Usage of WordNet in Natural Language Generation

Hongyan Jing
Department of Computer Science
Columbia University
New York, NY 10027, USA
hjing@cs.columbia.edu

Abstract

WordNet has rarely been applied to natural language generation, despite of its wide application in other fields. In this paper, we address three issues in the usage of WordNet in generation: adapting a general lexicon like WordNet to a specific application domain, how the information in WordNet can be used in generation, and augmenting WordNet with other types of knowledge that are helpful for generation. We propose a three step procedure to tailor WordNet to a specific domain, and carried out experiments on a basketball corpus (1,015 game reports, 1.7MB).

1 Introduction

WordNet (Miller et al., 1990) has been successfully applied in many human language related applications, such as word sense disambiguation, information retrieval, and text categorization; yet generation is among the fields in which the application of WordNet has rarely been explored. We demonstrate in this paper that, as a rich semantic net, WordNet is indeed a valuable resource for generation. We propose a corpus based technique to adapt WordNet to a specific domain and present experiments in the basketball domain. We also discuss possible ways to use WordNet knowledge in the generation task and to augment WordNet with other types of knowledge.

In Section 2, we answer the question why WordNet is useful for generation. In Section 3, we discuss problems to be solved to successfully apply WordNet to generation. In Section 4, we present techniques to solve the problems. Finally, we present future work and conclude.

2 Why a valuable resource for generation?

WordNet is a potentially valuable resource for generation for four reasons. First, Synonym sets in WordNet (synsets) can possibly provide large amount of lexical paraphrases. One major shortcoming of current generation systems is its poor expressive capability. Usually none or very limited paraphrases are provided by a generation system due to the cost of hand-coding in the lexicon. Synsets, however, provide the possibility to generate lexical paraphrases without tedious hand-coding in individual systems. For example, for the output sentence "Jordan hit a jumper", we can generate the paraphrase "Jordan hit a jump shot" simply by replacing the word *jumper* in the sentence with its synonym *jump shot* listed in WordNet synset. Whereas, such replacements are not always appropriate. For example, *tally* and *rack up* are listed as synonyms of the word *score*, although the sentence like "Jordan scored 22 points" are common in newspaper sport reports, sentences like "Jordan tallied 22 points" or "Jordan racked up 22 points" seldomly occur. To successfully apply WordNet for paraphrasing, we need to develop techniques which can correctly identify interchangeability of synonyms in a certain context.

Secondly, as a semantic net linked by lexical relations, WordNet can be used for lexicalization in generation. Lexicalization maps the semantic concepts to be conveyed to appropriate words. Usually it is achieved by step-wise refinements based on syntactic, semantic, and pragmatic constraints while traversing a semantic net (Danlos, 1987). Currently most generation systems acquire their semantic net for lexicalization by building their own, while WordNet provides the possibility to acquire such knowledge automatically from an existing resource.

Next, WordNet ontology can be used for building domain ontology. Most current generation systems manually build their domain ontology from scratch. The process is time and labor intensive, and introduction of errors is likely. WordNet ontology has a wide coverage, so can possibly be used as a basis for building domain ontology. The problem to be solved is how to adapt it to a specific domain.

Finally, WordNet is indexed by concepts rather than merely by words makes it especially desirable for the generation task. Unlike language interpretation, generation has as inputs the semantic concepts to be conveyed and maps them to appropriate words. Thus an ideal generation lexicon should be indexed by semantic concepts rather than words. Most available linguistic resources are not suitable to use in generation directly due to their lack of mapping between concepts and words. WordNet is by far the richest and largest database among all resources that are indexed by concepts. Other relatively large and concept-based resources such as PENMAN ontology (Bateman et al., 1990) usually include only hyponymy relations compared to the rich types of lexical relations presented in WordNet.

3 Problems to be solved

Despite the above advantages, there are some problems to be solved for the application of WordNet in a generation system to be successful.

The first problem is how to adapt WordNet to a particular domain. With 121,962 unique words, 99,642 synsets, and 173,941 senses of words as of version 1.6, WordNet represents the largest publically available lexical resource to date. The wide coverage on one hand is beneficial, since as a general resource, wide coverage allows it to provide information for different applications. On the other hand, this can also be quite problematic since it is very difficult for an application to efficiently handle such a large database. Therefore, the first step towards utilizing WordNet in generation is to prune unrelated information in the general database so as to tailor it to the domain. On the other hand, domain specific knowledge that is not covered by the general database needs to be added to the database.

Once WordNet is tailored to the domain, the main problem is how to use its knowledge in the generation process. As we mentioned in section 2, WordNet can potentially benefit generation in three aspects: producing large amount of lexical paraphrases, providing the semantic net for lexicalization, and providing a basis for building domain ontology. A number of problems to be solved at this stage, including: (a)while using synset for producing paraphrases, how to determine whether two synonyms are interchangeable in a particular context? (b)while WordNet can provide the semantic net for lexicalization, the constraints to choose a particular node during lexical choice still need to be established. (c) How to use the WordNet ontology?

The last problem is relevant to augmenting WordNet with other types of information. Although WordNet is a rich lexical database, it can not contain all types of information that are needed for generation, for example, syntactic information in WordNet is weak. It is then worthwhile to investigate the possibility to combine it with other resources.

In the following section, we address the above issues in order and present our experiment results in the basketball domain.

4 Solutions

4.1 Adapting WordNet to a domain

We propose a corpus based method to automatically adapt a general resource like WordNet to a domain. Most generation systems still use hand-coded lexicons and ontologies, however, corpus based automatic techniques are in demand as natural language generation is used in more ambitious applications and large corpora in various domains are becoming available. The proposed method involves three steps of processing.

Step 1: Prune unused words and synsets

We first prune words and synsets that are listed in WordNet but not used in the domain. This is accomplished by tagging the domain corpus with part of speech information, then for each word in WordNet, if it appears in the domain corpus and its part of speech is the same as that in the corpus, the word is kept in the result, otherwise it is eliminated; for each synset in WordNet, if none of the words in the synset appears in the domain corpus, the synset as a whole is deleted. The only exception is that if a synset is the closest common ancestor of two synsets in the domain corpus, the synset is always kept in the result. The reason to keep this kind of synsets is to generalize the semantic category of verb arguments, as we illustrate in step 2. The frequency of words in such synsets will be marked zero so that they will not be used in output. Figure 1 shows two example pruning operations: (A) is a general case, and (B) is the case involving ancestor synset. In this step, words are not yet disambiguated, so all the senses of a word remain in the result; the pruning of unlikely senses is achieved in step 2, when verb argument clusters are utilized. Words that are in the corpus but not covered by WordNet are also identified in this stage, and later at step 3, we guess the meanings of these known words and place them into domain ontology.

A total of 1,015 news reports on basketball games (1.7MB, Clarinet news, 1990-1991) were collected. The frequency count reported totally 1,414 unique nouns (proper names excluded) and 993 unique verbs in the corpus. Compared to 94,473 nouns and 10,318 verbs in WordNet 1.6, only 1.5% of nouns and 9.6% of verbs are used in the domain. As we can see, this first pruning operation results in a significant reduction of entries. For the words in the domain corpus, while some words appear much more often (such as the verb *score*, which appear 3,141 times in 1,015 reports, average 3.1 times per article), some appear rarely (for example, the verb *atone* only occur once in all reports). In practical applications, low frequency words are usually not handled by a generation system, so the reduction rate should be even higher.

47 (3.3%) nouns and 22 (2.2%) verbs in the corpus are not covered by WordNet. These are domain specific words such as *layup* and *layin*. The small portion of these words shows that WordNet is an appropriate general resource to use as a basis for building domain lexicons and ontologies since it will probably cover most words in a specific domain. But the situation might be different if the domain is very specific, for example, astronomy, in which case specific technical terms which are heavily used in the domain might not be included in WordNet.

```
      { A }
     /     \             { A }
  { B }     ... ===>      /  \
    /  \               { D } ...
 { C } { F }
   /  \
{ D }{ E }

   before              after
```

(A) Synset A and D appear in the corpus, while B, C, E, and F do not.

```
    { A }
    /   \               { A }
 { B } { C }   ===>     /  \
        / \          { B }{ D }
     { D }{ E}

   before             after
```

(B) Synset B and D appear in the corpus, A, C, and E do not. Note Synset A is not removed since it's the closest ancestor of B and D.

Figure 1: Examples for corpus based pruning

Step 2. Pruning unrelevant senses using verb argument clusters

Our study in the basketball domain shows that a word is typically used uniformly in a specific domain, that is, it often has one or a few predominant senses in the domain, and for a verb, its arguments tend to be semantically close to each other and belong to a single or a few more general semantic category. In the following, we show by an example how the uniform usage of words in a domain can help to identify predominant senses and obtain semantic constraints of verb arguments.

In our basketball corpus, the verb *add* takes the following set of words as objects: (*rebound, assist, throw, shot, basket, points*). Based on the assumption that a verb typically take arguments that belong to the same semantic category, we identify the senses of each word that will keep it connected to the largest number of words in the set. For example, for the word *rebound*, only one out of its three senses are linked to other words in the set, so it is marked as the predominant sense of the word in the domain. The algorithm we used to identify the predominant senses is similar to the algorithm we introduced in (Jing et al., 1997), which identifies predominant senses of words using domain-dependent semantic classifications and WordNet. In this case, the set of arguments for a verb is considered as a semantic cluster. The algorithm can be briefly summarized as follows:

- Construct the set of arguments for a verb
- Traverse the WordNet hierarchy and locate all the possible links between senses of words in the set.
- The predominant sense of a word is the sense which has the most number of links to other words in the set.

In this example, the words (*rebound, assist, throw, shot, basket*) will be disambiguated into the sense that will make all of them fall into the same semantic subtree in WordNet hierarchy, as shown in Figure 2. The word *points*, however, does not belong to the same category and is not disambiguated. As we can see, the result is much further pruned compared to result from step 1, with 5 out of 6 words are now disambiguated into a single sense. At the mean while, we have also obtained semantic constraints on verb arguments. For this example, the object of the verb *add* can be classified into two semantic categories: either *points* or the semantic category *(accomplishment, achievement)*. The closest common ancestor *(accomplishment, achievement)* is used to generalize the semantic category of the arguments for a verb, even though the word *accomplishment* and *achievement* are not used in the domain. This explains why in step 1 pruning, synsets that are the closest common ancestor of two synsets in the domain are always kept in the result.

A simple parser is developed to extract subject, object, and the main verb of a sentence. We then ran the algorithm described above and obtained selectional constraints for frequent verbs in the domain. The results show that, for most of frequent verbs, majority of its arguments can be categorized into one or a few semantic categories, with only a small number of exceptions. Table 1 shows some frequent verbs in the domain and their selectional constraints.

```
                            {action}
                        /              \
    {accomplishment, achievement}    ...
      /        |      |      \
 {rebound} {assist} {throw} {basket}
                      |
                   {shot}
```

Figure 2: Argument cluster for the verb ''add''

WORD	FREQ	SUBJ	OBJ
score	789	player (789)	points (771) basket (18)
add	329	player	points (accomplishment) |-rebounds | throws | shots | assists - baskets
hit	237	player	(accomplishment) |-jumper | throws | shots - baskets
outscore	45	team	team
beat	11	team	team

Table 1: Selectional Constraints in Basketball Domain

Note, the existing of predominant senses for a word in a domain does not mean every occurrence of the word must have the predominant sense. For example, although the verb *hit* is used mainly in the sense as in *hitting a jumper*, *hitting a free throw* in basketball domain, sentences like “The player fell and hit the floor” do appear in the corpus, although rarely. Such usage is not represented in our generalized selectional constraints on the verb arguments due to its low frequency.

Step 3. Guessing unknown words and merging with domain specific ontologies.

The grouping of verb arguments can also help us to guess the meaning of unknown words. For example, the word *layup* is often used as the object of the verb *hit*, but is not listed in WordNet. According to selectional constraints from step 2, the object of the verb *hit* is typically in the semantic category *(accomplishment, achievement)*. Therefore, we can guess that the word *layup* is probably in the semantic category too, though we do not know exactly where in the semantic hierarchy of Figure 2 to place the word.

We discussed above how to prune WordNet, whereas the other part of work in adapting WordNet to a domain is to integrate domain-specific ontologies with pruned WordNet ontology. There are a few possible operations to do this: (1) Insertion. For example, in basketball domain, if we have an ontology adapted from WordNet by following step 1 and 2, and we also have a specific hierarchy of basketball team names, a good way to combine them is to place the hierarchy of team name under an appropriate node in WordNet hierarchy, such as the node (basketball team). (2) Replacement. For example, in medical domain, we need an ontology of medical disorders. WordNet includes some information under the node "Medical disorder", but it might not be enough to satisfy the application's need. If such information, however, can be obtained from a medical dictionary, we can then substitute the subtree on "medical disorder" in WordNet with the more complete and reliable hierarchy from a medical dictionary. (3) Merging. If WordNet and domain ontology contain information on the same topic, but knowledge from either side is incomplete, to get a better ontology, we need to combine the two. We studied ontologies in five generation systems in medical domain, telephone network planning, web log, basketball, and business domain. Generally, domain specific ontology can be easily merged with WordNet by either insertion or replacement operation.

4.2 Using the result for generation

The result we obtained after applying step 1 to step 3 of the above method is a reduced WordNet hierarchy, integrated with domain specific ontology. In addition, it is augmented with selection constraints and word frequency information acquired from corpus. Now we discuss the usage of the result for generation.

- **Lexical Paraphrases.** As we mentioned in Section 1, synsets can provide lexical paraphrases, the problem to be solved is determining which words are interchangeable in a particular context. In our result, the words that appear in a synset but are not used in the domain are eliminated by corpus analysis, so the words left in the synsets are basically all applicable to the domain. They can, however, be further distinguished by the selectional constraints. For example, if A and B are in the same synset but they have different constraints on their arguments, they are not interchangeable. Frequency can also be taken into account. A low frequency word should be avoided if there are other choices. Words left after these restrictions can be considered as interchangeable synonyms and used for paraphrasing.
- **Discrimination net for lexicalization.** The reduced WordNet hierarchy together with selectional and frequency constraints made up a discrimination net for lexicalization. The selection can be based on the generality of the words, for example, a *jumper* is a kind of *throw*. If a user wants the output to be as detailed as possible, we can say "He hit a jumper", otherwise we can say "He hit a throw."

  Selectional constraints can also be used in selecting words. For example, both the word *win* and *score* can convey the meaning of obtaining advantages, gaining points etc, and *win* is a hypernym of *score*. In the basketball domain, *win* is mainly used as win(team, game), while *score* is mainly used as score(player, points), so depending on the categories of input arguments, we can choose between *score* and *win*.

  Frequency can also be used in a way similar to the above. Although selectional constraints and frequency are useful criteria for lexical selection, there are many other constraints that can be used in a generation system for selecting words, for example, syntactic constraints, discourse, and focus etc. These constraints are usually coded in individual systems, not obtained from WordNet.

- **Domain ontology.** From step 3, we can acquire a unified ontology by integrating the pruned WordNet hierarchy with domain specific ontologies. The unified ontology can then be used by planning and lexicalization components. How different modules use the ontology is a generation issue, which we will not address in the paper.

4.3 Combining other types of knowledge for generation

Although WordNet contains rich lexical knowledge, its information on verb argument structures is relatively weak. Also, while WordNet is able to provide lexical paraphrases by its synsets, it can not provide syntactic paraphrases for generation. Other resources such as COMLEX syntax dictionary (Grishman et al., 1994) and English Verb Classes and Alternations(EVCA) (Levin, 1993) can provide verb subcategorization information and syntactic paraphrases, but they are indexed by words thus not suitable to use in generation directly.

To augment WordNet with syntactic information, we combined three other resources with WordNet: COMLEX, EVCA, and Tagged Brown Corpus. The resulting database contains not only rich lexical knowledge, but also substantial syntactic knowledge and language usage information. The combined database can be adapted to a specific domain using similar techniques as we introduced in this paper. We applied the combined lexicon to PLanDOC (McKeown et al., 1994), a practical generation system for telephone network planning. Together with a flexible architecture we designed, the lexicon is able to effectively improve the system paraphrasing power, minimize the chance of grammatical errors, and simplify the development process substantially. The detailed description of the combining process and the application of the lexicon is presented in (Jing and McKeown, 1998).

5 Future work and conclusion

In this paper, we demonstrate that WordNet is a valuable resource for generation: it can produce large amount of paraphrases, provide semantic net for lexicalization, and can be used for building domain ontologies.

The main problem we discussed is adapting WordNet to a specific domain. We propose a three step procedure based on corpus analysis to solve the problem. First, The general WordNet ontology is pruned based on a domain corpus, then verb argument clusters are used to further prune the result, and finally, the pruned WordNet hierarchy is integrated with domain specific ontology to build a unified ontology. The other problems we discussed are how WordNet knowledge can be used in generation and how to augment WordNet with other types of knowledge.

In the future, we would like to test our techniques in other domains beside basketball, and apply such techniques to practical generation systems.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IRI 96-19124, IRI 96-18797 and by a grant from Columbia University's Strategic Initiative Fund. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

J.A. Bateman, R.T. Kasper, J.D. Moore, and R.A. Whitney. 1990. A general organization of knowledge for natural language processing: the penman upper-model. Technical report, ISI, Marina del Rey, CA.

Laurence Danlos. 1987. *The Linguistic Basis of Text Generation.* Cambridge University Press.

Ralph Grishman, Catherine Macleod, and Adam Meyers. 1994. COMLEX syntax: Building a computational lexicon. In *Proceedings of COLING-94*, Kyoto, Japan.

Hongyan Jing and Kathleen McKeown. 1998. Combining multiple, large-scale resources in a reusable lexicon for natural language generation. In *To appear in the Proceedings* *of COLING-ACL'98*, University of Montreal, Montreal, Canada, August.

Hongyan Jing, Vasileios Hatzivassiloglou, Rebecca Passonneau, and Kathleen McKeown. 1997. Investigating complementary methods for verb sense pruning. In *Proceedings of ANLP'97 Lexical Semantics Workshop*, pages 58–65, Washington, D.C., April.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago, Illinois.

Kathleen McKeown, Karen Kukich, and James Shaw. 1994. Practical issues in automatic documentation generation. In *Proceedings of the Applied Natural Language Processing Conference 94*, pages 7–14, Stuttgart, Germany, October.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, **3**(4):235–312.