Using English Dictionaries to generate Commonsense Knowledge in Natural Language

Ali Almiman
University of Manchester
UK, Manchester
M13 9PL
a.almiman@mu.edu.sa

Prof. Allan Ramsay
University of Manchester
UK, Manchester
M13 9PL
allan.ramsay@cs.man.ac.uk

Abstract

This paper presents an approach to generating common sense knowledge written in raw English sentences. Instead of using public contributors to feed this source, this system chose to employ expert linguistics decisions by using definitions from English dictionaries. Because the definitions in English dictionaries are not prepared to be transformed into inference rules, some preprocessing steps were taken to turn each relation of word:definition in dictionaries into an inference rule in the form left-hand side ⇒ right-hand side. In this paper, we applied this mechanism using two dictionaries: The MacMillan Dictionary and WordNet definitions. A random set of 200 inference rules were extracted equally from the two dictionaries, and then we used human judgment as to whether these rules are 'True' or not. For the MacMillan Dictionary the precision reaches 0.74 with 0.508 recall, and the WordNet definitions resulted in 0.73 precision with 0.09 recall.

1 Introduction

According to Lin and Pantel (2001), text is considered to be the most important source of human knowledge. Thus, various algorithms have been implemented in the field of text mining, e.g., document clustering Larsen and Aone (1999), identifying prototypical documents Rajman and Besançon (1998), or finding term associations Lin et al. (1998) hyponym relationships Hearst (1992), and discovering inference rules Lin and Pantel (2001). This paper presents another way of extracting inference rules using English dictionaries, for instance: *"X is very old ⇒ X is aged", "X use a bicycle ⇒ X cycle", "X provide Y ⇒ X afford Y" and "X does Y in every occasion ⇒ X always Y".*
In some of the natural language processing (NLP) applications, common sense knowledge is considered to be an important topic. One of the common sense resources is inference rules. For instance, consider the query to a textual entailment (TE) system: *T: "The vaccination also affords protection against polio." H: "The vaccination also provides protection against polio."* If the system cannot recognise the relationship between "X afford Y" and "X provide Y", it might not be able to find the entailment between these two sentences. We call them inference rules as they contain directional (asymmetric) relations.
A traditional way of generating inference rules is to write them manually. However, although handcrafted rules might be very reliable and accurate, they are not the ideal procedure way to follow in order to generate thousands of rules. This is because it consumes a lot of time and effort, and it is not very likely that humans could make a complete set of rules. Some previous attempts have used public contributors to feed their set of common sense knowledge; for instance, Open Mind Common Sense (OMCS) Singh et al. (2002) and YAGO Suchanek et al. (2008) encourage people to participate in feeding their data by playing word games such as Verbosity. Although permitting the public to contribute to databases aids in enlarging these databases, it may also affect the projects credibility as there are no restrictions on the contributors' background and expertise. Another way of discovering inference rules was the work resulting from the Discovery of Inference Rules from Text algorithm (DIRT) by Lin and Pantel (2001). Lin and Pantel have applied a distributional hypothesis to paths in dependency trees in order to extract similar contexts, and then they generate in-

Proceedings of Recent Advances in Natural Language Processing, pages 58–63,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_009

ference rules.

In this paper, we want to make parsable rules that can be used in an inference engine that operates over parsed statements and questions. A suitable resource from which to extract inference rules are language dictionaries, as they show relationships between a word and its definition. However, dictionary definitions are not written as inference rules. Therefore, we need to preprocess the definitions into the required form. However, a lot of what can be found in dictionaries cannot be preprocessed into that form; therefore, we used a series of patterns to recognise sentences that will be transformable and subsequently we do the transformation.

The reminder of this paper is organized as follows. In the next section, we illustrate why we have chosen the MacMillan dictionary, followed by a discussion of some preprocessing steps. In section 3, we illustrate the mechanism of how we construct inference rules out of given definitions in The MacMillan dictionary. In section 3, we test the same mechanism on another dictionary (the WordNet definitions). In section 4, we evaluate our work by using human judgment to check the validity of the extracted inference rules. Finally, in section 5, there is a conclusion and future research.

2 The MacMillan Dictionary (TMDC)

A major advantage of language dictionaries is that they offer definitions for each word in a particular language. These definitions may be written differently from one dictionary to another in terms of length and complexity; however, we aim to use shorter definitions in order to make clear and concise inference rules. An appropriate choice for this task is *"The MacMillan Dictionary"* (TMDC), as it is usually utilizes short and patterned sentences in its definitions when compared to other well-known dictionaries, such as the Oxford and Cambridge dictionaries.

2.1 Word Collection

In addition to containing simple-form definitions, TMDC contains a set of 7,500 marked words that are believed to denote 93% of the everyday English words[1]; this set of words is used as the scope for this experiment during the current phase. This set of words is not provided in a separate list, and the only way to check whether a word is marked is to look up the targeted word in the dictionary. An automated solution was used to collect these words by employing Wiktionary's top 100,000[2] most frequently-used English words and looking up all of these words. If a word is in red font, it is an important word that needs to be added to our collection. The red words are added to our list of word collection if the source parge of the definition of the target word contains the class `<h1 class="redword">`. Each of these words has a part of speech tag (POS), and for common sense knowledge we are looking for the open class domain words, i.e., nouns, verbs, adjectives and adverbs. Additionally, some words are amongst the important words in one tag and are not in the other tags; e.g., *"knot"* is only added as a noun, not as a verb. After filtering the results, we have a collection of 6,606 words; Table [1] shows the categories of the extracted words and some statistics about them.

No	Category	Count
1	abbreviation	1
2	adjective	1283
3	adverb	335
4	conjunction	16
5	determiner	10
6	interjection	12
7	modal verb	10
8	noun	3447
9	number	6
10	preposition	22
11	pronoun	33
12	verb	1431
Total		6,606

Table 1: Collected domain

2.2 Definitions Extraction

In this step, the definitions for the extracted open-class words from the previous phase are investigated. For each word from the previous step, we have extracted its definitions and saved the word with its POS and definitions in a new dictionary. The target words represent the keys of that dictionary, and each word has an inner-dictionary with

[1] Source: http://www.macmillandictionaryblog.com/the-words-you-need-follow-the-red-words-and-stars

[2] If Wicktionary does not list all the tenses and forms for every word, then it is expected that the top 10,000 words is enough to do this step.

```
<span class="DEFINITION" resource="dict_
british"><span class="SEP
DEFINITION-before"> </span>a <a href=
"http://www.macmillandictionary.com/
dictionary/british/person" class="QUERY"
title="person">person</a></span> <div
class="EXAMPLES"...
```

Figure 1: A portion of a definition source page from MacMillan Dictionary

POS and definitions as values within it. To determine the POS, we have looked for the class `<span class="PART-OF-SPEECH">` in the source code, and have only added its definitions if it lies within one of the open-class word tags. Every definition has an example; therefore, it was easy to allocate the definitions on the source page between the definition class `<span class="DEFINITION">` and the examples class `<div class="EXAMPLES">`. Some words in the definitions have their own pages in the dictionary; these words are written in the definition class with a link to their pages. For example, the Figure [1] shows an example for the word (*"person"*) that is listed with the link to its own page in the dictionary.

2.3 Definiens Patterns

In order to generate inference rules out of these definitions, we need to turn each word:definition relationship into a parsable sentence. For instance, the sentence that we would like to receive from the definition *"human: a person"* is: *"X is a human if X is a person"*. Similarly, for all similar definitions, i.e., nouns with definitions constructed of a determiner and a noun should be produced in a similar way. To make rules for producing such sentences, we looked for the most frequently used patterns amongst each of the open-class word definitions. In order to find the most common patterns, we assigned another feature to the dictionary generated from the previous step called *"tagged-definition"*. Each definition has a tagged definition that is a copy of the definition that has been tokenized, and each of its words is assigned to its POS. For instance, the definition *"a person"* has the tagged definition `[a!!DT person!!NN]`, where the two exclamation marks are used to split the word from its POS. To produce these tags, we employ a *"brill tagger"* with the *"Maximum likelihood tagger"* as an underlying tagger that are trained on the English treebank from the Universal Dependency Treebank (UDT). After applying this feature to all of the definitions in the dictionary, we have counted how many times each pattern occurrs. A definition pattern is the sequence of the part of speech tags that is written in the tagged definition. From the collected pattern counters, we see that some patterns are used very frequently. Among the noun class definitions, the pattern `[DT NN]` *"human: a person"* occurs more than 80 times, and the pattern `[DT NN NN]`, e.g., *"advisory: an official warning"*occurs a similar number of times. Within the adjective definitions, the most commonpattern is `[RB JJ]`, e.g., *"ancient: very old"*, used in nearly 90 definitions, and `[RB]` is repeated more than 40 times amongst the adverbs, e.g., *"absolutely: completely"*. Nearly all of the patterns for the verb definitions begin with `TO`, and the most frequent pattern is `[TO VB DT NN]`, occurring more than 65 times, e.g., *"conquer: to win a victory."*

2.4 Re-writing Definitions

As mentioned in section [2.3], all of the words obtained and their definitions have to be transformed into complete sentences. To re-write these definitions, there are two main factors that specify the final output of the sentencethe word class, i.e., the noun, verb, adjective or adverb, and the definition pattern.

Nouns:

It is observed that the definition patterns for nouns very often begin with a determiner (DT. In addition to the `[DT NN]` and `[DT NN NN]` patterns mentioned earlier, there are other patterns, such as `[DT NN IN NN]`, e.g., *"age: a period of history"* and `[DT JJ NN]`, e.g., *"asset: a major benefit."* Other instances have a different type of determiner; such as cardinals (CD), as in the following definition: *"course: one of the parts of a meal."* By using the capturing groups feature in regular expressions, both `DT` and `CD` can be grouped under a determiner group, e.g., `(?P<det>\S*!!(DT|CD)?)`, which means that it announces a group called `det` that contains any word with one of the following tags (`DT` or `CD`), if they exist. For noun definitions, the rule can be straightforward, as most of the extracted patterns are similar; hence, the rule can be written

as:

```
("noun", "ˆ(?P<det>\S*!!(DT|CD)?)\s*(?P<MOD>
(\S*!!(IN|JJ|NN|DT)\s*)*)$", "X is a %s
if X is \g<det> \g<MOD>")
```

This regular expression pattern has three parts: the word tag, the definition pattern and the output sentence format. In this example, the word tag is noun, and the second part is looking for definitions that begin with or without a determiner, followed by the rest of the definition. The third part is going to print the string, where `%s` refers to the word itself and `\g<det> \g<MOD>` refers to the values for these groups. This single rule may be sufficient to cover all the noun definitions that we obtained in this experiment.

Adjectives:

In general, the extracted examples and patterns for adjective definitions begin with an adverb (RB). For instance, the most common pattern is `[RB JJ]`; followed by `[RB JJ CC JJ]`, e.g., *"appalling: very unpleasant and shocking"*; and `[RB VB]`, e.g., *"awake: not sleeping."* These definitions can be re-written in a similar way to the nouns. In adjective definitions, it is not common to have articles in their heads (at the beginning of the pattern); therefore, the most common expression pattern for adjectives is

```
("adjective", "ˆ(?P<MOD>(\S*!!(IN|JJ|NN|DT)
\s*)*)$", "X is %s if X is \g<MOD>")
```

This regular expression used to rewrite the previous example as *"X is awake if X is not sleeping."*

Verbs:

Verbs are different than the previous couple of categories as there may be different types of verbs. A verb is called an *intransitive verb* if it does not take an object, e.g., *"He ran"*, or a *transitive verb* if the verb requires an object, e.g., *"He drives a bus for living"*, and a *ditransitive verbs*, when the verb requires two objects *"Maureen gave Dan the pencil"*. All of these types of verbs exist, but the dictionary did not provide enough information about transitivity. Looking at the examples of verb definitions we obtained that contain direct objects, we conclude that there are two distinct forms of objects. An object might occur as a particular word that delivers a special meaning to the definition, e.g., *"cycle: to use a bicycle"*, as there are not many objects that can replace the word "bicycle". Another form of direct object occurs as generic objects that can take different values, as in *"afford: to provide something."* The general object word "something" can be substituted with any noun, e.g., *"afford a car"* means *"to provide a car"*. So, to make generic rules for verbs, we look for indefinite NPs with empty nouns such as *"something", "someone",* or *"somebody"*, which we replace with variables. We defined a variable called 'GENERIC' that contains all of the empty noun examples in order to re-write them as variables within the regular expression. In the output, we turn the existing 'GENERIC' into a variable (Y). Consider the following regular expression (regex) pattern:

```
("verb", "ˆ(?P<to>\S*!!TO?)\s*(?P<MV>verb)
(?P<REST1>word*?)(?P<OBJ>GENERIC) \s*
(?P<REST2>word*?)$", "X %s Y if X \g<MV>
\g<REST1> Y \g<REST2>").
```

Using this rule turn an examples such as *"assist: to help someone"* into *"X assist Y if X help Y"*. Similarly, for definitions that contain more than one generic object, we turn *"give: to pass something to someone"* into *"X give Y Z if X pass Y to Z"*.

Adverbs:

Generally, we find the extracted adverb definitions to be very short. There are many cases in which the definition contains only one word, e.g., *"commonly: usually"*. An example of a definition that contains more than one word can be seen in *"always: on every occasion."* An adverb is, generally, used to modify verbs, adjectives, or other adverbs; therefore, variables are introduced in all of their definitions. One word definitions can take the variable just after the definition, e.g., *"X commonly Y if X usually Y"*, according to this rule:

```
("adverb", "ˆ(?P<RB>\S*!!RB?)\s*$", "X %s
Y if X \g<RB> Y")
```

In the other case, in which the definition contains more than one word, the definition often begins with a preposition (IN), so we add an auxiliary verb (does) followed by the variable (Y) in front of the preposition, e.g., *"X always Y if X does Y on every occasion"*, according to the following rule:

```
("adverb", "ˆ(?P<IN>\S*!!IN?)\s*(?P<MOD>
(\S*!!(DT|JJ)\s*)*)(?P<HD>nn*)
(?P<POSTMOD>(\S*!!IN\s*nn\s*)*)$",
"X does Y %s if X does Y \g<IN> \g<MOD>
```

\g<HD> \g<POSTMOD>")

When we applied these regular expressions to the set of definitions we obtained, we were able to extract 4,613 sentences. Most of these sentences are "nouns", which is not surprising as most of the tokens for words are actually nouns. Nouns are 62.9% of the total generated sentences, followed by verbs which are 28.6%. In contrast, the adjective rules produced only 6.6% of the total sentences generated, and the lowest ranking is for adverbs, which are only 1.8% of the total output.

2.5 Transforming to Inference Rules

In this step, we are turning the sentences obtained from section [2.4] into inference rules of the form: `LHS` $\Rightarrow$ `RHS`. All the sentences generated have a form where there is a main-clause followed by an if-clause, where each clause has a shared variable; e.g., X, Y, or both. The example *"X is awake if X is not sleeping"* shows that there is a shared variable (X). If a main-clause precedes an if-clause, a reader can understand it as whenever the if-clause is true, the main-clause is true too. This is a similar notion to the Horn clause inference rules, where for each rule `A` $\Rightarrow$ `B`, whenever A is true, then so is B. Inspired by this notion, we converted all of these sentences into rules, in which each rule has an antecedent (left-hand side, LHS) and a consequence (right-hand side, RHS) in the form of `LHS` $\Rightarrow$ `RHS`. The consequences of the main-clauses occupying the rules are in the obtained sentences, while if-clauses, excluding the conditional word "if" at the beginning, represent the antecedents of the rules.

To check the validity of our process of making inference rules, we tried the same procedure on another dictionary, which is the *"WordNet Definitions"*.

3 WordNet Definitions (WDFS)

In WordNet, there is a library incorporating the definitions of most of the words it contains, and nearly all of the words that we collected from TMDC exist in WordNet. The difference is that in WordNet, the dictionary deals with the words as synsets that contain all the similar words. For instance, `wordnet.synsets("back")` returns a list of similar items, as follows:

```
[Synset('back.n.01'),
Synset('rear.n.05'),
Synset('back.n.03'),
Synset('back.n.04'),
Synset('spinal_column.n.01'),
Synset('binding.n.05'),
Synset('back.n.07'),Synset('back.n.08'),
Synset('back.n.09'),
Synset('back.v.01'),
Synset('back.v.02'), ...  etc].
```

Each of the 117,000 synsets in WordNet are linked together depending on their relational concept, and this explains why we can see synsets other than "back". As they fall in the same synset list, we extracted all the definitions for the synsets of the word "back". As the word class is essential in the rewriting process, we make our search for the synset more precise by looking for `wordnet.synsets("back", "n")` if we want nouns, "v" for verbs, and so on. We collected the words' definitions in a separate dictionary in the same way that we did for TMDC. Next, we applied the regular expressions in section [2.4] to the obtained definitions. We were able to rewrite 3,630 sentences in this collection. Amongst these sentences, nouns represent 56.6% of the domain size, while adjectives represent 31.2% and only 10.5% are adverbs. The most surprising result is that the minority were verbs, which are only 1.7% of the total domain size. The main reason for this is that our very selective regular expression rule was not suitable for the definitions in WordNet, as we expected verb definitions to begin with the word 'to' and contain generic objects.

4 Results

At this stage we have approximately 8,000 inference rules, and the accuracy of these rules must be assessed. To evaluate these rules, we took a random sample of 200 inference rules, 100 rules from each dictionary. We took into consideration the percentage that each word class represents in both dictionaries. The sample rules were uploaded to a web page and native English speakers were asked to see if these rules are correct in regards to both meaning and grammar. For each rule, there are three options: "yes" if a referee believes the rule is correct, "no" if the rule is wrong, and "skip" if the referee was not sure about the answer. As the number of rules is quite large for an online survey, we made the order of rules dynamic so that the rules with fewer answers appear first in the list. Additionally, to get precise percentages, we want to have the same number of answers on all

of the questions, so we make the question disappear when there are a certain amount of answers. In this experiment, four answers for each question was our target, and whenever a rule received 75% "yes" answers the rule is considered to be true; otherwise it is false. Of the 200 inference rules, 148 rules were judged as 'true', and from these answers we calculated the precision for each word class; they are presented in Table [2]. To calculate the precision and the recall, we used the following equations [1] and [2],respectively:

$$precision = \frac{R}{100} \quad (1)$$

$$recall = \frac{R}{100} \times \frac{M}{N} \quad (2)$$

Where N represents the number of definitions that we were looking at, and M denotes the number of definitions that were picked out as being potential rules. the constant (100) denotes the sample domain size for each dictionary and R means inference rules that were judged 'True'. Applying equation [2] for TMDC, we got 0.68 recall, and 0.126 recall for WDFS.

	TMDC	WDFS	TMDC+WDFS
Nouns	69.3%	76.7%	72.8%
Verbs	78.5%	100%	80.0%
Adj	83.3%	67.7%	70.2%
Adv	100%	72.7%	80.0%
Total	74.0%	73.0%	74.0%

Table 2: TMDC & WDFS sample's precision

The rules obtained from TMDC are slightly more precise than the ones taken from the WDFS, as they were 'True' in 74% of the cases compared to 73% in WDFS, as shown in Table [2]. There are two samples that afforded 100% precision, verbs in WDFS and adverbs in TMDC; this is because we used tight patterns for them, meaning that while we only got a small number of examples, they were very precise.

5 Conclusion

In this paper we have presented an algorithm to extract common sense knowledge. This knowledge is essential for inference systems, as it provides tools that help prove systems can derive a match between two items if they have a relationship. In this experiment, The MacMillan Dictionary and WordNet definitions were used to extract inference rules from definitions of a set of more than 5,000 important words. Even though the recall was quite low in the WDFS, it gives the impression that if this mechanism was applied to all the +155,000 WordNet definitions[3], it would generate a large number ($\approx$ 14,000) of inference rules. For future research, we are going to extract more inference rules from other dictionaries so that we can build a richer common sense knowledge base.

References

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pages 539–545.

Bjornar Larsen and Chinatsu Aone. 1999. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 16–22.

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 323–328.

Shian-Hua Lin, Chi-Sheng Shih, Meng Chang Chen, Jan-Ming Ho, Ming-Tat Ko, and Yueh-Ming Huang. 1998. Extracting classification knowledge of internet documents with mining term associations: a semantic approach. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 241–249.

Martin Rajman and Romaric Besançon. 1998. Text mining: natural language techniques and text mining applications. In *Data mining and reverse engineering*, Springer, pages 50–64.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, pages 1223–1237.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(3):203–217.

[3] source: http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html