{array}}{\text{CS-F}(\textit{can find in a kitchen}, \textit{vegetable})}$$

As explained later, all the information needed to instantiate the basic metarule (commonsense rule and properties) can be automatically extracted.

Extensions

In this section, we explore extensions to boost the number of commonsense facts inferred by the basic metarule. The basic idea is to combine inferred commonsense facts with other semantic relations in order to infer more facts.

Extension 1: Hyponyms Inherit Commonsense Facts Once a commonsense fact CS-F(r, x) has been identified, we can easily infer that rule r also constitutes a commonsense fact for all the concepts with x as their hypernym, i.e., concepts y such that HYP(x, y). The rationale behind this extension is in the definition of the HYPERNYM relation, and its introduction is justified by the existence of resources encoding this kind of hierarchy (e.g., WordNet).

For example, knowing that one *can find vegetables in a kitchen* and *vegetables is a hypernym of lentils*, we can automatically infer that one *can find lentils in a kitchen*. Formally, we denote $\text{CS-F}(r, x) \circ \text{HYP}(x, y) \rightarrow \text{CS-F}(r, y)$. Following with the example, we have:

$$\frac{\begin{array}{l}\text{CS-F}(\textit{can find in a kitchen}, \textit{vegetables})\\ \text{HYP}(\textit{vegetable}, \textit{lentils})\end{array}}{\text{CS-F}(\textit{can find in a kitchen}, \textit{lentils})}$$

[2]WordNet definition for the first sense of vegetable.

Rule, *r*	Property *p*	Concept, *x*	Extension 1, x_1	Extension 2, x_2
cannot check in for flight	sharp	knife	slicer, parer	-
cannot see alive	extinct	dinosaur, moa	trachodon, ornithomimid	-
cannot touch	imaginary	bogeyman, equator	-	-
can see into / out of	transparent	window, lens	quarterlight, condenser	car, cockpit
will roll in inclined path	round	ball	golf ball, tennis ball	-
spills if not in a container	liquid	beverage, soup, draft	coke, potage, gazpacho, vichyssoise	-
can eat	edible	potato, radish	french fries, mashed potatoes	-

Table 1: Examples of commonsense knowledge facts extracted using the basic metarule $\text{CS-R}(r, p) \circ \text{PRO}(p, x) \rightarrow \text{CS-F}(r, x)$. Using extension 1 adds the facts CS-F(r, x_1), and using extension 2 adds the facts CS-F(r, x_2).

Extension 2: Wholes Inherit Commonsense Facts Once a commonsense fact CS-F(r, x) has been identified we can, under certain restrictions, infer that rule r also constitutes a commonsense fact for the wholes y of which x is a part of.

For example, knowing that *one can find lentils in a kitchen* and *lentils is a part (ingredient) of lentil soup*, we automatically infer CS-F(*one can find in a kitchen*, *lentil soup*) Formally, we denote $\text{CS-F}(r, x) \circ \text{PW}(x, y) \rightarrow \text{CS-F}(r, y)$. Following with the example, we have:

$$\frac{\begin{array}{l}\text{CS-F}(\textit{can find in a kitchen}, \textit{lentils})\\ \text{PW}(\textit{lentil}, \textit{lentil soup})\end{array}}{\text{CS-F}(\textit{can find in a kitchen}, \textit{lentil soup})}$$

Restrictions and Exceptions

The basic metarule and both extensions come after examining examples of concepts and their semantic representations. Intuitively, given *correct* commonsense rules and properties, the metarule should be able to generate commonsense facts with high accuracy. Extension 1 is relatively straightforward; given a correct commonsense fact and hypernym information, the new inferred facts should also hold.

Extension 2 needs further explanation. Restrictions and exceptions are needed in order to combine a commonsense fact CS-F(r, x) and meronymy information PW(x, y) to obtain a new commonsense fact CS-F(r, y). First, rule r should not encode physical characteristics such as weight or size. Simply put, rules stating physical characteristics of parts do not transfer to their wholes. For example, *[one can lift]$_r$ [car seat cushions]$_x$, [car seat cushions]$_x$ are part of [cars]$_y$*, and yet one *cannot lift cars*. Second, rule r should not encode an event. We define an event as something that happens at a time and place and implies a change to some concept. For example, *roll* and *burn* are events, whereas *can see alive* and *is not likable* are not. Consider the commonsense rule CS-R(*will roll in inclined path*, *round*). Since it encodes an event, it does not qualify for extension 2. Otherwise, we would make mistakes: *[balls]$_x$ [can roll on inclined paths]$_r$, some [computer mice]$_y$ have [balls]$_x$*, and yet *computer mice do not roll on inclined paths*. On the other hand, consider the commonsense fact CS-F(*cannot see alive*, *dinosaur*). It encodes a state and therefore anything having as part a dinosaur *cannot be seen alive*.

We denote restrictions for the composition operator ∘ after the premises, introduced by an ampersand and between brackets. Extension 2 is finally defined as $\text{CS-F}(r, x) \circ \text{PW}(x, y)\&[\textit{r is a state, no physical}] \rightarrow \text{CS-F}(r, y)$.

Table 1 gives examples of commonsense rules (CS-R(r, p)), properties (PRO(p, x)), commonsense knowledge facts extracted using the basic metarule (CS-F(r, x)), and facts extracted after using extension 1 (CS-F(r, x_1)), and extension 2 (CS-F(r, x_2)).

Implementation

In this section, we discuss possible sources of commonsense rules, the extraction of the basic semantic representation and the instantiation of the metarule and extensions.

Sources of Commonsense Rules

Commonsense rules state a rule r that holds given a property p. Formally, we denote CS-R(r, p). We obtained commonsense rules both manually and automatically.

Humans as a source of commonsense rules First, we performed a simple study to manually define commonsense rules. Three subjects were asked to fill the gap in the following statement: *If something has property p, then it [____]$_r$*. They were presented with the statement 30 times, each with a different property extracted from the adjective collection in WordNet 2.0. A property was presented as a WordNet synset and its gloss, so no ambiguities in meaning are possible. Subjects were told to be as general and succinct as possible. Their answers were manually reviewed and clustered, and a commonsense rule of the form CS-R(r, p) for each property was generated. Some of the rules obtained by this method are CS-R(*spills if not in a container*, *liquid.a.1*), CS-R(*can kill if eaten*, *toxic.a.1*), CS-R(*will occur again*, *periodic.a.1*) and CS-R(*requires power to work*, *electric.a.1*).

WordNet as a source of commonsense rules Commonsense rules have also been automatically extracted from existing resources. WordNet (Miller 1995) contains, among others, definitions for adjectives, which typically encode properties of concepts. Unlike other resources, WordNet glosses tend to follow patterns and knowledge extraction from them is easier than from regular text. We identified several patterns that allow us to automatically extract commonsense rules, e.g. `able to VP`.

For this task, we use eXtended WordNet[3] (XWN), which enhances WordNet glosses by adding POS tags, syntactic annotation, word sense disambiguation and logic forms. In this work, we do not make use of the logic forms.

[3]`http://xwn.hlt.utdallas.edu/`

Pattern	Rule r	Example		
		Synset	Gloss	Commonsense Rule
able to [VP]	can [VP]	spongy.a.2	… able to [absorb liquids]$_{VP}$	CS-R(*can absorb liquids*, *spongy.a.2*)
not able to [VP]	cannot [VP]	illiterate.a.1	not able to [read or write]$_{VP}$	CS-R(*cannot read or write*, *illiterate.a.1*)
capable of [VP]	can [VP]	concrete.a.1	capable of [being perceived by the senses]$_{VP}$	CS-R(*can be perceived by the senses*, *concrete.a.1*)
(not capable | incapable of) [VP]	cannot VP	nonlethal.a.1	not capable of [causing death]$_{VP}$	CS-R(*cannot cause death*, *nonlethal.a.1*)
causing [NP]	can cause [NP]	annoying.a.1	causing [irritation or annoyance]$_{NP}$	CS-R(*can cause irritation or annoyance*, *annoying.a.1*)
lacking [NP]	does not have [NP]	plain.a.6	lacking [embellishment or ornamentation]	CS-R(*does not have embellishment or ornamentation*, *plain.a.6*)

Table 2: Patterns used for extracting commonsense rules from WordNet and examples. All tokens carry over their word sense number as provided by XWN. A synset is indicated as *word.POStag.sense*.

Table 2 shows the patterns and examples of commonsense rules automatically extracted. The pattern matching procedure loops over all adjectives in WordNet and checks which glosses match a pattern; a commonsense rule is extracted for each match. Syntactic annotation (VP and NP) comes straight from XWN.

A total of 1160 commonsense rules were extracted. Note that each synset may be associated with several lemmas, meaning that each commonsense rule may be expressed by several words semantically expressing the same property. For example, the synset *transparent.a.1* has the following lemmas: *crystalline*, *crystal clear*, *limpid*, *lucid*, *pellucid* and *transparent*. After expanding each property synset with all its lemmas, 2,020 commonsense rules are generated.

Algorithm 1 Procedure to extract commonsense rules from ConceptNet

```
props = []; count = {}
props = all properties p such that HasProperty(x, p)
for all p in props do
    HasP = all concepts x such that HasProperty(x, p)
    for all R such that R(x, y) and x ∈ HasP do
        count[p,R][y] ++
    end for
end for
```

ConceptNet as source of commonsense rules ConceptNet is a commonsense knowledge semantic network in which concepts are connected via relations. Among others, it encodes HasProperty, UsedFor and AtLocation. We followed the pseudo code described in Algorithm 1 to extract commonsense rules from ConceptNet. In simple terms, the algorithm counts how many concepts *y* are related with objects having a property *p* via a relation R. For example, we obtained that 8 concepts having property *edible* are AtLocation *supermarket*, 6 AtLocation *fridge*, and 11 concepts having property *fun* are UsedFor *entertainment*. For the inner loop, we only consider the relations CapableOf, UsedFor, Cause and AtLocation; the rest did not provide any valid commonsense rules. We only consider assertions with positive polarity, with a score higher than one and frequency higher than three.

Commonsense rules are extracted by using the triples *(p, R, y)* with higher counts. Given *count[p,R][y] = n*, we generate CS-R(*can be R y*, *p*). For example, given *count[edible, AtLocation][kitchen] = 12*, we generate CS-R(*can be AtLocation kitchen*, *edible*). Some examples are:

- concepts with *fun* property can be UsedFor fun, entertainment, enjoyment, play, relax, learn, pleasure, relaxation, competition, get exercise, recreation, …
- concepts with *sweet* property can be UsedFor eat, make jelly, make wine, flavor dessert, sexual pleasure, fun, show affection person, affection, say hello, relaxation, …
- concepts with *bad* property can Cause death, guilt, cancer, get lung cancer, go to jail, die, lung cancer, war, destruction, distrust, deceive, pollution, …
- concepts with *edible* property can be AtLocation supermarket, store, market, fridge, plate, refrigerator, dinner, house, grocery store, refrigerator, home, freezer, …

Note that because ConceptNet is not semantically disambiguated, different meanings for certain properties are mixed, e.g., *sweet person* and *sweet food*. Also, highly ambiguous adjectives are not useful for extracting commonsense rules. For example, *bad* has 14 different meanings according to WordNet (e.g. defective, unsound, immoral); the rules listed above for *bad* are not usable without semantic disambiguation. In this work, we manually assign synsets to the commonsense rules extracted from ConceptNet.

More thoughts about commonsense rules The three proposed methods differ significantly. Manually given rules are slow to obtain and usually express complex rules, e.g., *if something has property sweet, it can result in weight gain if eaten in excess*. On the other hand, rules automatically extracted from WordNet are more basic, almost always valid and very valuable. For example, *weatherproof.a.1 concepts can withstand exposure to weather without damage*, *free.a.1 concepts can act at will* and over 1100 more commonsense rules. Some rules automatically extracted from ConceptNet are valuable, but most of them are not useful to instantiate metarules. While WordNet encodes definitions that can hardly be proven wrong, ConceptNet encodes commonsense which is defeasible by definition. For example, an invalid rule extracted from ConceptNet is *green objects are* AtLo-

Relation	Abbr.	Definition	Example
PROPERTY	PRO(x, y)	x is a property of Y	[candy]$_y$ is [sweet]$_x$
PART-WHOLE	PW(x, y)	x is a part of y	[foot]$_x$ is a part of [human]$_y$
HYPER	HYP(x, y)	x is a hypernym of y	[car]$_x$ is a hypernym of [convertible]$_y$

Table 3: Semantic relations used in this work.

CATION *garden*. Although intuitively right, something green does not have to be in a garden: emeralds are green by definition and rarely found in gardens.

It is important to note that properties are represented by synsets, not words. For example, consider the first three meanings of the adjective *flexible*: (1) capable of being changed; (2) able to flex [. . .]; and (3) able to adjust readily to different conditions . Three commonsense rules are extracted: (1) CS-R(*can change*, *flexible.a.1*); (2) CS-R(*can flex*, *flexible.a.2*); and (3) CS-R(*can adjust readily to different conditions*, *flexible.a.3*) . When combining commonsense rules and properties with a metarule, the matching is enforced between synsets, not words. Thus, *flexible.a.03 person* will correctly be combined only with the third rule.

Algorithm 2 Procedure to instantiate the basic metarule and extensions

```
for all CS-R(r, p) do
  for all PRO(p, x) do
    yield CS-F(r, x) {Metarule 1}
    for all HYP(x, y) do
      yield CS-F(r, y) {Extension 1}
    end for
    for all PW(x, y) do
      if r is a state, no physical characteristic then
        yield CS-F(r, y) {Extension 2}
      end if
    end for
  end for
end for
```

Basic Semantic Representation

The basic semantic representation consists of the relations PROPERTY, HYPERNYM and PART-WHOLE as defined in Table 3. We use HYPER and PART-WHOLE annotation present in WordNet without any further improvements. In addition, we use the output of a state-of-the-art semantic parser tuned for those three relations. The parser has participated in the SemEval07 Task 4 (Badulescu and Srikanth 2007).

In this work, we apply our method to WordNet glosses. Note that XWN provides parse trees and word senses for WN glosses, so a wealth of annotation is available. Also, detecting the three relations in WordNet glosses is a much easier task for the semantic parser than detecting relations from open text. First, syntactic and semantic annotation is available and heavily used by the parser. Second, WordNet glosses use simple English words and grammar.

How does it work?

The method is based on instantiating the basic metarule and applying extensions. After commonsense rules are extracted, a manual checking takes place. Manually defined rules and the ones automatically extracted from WordNet are always considered valid. Rules coming from ConceptNet are often not valid, as discussed before. All rules regardless of their source are annotated as encoding an event or state and as encoding or not a physical characteristic such as weight or size. This annotation is used to decide whether or not the rule can instantiate extension 2.

	Source	# CS-F Total	# CS-F Unique	Avg. Score
Basic metarule	M	1,103	848	4.07
	WN	1,739	1,286	4.02
	CN	4,207	3,153	4.16
	All	7,049	5,287	4.11
Basic + ext 1	M	4,484	2,801	4.38
	WN	5,415	3,671	3.81
	CN	24,272	10,710	4.32
	All	34,171	17,182	4.22
Basic + ext 2	M	151	118	2.91
	WN	257	161	2.19
	CN	951	728	4.04
	All	1,359	1,007	3.61
All	M	5,738	3,767	4.26
	WN	7,411	5,118	3.81
	CN	29,430	14,591	4.27
	All	42,579	23,476	4.17

Table 4: Number of commonsense facts obtained using the basic metarule, extension 1 and extension 2. Results are reported depending on the source of commonsense rules (manual (M), WordNet (WN), ConceptNet (CN)).

It is important to note that validating and annotating an automatically generated commonsense rule is a significantly less labor intensive task than manually defining a rule from scratch. When subjects were asked to define rules, it took them on average 3 minutes per rule, and a post processing is needed in order to cluster different rules by different subjects (on average, it took 5 minutes to cluster subjects' responses and generate a rule). However, validating and annotating a commonsense rule only took on average 30 seconds. In other words, manual generation of a rule using 3 subjects took 14 minutes and 30 seconds of human labor, whereas using automatic generation and manual post processing took only 30 seconds. The average time required for manual validation and annotation may seem abnormally short. We found that commonsense rules CS-R(r, p) can be often discarded in bulk depending on the property p, e.g. all rules of the form CS-R(*can be AtLocation* ____, *green*) are discarded without further analysis.

The next step is to obtain the basic semantic representation. We do so using WordNet annotation for HYPERNYM

The statement above is a reasonably clear, entirely plausible general claim and seems neither too specific nor too general or vague to be useful:
5. I agree
4. I lean towards agreement
3. I'm not sure
2. I lean towards disagreement
1. I disagree

Figure 1: Instructions for assigning a score to a commonsense fact (Van Durme, Michalak, and Schubert 2009).

and PART-WHOLE, and the semantic parser for PROPERTY. Finally, the basic metarule and extensions are instantiated following Algorithm 2. The procedure loops over all the commonsense rules CS-R(r, p), finding instantiations of the metarule (i.e., looking for PRO(p, x)), and instantiations of both extensions (i.e., looking for HYP(x, y) and PW(x, y)).

Results and Evaluation

A total of 1390 commonsense rules (30 from humans, 1160 from WordNet and 200 from ConceptNet) were used during our experiments. When combining them with the semantic representation following Algorithm 2, a total of 23,476 different commonsense facts were generated.

Evaluating the validity of commonsense knowledge is a difficult task. First, because of the vast amount of generated facts, it is hard to make an exhaustive evaluation. Second, the intrinsic defeasibility of this kind of knowledge makes it hard to decide if a fact is valid or not as a binary decision. Following previous evaluation methods, we randomly selected a sample of generated facts (10%) and assessed them according to the instructions in Figure 1.

Table 4 shows the amount of commonsense facts inferred by the basic metarule and both extensions depending on the source of commonsense rules, as well as their average score. The validity of the commonsense facts extracted is comparable with previous aproaches to commonsense extraction (Singh 2002; Van Durme, Michalak, and Schubert 2009).

The vast majority of commonsense rules used in our experiments (83%) are automatically extracted from WordNet, and they obtain an average score of 3.81. Manual rules and the ones extracted from ConceptNet, whose validity are manually checked, obtain an average score of 4.26 and 4.27. In general terms, the basic metarule and extension 1 obtain higher scores than extension 2 (4.11 and 4.22 vs. 3.61).

Conclusions

We have presented a highly automated method to extract commonsense knowledge with high accuracy. Commonsense rule extraction requires minimal human interaction. Semantic relations are extracted either from publicly available resources or a semantic parser. Instantiating the rule and applying extensions is a completely automated process.

We intend to extend this method is several ways. More extensions using semantics could be built, e.g., given CS-F(r, x) and knowing that y is the PURPOSE or EFFECT of x, we could infer CS-F(r, y). We envision this method of acquiring commonsense knowledge as part of a commonsense reasoning system that will provide commonsense knowledge on demand given a domain and context.

References

Ahn, L. V.; Kedia, M.; and Blum, M. 2006. Verbosity: a game for collecting common-sense facts. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, 75–78. New York, NY, USA: ACM.

Badulescu, A., and Srikanth, M. 2007. Lcc-srn: Lcc's srn system for semeval 2007 task 4. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 215–218.

Blanco, E.; Cankaya, H. C.; and Moldovan, D. 2010. Composition of Semantic Relations: Model and Applications. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*.

Dreifus, C. 1998. A conversation with Dr. Marvin Minsky. Why Isn't Artificial Intelligence More Like the Real Thing? The New York Times.

Havasi, C.; Speer, R.; and Alonso, J. 2007. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Recent Advances in NLP*.

Iwasaki, Y. 1997. Real-World Applications of Qualitative Reasoning. *IEEE Intelligent Systems* 12(3):16–21.

Lenat, D. B. 1995. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM* 38(11):33–38.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38:39–41.

Minsky, M. 2000. Commonsense-based interfaces. *Commun. ACM* 43(8):66–73.

Rissland, E. L. 2006. AI and Similarity. *IEEE Intelligent Systems* 21(3):39–49.

Schubert, L., and Tong, M. 2003. Extracting and evaluating general world knowledge from the Brown corpus. In *Proc. of the HLT-NAACL 2003 Workshop on Text Meaning*.

Singh, P.; Lin, T.; Mueller, E.; Lim, G.; Perkins, T.; and Li Zhu, W. 2002. Open Mind Common Sense: Knowledge Acquisition from the General Public. Springer Berlin.

Singh, P. 2002. The Public Acquisition of Commonsense Knowledge. In *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access.*

Speer, R.; Havasi, C.; and Lieberman, H. 2008. AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*.

Van Durme, B.; Michalak, P.; and Schubert, L. 2009. Deriving generalized knowledge from corpora using WordNet abstraction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 808–816.

Zhui, Y.; Zang, L.-J.; Wang, D.-S.; and Cao, C.-G. 2008. A manual experiment on commonsense knowledge acquisition from web corpora. In *2008 International Conference on Machine Learning and Cybernetics*, 1564–1569. IEEE.