Distinguishing Concepts and Instances in WordNet

Enrique Alfonseca and *Suresh Manandhar*

Abstract

Many lexical databases make a distinction between concepts (synsets that represent a class of things of interest) and instances (examples of concepts). However, that information is not present in WordNet. We use empirical evidence that concepts and instances are treated in different ways in language, to show that the distinction is not merely theoretical, but that it also affects how a word is used. We also describe several NLP applications that could benefit from it, and propose a criterion to annotate WordNet with that information.

1 Introduction

Many taxonomies handle two different kinds of objects. A **concept** represents a set of things of interest that have something in common; while an **instance** is a single example of a concept. For illustration, *human* is a concept that can denotate each one of the instances from the *Homo sapiens* species; while *Shakespeare* is an instance of that concept, and denotates a single instance. This distinction will be made clear in the following sections. Usually, but not always, common nouns represent concepts and proper nouns represent instances.

As far as we know, there has been little attention to the fact that WordNet contains both concepts and instances in the semantic network of nouns, with no distinction between them. However, it would be useful for many applications to know which synsets are concepts and which ones are instances, because they have different properties. Other taxonomical resources, such as Cyc [Lenat and Guha, 1990] and Ontolingua [Farquhar et al., 1997], have implemented this distinction. We have also collected experimental evidence that concepts and instances are used in different ways in language, so there are good reasons to consider them apart.

We describe here a work to enrich WordNet with information about which synsets represent instances. In section 2 we describe the original idea and its applications; section 3 deals with the manual annotation of WordNet, and section 4 with the experiments performed. Finally, section 5 lists the conclusions of our work.

1.1 Applications of our work

An important application of this work is to be able to merge WordNet with other existing ontologies, such as Cyc or ontologies developed using Knowledge Representation Systems like Ontolingua, with no loss of information. Importing and merging ontologies between systems is a task that has received much attention in the last few years, and which is important for projects such as the Semantic Net to be successful [Maedche and Staab, 2001].

Another application in which we have actually used this work [Alfonseca and Manandhar, 2002] is that of extending ontologies, such as WordNet, with new concepts. A system that identifies domain-dependent synsets and extends WordNet with them (a task called Ontology Refinement), needs to find the WordNet synset that is the hypernym of each of the learnt concepts. If instances were marked as such, then they would automatically be non-candidates to be hypernyms, because only concepts can be hypernyms. For example, if a program is analysing texts about sailing and it finds the word *pram* referring to a sailboat, it may suggest the WordNet synset *sailboat* as a possible hypernym, but it will never suggest a synset such as *Mayflower*, because it refers to an instance of a boat, and cannot have hyponyms.

For Question Answering systems, it is useful to know whether the answer is a concept or an instance. For example, the answer to a question such (1a) is an instance of *president*, and the answer to (1b) is a concept.

(1) a. Which U.S. president was the first to live in the White House?

b. What drug can be used to treat malaria?

Finally, this work can also be applied to Text and Explanation Generation, because concepts usually need to be used with a determiner, while instances need not.

2 A taxonomy with instances and concepts

We can consider the WordNet semantic network, in its current implementation (version 1.7), as a tuple $\mathcal{W} = (\mathcal{L}, \mathcal{S}, f_{\mathcal{L}}, h_{\mathcal{S}}, \mathcal{R})$ where

- $\mathcal{L}$ is the set of lexical entries (words).
- $\mathcal{S}$ is the set of synsets.
- $f_{\mathcal{L}} : \mathcal{L} \rightarrow \mathcal{S}^+$ is a function that links the lexical entries with the synsets that contain them.
- $h_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{S}^*$, called *hypernymy*, arranges the concepts and instances in a hierarchy.
- $\mathcal{R}$ is the set of non-taxonomic relations.

We argue that, inside $\mathcal{S}$, there are two different kinds of entities: concepts and instances, as defined by Degen et al. [2001] (he calls instances *individuals*, and concepts *universals*):

> Individuals belong to the realm of concrete entities, which means that they exist within the confines of space and time. Universals, in contrast, are entities that can be instantiated simultaneously by a multiplicity of different individuals that are similar in given respects. We can think of universals as patterns of features which are realized by their instances.

There is disagreement in the related literature about the interpretation of instances and concepts. One of the most widely accepted consider concepts as the sets of their instances [Montague, 1974]. Using an example from [Welty and Ferucci, 1999], *bird* would be the set containing all possible birds, real and imagined, past and future; *eagle* would be a subset of *bird*, and *the eagle Harry* would be an instance (a member) of both sets.

In First Order Predicate Logics (FOPL), instances are usually represented using constants, and concepts by using unary predicates. For example, the following sentence

(2) John saw a man

is usually translated into FOPL as

$$\exists x.saw(John, x) \wedge man(x),$$

and the concept *man* represents the set $\{x : man(x)\}$. Nevertheless, there are some frameworks in which, for simplicity, instances are also represented with predicates.

On the other hand, there are theories that consider instances are sets of universals, or sets of individualised properties, also called tropes. However, as Degen et al. [2001] points out, this interpretation poses some difficulties in dealing with different temporal profiles of the entities.

2.1 An instance by itself, or an instance of something else?

According to the definition stated above, when a synset represents features that can be realized by distinguishable instances, it is considered a concept, while when it refers to a concrete entity, or different manifestations of it, then it is an instance. But this definition still has to be further clarified.

As Welty and Ferucci [1999] notes, something can be both a concept and an instance. He proposes an example in which we have four synsets: *species*, *bird*, *Imperial eagle (Aquila eliaca)*, and *Harry*, which is an imperial eagle. Here *Aquila eliaca* is an instance of *species*, but at the same time it is a concept, one of whose instances is *Harry* (see Figure 1a).

In fact, everything can behave as an instance or a concept, depending on the interpretation. For example, *the McMillans* can be an an instance of *family* or *clan*, but it is as well a concept that includes all its members, as in sentence (3).

(3) That man is a McMillan

Or a University software might be interested in creating new entries every year for each student. In that context, *John Smith-1995* and *John Smith-1996* can be considered instances of *John Smith*, which is an instance of *student* (see Figure 1b). Summarising,

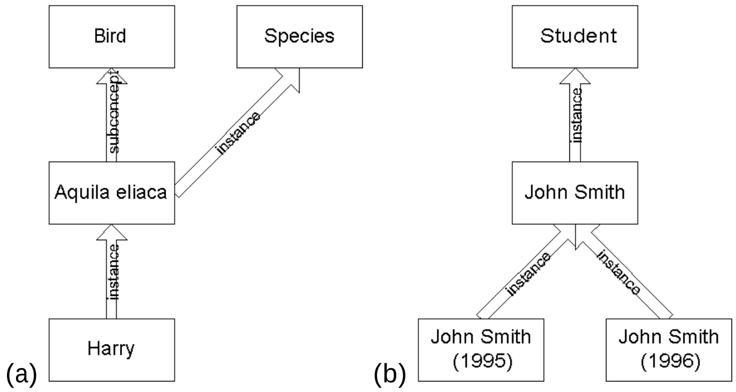

Figure 1: In (a), *Aquila eliaca* is at the same time a concept and an instance [Welty and Ferucci, 1999]. In (b), although the student *John Smith* would normally be used as an instance, in some occasions it may be useful to consider it as a concept.

- On one hand, every concept is an instance of *Concept*.
- On the other hand, every instance may be considered a concept whose instances are different manifestations of that same concept (e.g. at different times or observed by different people.)

The previous reasoning indicates that, rather than classifying synsets as instances or concepts, we should label the hypernymy links as instance links or sub-concept links instead. However, by doing this we would lose the intended objective of making WordNet more similar to other existing lexical databases or Knowledge Representation Systems, such as Cyc or Ontolingua, in which entities have to be either instances or concepts. Therefore, we have taken the following middle point:

- We classify synsets either as concepts or instances.
- If a synset has hyponyms, it is being considered as a concept in the taxonomy, and thence we mark it as such.
- Leaf synsets (synsets with no hyponyms) will be annotated either as *instances* or *concepts*, according to the relation they hold with respect to to their immediate hypernyms in the taxonomy.
- If a leaf synset has several immediate hypernyms, and it is a subconcept of at least one of them, then we shall classify it as a concept, because there is at least one interpretation in which it will have its own instances.

As we have discussed, probably this is not the optimal solution according to the theory, but we thought it is a good criterion in order to make WordNet more similar to other existing taxonomical resources.

2.2 Changes to WordNet

If $\mathcal{S}$ is the set of synsets in WordNet, and $h_{\mathcal{S}}$ is the hypernymy relationship, we propose to divide S in two subsets $\mathcal{C}$ and $\mathcal{I}$, and to modify $h_{\mathcal{S}}$ in the following way:

- $\mathcal{C}$ will be a set of concepts.
- $\mathcal{I}$ will be a set of instances.
- $\mathcal{S} = \mathcal{C} \cup \mathcal{I}$
- $h_{\mathcal{S}}$ is modified to $h_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{C}^*$.

Hence we modify the definition of $\mathcal{W}$ to include the instances: $\mathcal{W} = (\mathcal{L}, \mathcal{S}, \mathcal{I}, f_{\mathcal{L}}, h_{\mathcal{S}}, \mathcal{R})$. If we define a leaf as any synset that has no hyponyms,

$$\mathit{Leaves}(\mathcal{W}) = \{s \epsilon \mathcal{S}, \nexists n : s \epsilon h_{\mathcal{S}}(n)\}$$

then, in our framework, instances can only be leaves in the WordNet taxonomy. However, some leaves can represent concepts, if they have not been instantiated.

synset id	**synset word**	**concept-leaves**	**instance-leaves**
n00005145	person	4,534	2,913
n00018241	location	735	1,773
n00016210	psychological_feature	2,558	618
n00015211	artifact	7,494	120
n00021056	act, human_action	4,213	210
other		32,019	1,399
Total		51,553	7,033

Table 1: Results of the manual annotation of instances and concepts in WordNet.

3 Manual annotation of WordNet

In English, the instances of some concepts are rarely named. These concepts include *psychological features*, *acts*, etc. For example *the fear I felt yesterday at 12 noon* is an instance of the concept *fear*. Although it is possible to give it a name such as *My Midday Fear* or any other identifier, these kinds of entities do not usually receive a proper name in the English language. In English, concepts whose instances are usually named are *animate beings* (e.g. people, animals, even plants), *locations* (e.g. cities, etc.), *ideas and intellectual works* (e.g. theorems, books, etc.) and some objects (e.g. ships, such as *Mayflower*).

However, after examining WordNet in detail, we arrived to the conclusion that the language can contain, in theory, instances of practically every concept. A few examples of instances of unlikely entities are:

- *Creation*, meaning *God's act of bringing the universe into existence*, which is a hyponym of *action*.
- *Gettisburg's Address*, which is a speech addressed by Abraham Lincoln during the war, and is a hyponym of *speech_act*.

Therefore, we have classified by hand all leaf synsets in the nouns taxonomy, according to the relation they hold with their hypernyms.

3.1 Manual annotation results

WordNet version 1.7 contains 58,586 leaf-synsets, i.e. synsets with no hyponyms. All of them were manually annotated according to the criteria described above, and the results are displayed in Table 1. If our annotations are correct, there are 51,553 concepts and 7,033 instances among the leaves. Some of the branches with a high number of instances are person, location and psychological feature, this last branch because it includes all the mythological characters.

3.2 Annotating difficult cases

Language is always changing, and something that is considered an instance at a certain moment can, with time, come to be a concept. For example, the first *Unix* could be considered, at the moment it was released, an instance of the concept *operating system*. However, it can now be considered as any of the operating systems that have a similar architecture and a common set of commands, and that includes, among others, *solaris*, *BSD Unix*, *AIX*, *IRIX* and *Linux*. As said above, when we saw that there exists a plausible interpretation in which a synset could be considered a concept, we have classified it as such.

Other decisions were also difficult to take because the meaning of the synset had different points of view. For example, literary works such as *Genesis*, *Exodus* or *Aesop's Fables*, can be interpreted, depending on the context, in different ways, as the following sentences show. In (4a), *Genesis* refers to the text contents, the intellectual work; while in (4b) the book title refers to the physical book, and in (4c) it refers to a set of pages in a book. The theory about the same word representing different views of the same thing was developed mainly in [Pustejovsky, 1995].

(4) a. Genesis was translated to Greek.
b. Aesop's Fables looks nice on the shelf.
c. The boy tore off Genesis from his Bible.

avatar
    => Jagannath
    => Kalki
    => Krishna
    => Rama
        => Ramachandra
        => Balarama
        => Parashurama

Figure 2: *Rama* should be an instance of *avatar*, but it is also a concept which has three different instances: the three incarnations.

However, in WordNet these concepts are located under *abstraction*, not under *object*. Therefore, only the meaning in (4a) was considered to make the decision, and they were considered as instances, because the contents of the book are unique.

If, while processing a text, we found sentence (4b), then we should be able to infer that we are talking about an instance of *physical book*, whose contents are the abstraction represented by the synset *Genesis*.

Some concepts can have different names depending on their manifestations. For example, the planet *Venus* can be called *morning star* if it is visible in the early morning, or *evening star* if visible at sunset. If we were doing a full annotation of all hypernymy relationships, we could say that *Venus* is an instance of *planet*, and both *morning star* and *evening star* are instances of *Venus*. However, as we finally decided to annotate synsets and not relations, we are not able to capture this distinction. Although we only found a handful of problematic cases like this in the whole semantic network (see Figure 2), we think it is something that has to be addressed in the future.

4 Automatic annotation of WordNet

After the manual annotation, we have repeated the same procedure, but using an automatic algorithm for deciding which synsets represented instances. The motivation for this was twofold. In the first place, the automatic procedure can be used to predict whether a new domain-specific concept, not present in WordNet, is an instance or a concept without the need of human annotators. Secondly, by using very simple features, we want to show that instances and concepts, in language, are indeed used in different ways and can be easily detected, that is, there is empirical evidence that it is a difference that really exists in language.

The learning model we chose was a Maximum Entropy model [Berger et al., 1996] [Ratnaparkhi, 1998]. In this framework, the problem consists in learning a probability model

$$P_{ME}(s) = \frac{1}{Z} \cdot P_0(s) \cdot \exp\left(\sum_i \lambda_i f_i(s)\right)$$

where $P_{ME}(s)$ is the probability that s is an instance, $P_0(s)$ is an initial probability distribution, Z is a normalising constant, and f_i are binary features about the examples. Using an iterative algorithm, it is possible to obtain values for the parameters λ_i so that the model classifies the training data as best as possible. We have used the Java package `quipu.maxent`, which is freely distributed [Baldridge et al., 2001].

4.1 Features chosen

Instances, in language, have some properties in common with mass nouns. For example, they are rarely preceded by articles *the* and *a* or used in plural number. On the other hand, mass nouns can be quantified with weight, volume, etc. while instances cannot. We made use of these facts in order to choose the right features to distinguish them.

At this point, it is necessary to make a distinction. Some instance names, such as *Judas* –denoting the man that lived in Judea (sense `08885770`) have undertaken other meanings such as *someone who betrays* (sense `08201644`). These are different synsets: the first one represents an instance and, as such, cannot have an article in the specifier position; and the second one represents a concept and can be quantified or preceded by a determiner.

(5) a. Judas hung himself.
b. Don't trust him, I think he's a Judas.
c. There are several 'Judases' in that political party.

We collected the material for each synset from the Internet. We performed for every synset an automatic search on an Internet search engine, using the words in the synset and the gloss and, sometimes, hypernyms and hyponyms, and retrieved several web pages that were examined with a program. In this way, we had a different corpus for each WordNet synset from which we could extract statistics about whether the words in a synset can be used with determiners or quantifiers, or whether they can be used in plural number.

The following are several examples of features:

$f_1(s)$ = $true$ if any word in synset s was found preceded by the determiner the in the documents; $false$ otherwise

$f_2(s)$ = $true$ if no word in synset s was found with any determiner in the sample documents; $false$ otherwise

We also used capitalisation as a feature. Although not every capitalised word is an instance, many of them are, so this feature provides support in favour of considering the word an instance.

$f_3(s)$ = $true$ if any every word in synset s is capitalised; $false$ otherwise

4.2 Collection of features from Internet

Internet has been used as a corpus to collect features about the way words are used. The procedure we chose is similar to the one described in [Agirre et al., 2000]. For each WordNet synset, a query is automatically generated containing the words in that synset and its hyponyms and hypernyms as positive examples; and the words from other synsets that contain different word-senses as negative examples. For example, the Altavista query for the word *country*, with the sense 06621523:

> state, nation, country, land, commonwealth, res publica, body politic –(a politically organized body of people under a single government; "the state has elected a new president"; "African nations"; "students who had come to the nation's capitol"; "the country's largest manufacturer"; "an industrialized land")

is the following:

> "country" AND ("body politic" OR "commonwealth" OR "land" OR "nation" OR "res publica" OR "state" OR "Reich" OR "suzerain" OR "sea power" OR "great power" OR "major power" OR "power" OR "superpower" OR "world power" OR "city state" OR "ally") AND NOT ("a people"OR "area" OR "rural area")

Next, the program performs a query to the Altavista search engine[1], downloads the documents, preprocesses them and extracts the features.

4.3 Results

The training set was built with 300 concepts and 150 instances selected randomly between the WordNet leaves. We used a five-fold evaluation: the training set was divided in five subsets of the same size, each one with 60 concepts and 30 instances; in each experiment, four of them were chosen for training and the fifth for testing. The program automatically downloaded the documents and extracted the features. To measure the accuracy of this procedure, we used the manual annotation of WordNet that we had performed earlier (see the previous section).

We calculated a baseline by considering that every synset is a concept. This gives us an accuracy of $\frac{51,553}{58,586}$ = 88%.

The resulting accuracy for each of the five experiments is shown in Table 4.3. The final accuracy is 96.62%, with a mean of three mistakes in each test set. We believe this indicates that instances are indeed used in a different way in language, and they can be recognised using just a very small set of features.

[1] http://altavista.digital.com

experiment	**accuracy**
1st.	95.6%
2nd.	98.9%
3rd.	94.3%
4th.	98.9%
5th.	95.4%
Mean	96.62%
Baseline	88%

Table 2: Results of the five-fold evaluation

	Synset	**Type**		**Synset**	**Type**
(a)	hobbit, Hobbit	concept	(b)	Danaan	concept
	orc, Orc	concept		Ajax	concept
	Ent	concept		Atreus	instance
	gollum, Gollum	instance		Idomeneus	instance
	Frodo	instance		Tydeus	instance
	Bag_End	instance		Diomed	instance

Table 3: Synsets extracted from *The Lord of the Rings* (a) and *The Iliad* (b), and classification received by the maximum entropy algorithm

4.4 Analysis of the errors

By examining the erroneous classifications by hand, we noted that the major source of errors were synsets that describe languages, such as *English*, *French*, *Chinese*, etc. They had been manually annotated as *concepts*, because a language can be considered as the concept that groups all its dialects. We based this decision, as well, in the fact the the synset for the *English language* (synset number 05689601) has as many as eight hyponyms, which refer to the more common English dialects: *American English*, *cockney*, *Middle English*, etc.

However, languages are usually used without determiners, never in plural, and written capitalised, so the automatic algorithm misclassified them. The addition of the new feature

$f_4(s)$ = *true* if it is defined as a *language* or a *dialect* in the synset gloss; *false* otherwise

increased the accuracy to 97.2%.

The remaining classification errors were due to a variety of reasons: *1530s* and *1770s* were considered as plural words, and were misclassified as concepts; or *Allen_wrench* never appeared in our sample documents in plural, so it was finally mistagged as an instance. In fact, the other ten misclassifications were due to a lack of enough data, because words that can be used with determiners or in plural form never appeared like that in the small corpora downloaded from Internet.

5 Classification of new concepts

We have done another experiment by looking for new synsets in two different texts, Tolkien's *The Lord of the Rings* and Homer's *The Iliad*, some of which are displayed at Table 3. The features used are: the determiners with which they were seen in the texts; whether they have been used in plural or not; and whether they were capitalised always, sometimes or never.

In the first case, we extracted the 42 unknown words that appeared 50 or more times in the whole document, and all of them were correctly classified. The word *Gollum*, which is a character of the book, appeared written in lowercase because that character used his name as an interjection when he spoke.

In the second case, 28 unknown words were extracted. The most interesting case is that of the word *Ajax*, which was classified as a concept. In the text, we found that there are two characters in the text called Ajax, the author referred to them together quite often with the phrase "`the two Ajaxes`", so it could be considered as a concept that includes both people (Figure 3), similar to the case of *Rama* in Figure 2:

Ajax
    => Ajax son of Telamon
    => Ajax son of Oileus

Figure 3: Interpretation of the word `Ajax` found in *The Iliad*. When it refers to any person called *Ajax*, then it is a concept; while when it refers to a particular person, it is an instance
.

6 Conclusions and Future Work

We described here a work aimed at identifying concepts and instances in WordNet. We believe that this kind of information present in other lexical knowledge bases, such as Cyc [Lenat and Guha, 1990], is important with many possible uses.

As discussed in section 2, we believe that this work would be more complete if, instead of annotating synsets, we had annotated the hypernymy relationships between them. Therefore, that is an open work that can be attempted in the future.

Our experimental results show that with a very reduced set of features (capitalisation and determiners), and a rather small training set, a high accuracy can be obtained in distinguishing instances and concepts. That is a strong indication that the distinction between concepts and instances indeed exists.

This work we believe will have applications in other areas such as a question answering and ontology acquisition. Our future work will also consist in applying the results of this work to these two fields.

7 Acknowledgements

This work has been partially sponsored by CICYT, project number TIC2001-0685-C02-01.

8 Authors affiliation

Enrique Alfonseca is an assistant lecturer at the Computer Science Department, Universidad Autónoma de Madrid, and a part-time research student at the University of York. Suresh Manandhar is a lecturer at the Computer Science Department, University of York.

Contact: {enrique, suresh}@cs.york.ac.uk

References

E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the www. In *In Proceedings of the Ontology Learning Workshop, ECAI*, Berlin, Germany, 2000.

E. Alfonseca and S. Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, 2002.

Jason Baldridge, Tom Morton, and Gann Bierner. Quipu maxent, https://sourceforge.net/projects/maxent/, 2001.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

W. Degen, B. Heller, H. Herre, and B. Smith. Gol: Towards an axiomatized upper-level ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS-2001*, 2001.

Farquhar, Fikes, and Rice. *Tools for assembling modular ontologies in ontolingua*. AAAI Press, Menlo Park, California, 1997.

D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems*. Addison-Wesley, Reading (MA), USA, 1990.

A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2), 2001.

R. Montague. *Formal Philosophy*. New Haven: Yale University Press, 1974.

J. Pustejovsky. *The Generative Lexicon*. The MIT Press, Cambridge, Massachusetts, 1995.

Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. Dissertation. University of Pennsylvania, 1998.

A. C. Welty and D. A. Ferucci. Instances and classes in software engineering. *Intelligence Magazine*, 10 (2):24–28, 1999.