Towards Semantics Generation

Third stage project report

Submitted in partial fulfillment of the requirement of degree of

Master of Science

By

Ashish Almeida
Roll number: 03M05601

Under the guidance of
Prof. Pushpak Bhattacharyya

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Powai, Mumbai-76.
Acknowledgement

I would like to thank my Guide Prof. Pushpak Bhattacharyya for his scholarly guidance and useful insights. I wish to thank Dr. Rajat Kumar Mohanty with whom I had useful discussions.

Ashish Almeida
CFILT, IIT Bombay
Abstract

Analysis of natural language is difficult as each phenomenon in language needs special attention to accurately handle it. In this knowledge based approach, we attempted to solve the analysis problem for prepositional phrase and infinitival clause as a special case of to. This improves the overall analysis of language as to is frequently used lexical element in English. In addition to this, the present work took into account the empty element (implicit word), which will be a step closer to achieve ideal semantic representation.
# TABLE OF CONTENT

1 INTRODUCTION .................................................................................................................................................. 2
   1.1 STRUCTURAL AMBIGUITY .......................................................................................................................... 2
       1.1.1 Prepositional Phrase attachment ........................................................................................................ 2
       1.1.2 Clausal attachment ............................................................................................................................ 3
       1.1.3 Complexities in processing of to ....................................................................................................... 3
   1.2 RELATED WORK ........................................................................................................................................ 4
   1.3 INTRODUCTION TO UNL ..................................................................................................................... 4
       1.3.1 UNL .................................................................................................................................................. 4
   1.4 UNL SYSTEM ........................................................................................................................................... 5
       1.4.1 Analysis module ............................................................................................................................... 6
       1.4.2 Dictionary ....................................................................................................................................... 6
       1.4.3 Rule base ......................................................................................................................................... 7

2 ANALYSIS OF LEXEME TO .......................................................................................................................... 8
   2.1 INTRODUCTION ....................................................................................................................................... 8
   2.2 DETECTING THE PART OF SPEECH ....................................................................................................... 9
   2.3 ANALYSIS OF TO-PREPOSITIONAL PHRASE .................................................................................. 10
       2.3.1 Semantic relation ............................................................................................................................ 10
   2.4 ANALYSIS OF TO-INFINITIVAL CLAUSE ...................................................................................... 12
       2.4.1 UNL representation of to-infinitival clause ................................................................................... 13
       2.4.2 Attachment algorithm .................................................................................................................... 14

3 ACQUISITION OF SEMANTIC ATTRIBUTES ............................................................................................ 16
   3.1 STRUCTURE OF DICTIONARY ............................................................................................................... 16
   3.2 BUILDING SEMANTIC HIERARCHY OF ATTRIBUTES ........................................................................... 16
       3.2.1 WordNet and noun ontology .......................................................................................................... 16
   3.3 SUBCATEGORIZATION ............................................................................................................................ 18
       3.3.1 Automatic acquisition of subcategorization frame information .................................................. 18
   3.4 EXTRACTING PRO INFORMATION FROM PENN TREEBANK ........................................................................ 19

4 UNL GENERATION ...................................................................................................................................... 21
   4.1 ANALYSIS SYSTEM .................................................................................................................................. 21
       4.1.1 Look-ahead .................................................................................................................................... 22
       4.1.2 Rules for preposition-to ................................................................................................................ 22
       4.1.3 Rules for to-infinitival clauses ....................................................................................................... 23
   4.2 EVALUATION ........................................................................................................................................... 24
       4.2.1 Preparation of test sentences ....................................................................................................... 24
       4.2.2 Creation of dictionary entries ..................................................................................................... 25
       4.2.3 Testing ......................................................................................................................................... 25
   4.3 CONCLUSION ......................................................................................................................................... 26
   4.4 FUTURE WORK ...................................................................................................................................... 26

REFERENCES ................................................................................................................................................ 27

APPENDIX 1 .................................................................................................................................................... 28
1 Introduction

In this report, the problems in analysis of lexical element *to* in English are discussed. The first chapter introduces the problem and the UNL system in brief. Chapter two delves into linguistic theory related to ‘to’ and proposes the design to solve the problem. Chapter three discusses all the techniques used to enhance the power of dictionary. The last chapter discusses the implementation and testing.

1.1 Structural ambiguity

A sentence is said to be ambiguous if it can be interpreted in two or more than two ways. A structural ambiguity arises due to more than one possible structure for the sentence. Consider the example (1) which has a structural ambiguity.

1. Time flies like an arrow.

This sentence has two possible interpretations as given in (1a) and (1b). (1a) means flies named ‘time’ love an arrow. Whereas (1b) means time passes like an arrow (as fast as arrow).

1a. (Time flies) like/\text{V} (an arrow).
1b. (Time) flies/\text{V} (like/\text{P} (an arrow)).

1.1.1 Prepositional Phrase attachment

Prepositional phrase attachment ambiguity is a type of structural ambiguity. It arises due to multiple attachment possibilities of the prepositional phrase within the sentence.

![Figure 1.1: two possibilities in prepositional phrase attachment](image-url)
Consider the sentence (2). In this sentence, the prepositional phrase ‘on the Language modeling’ can attach to the noun *report* or to the verb *read*. The correct attachment is to the noun *report*. This is explained in the tree structure in figure 1.1.

2. He read the report on the Language modeling.

### 1.1.2 Clausal attachment

This is also a special case of structural ambiguity. It arises due to multiple attachment possibilities of the nested clause within the sentence. This is similar to the prepositional phrase as the ambiguity arises while attaching the clause. Consider sentence (3). In this sentence, the clause ‘that he was very happy’ has two possible sites of attachment. This clause can attach to pronoun *her* or to the verb *told*. The correct attachment is to the verb *told*. This is explained in figure 1.2.

3. Ram told her that he was very happy.

![Figure 1.2: two possibilities in clause attachment problem](figure)

### 1.1.3 Complexities in processing of *to*

*to* is the lexeme commonly used as a preposition or the infinitive marker in English. The prepositional phrase is formed by *to* followed by a noun phrase. Whereas the *to* followed by a base form of verb constitutes an infinitival clause. Consider the sentence (3) and (4) for the examples.

3. He went [to school.]_PP
4. He wants [to go.]_IP

In (3) ‘to school’ is a prepositional phrase, whereas in (4), ‘to go’ forms an infinitival clause. Given a sentence involving *to* it is difficult for machine to detect its function as...
many verbs and nouns have noun and verb senses respectively. This problem along with the prepositional phrase attachment and infinitival clause attachment problem forms the core of this report.

1.2 Related work

This work addresses the prepositional phrase attachment problem within the UNL framework. Since not many have attempted to solve this problem at the semantics extraction level, it would be difficult to compare or apply other’s work to this problem directly. However, some work done towards solving this problem in the semantic domain is mentioned below.

In the paper Application of WordNet to Prepositional Attachment [14], the author has tried to use WordNet as the main source of knowledge for preposition sense disambiguation. The prepositional attachments are classified according to the semantic equivalence of the phrase heads. In addition to the semantic relations and the synsets, it also uses the information provided by WordNet such as textual glosses. The preposition of is studied in this paper in detail. It uses synsets and hypernymy classes to group the words into the similar concepts.

Another related work, a rule-based approach to prepositional phrase attachment disambiguation [15] by Eric Brill and Philip Resnik, has used error driven prepositional phrase attachment learner. Penn Treebank is used as the training corpus of 4-tuples, of the form V-N1-P-N2. (e.g. see/V the boy/N1 on/P the hill/N2). It needs a supervised training.

In the paper Classification of Preposition semantic roles using class-based lexical associations [16], Tom and Janyce have used a corpus based training approach to classify the prepositions. They have mentioned that the semantic roles in Penn Treebank are few in number and disambiguation of the PP till the finer level of typical learner dictionary is difficult. It also mentions that the machine learning approach is better applied to produce automatic semantic relations than to resolve the PP attachment in general.

1.3 Introduction to UNL

Universal networking Language (UNL) is an interlingua for multilingual information access on the web. It is an electronic language for computers to express and exchange information. In UNL, the knowledge content of sentences is expressed in the form of semantic net like structures called UNL expressions. This chapter introduces the UNL system in the light of English language.

1.3.1 UNL

UNL consists of Universal words (UW), relations, attributes, and the UNL knowledge base (KB). The UWs constitute the vocabulary of UNL, relations and attributes contributes the syntax and the UNL KB contributes the semantics of the framework. UNL represents information sentence-by-sentence as a hyper-graph with concepts as nodes and relations as arcs. Consider sentence (5). Figure 1.3 represents the UNL for (5) graphically.
5. John wrote a letter to Mary.

In this figure, the nodes John(iof>person), Mary(iof>person), letter(icl>document) and write(icl>do) are the Universal Words (UW) corresponding to the words John, Mary, letter and wrote in (5). These are the words with restrictions in parentheses for the purpose of denoting unique sense. icl stands for inclusion and iof stands for instance of. UWs can be annotated with attributes like number, tense etc., which provide further information about how the concept is being used in the specific sentence. The arrows labeled with agt(agent), obj(object) and ins(instrument) are the relations. Each relation exactly relates two UWs. And the relations are directional. The most recent specification of the UNL contains 43 relation labels[4].

1.4 UNL system

The UNL framework consists of two parts: analysis and generation. The analysis module converts the natural language input sentence to the UNL expression and the generation module converts the UNL expression to the natural language sentence. This is clear from figure 1.4. We are mainly concerned with the analysis module for this project. The analysis module needs a dictionary and a rule-base.
1.4.1 Analysis module

The language independent analyzer, called Enconverter[9], does morphological, syntactic and semantic analysis synchronously by accessing a knowledge-rich lexicon and interpreting the Analysis Rules. It is essentially a multi-headed Turing Machine which has two kinds of heads: processing heads and context heads. The processing heads are also called Analysis Windows and are two in number: the left analysis window (LAW) and the right analysis window (RAW). The context heads are also called condition windows and there can be any number of them.

The machine has functions like shifting the windows right or left by one node, adding a node to the node-list (tape of the machine), deleting a node, exchange of nodes under processing heads, copying a node and changing the attributes of the nodes. During the analysis, whenever a UNL relation is produced between two nodes, one of these nodes is deleted from the tape and is added as a child of the other node to the tree. Forming the analysis rules for the Enconverter is equivalent to programming a sophisticated symbol processing machine.

1.4.2 Dictionary

The Enconverter is driven mainly by the dictionary. Every word that is processed by Enconverter system has to be present in the dictionary. Each entry in the dictionary consists of Headword (HW), the Universal Word (UW) and the properties of HW called attributes. For example, the lexical entries for (6a) are given in (6b).

6a. John ate rice with a spoon.

6b. [[John] {} “John(iof>person)” (N, MALE, PROPER, ANIMATE);
[ate] {} “eat(icl>do)” (V, Vol);
[rice] {} “rice(icl>food)” (N, FOOD);
[spoon] {} “spoon(icl>artifact)” (N, INSTRMNT);

The HWs John, eat, rice, spoon are enclosed in square brackets, the UWs in quotes and the attributes of the HWs in parentheses. The attributes are fairly obvious except possibly for Vol which means verb of ingestion and INSTRMNT which means instrument. The attributes are categorized as morphological, syntactic and semantic attributes.

---

1 Head word is the word of the language for which the entry is made.
2 Universal word is the concept represented using one equivalent English word and set of constraints restricting its meaning
1.4.3 Rule base

Rule base consists of rules which are used by Enconverter. The rules are made up of conditions windows, analysis windows and priority at the end. There are around 5000 rules in the rule base. The implementation in rule base involves activities such as deciding rules sequence, broadly categorizing the types of rules (look-ahead, parsing, creating relations), and deciding their priority. Consider the rule in (7).

7.  >\{ART, THE:::\}\{N, ABS:+@def::\}\{PRE, OF\}\{BLK\}\{P30;\}

The ‘>’ indicates it is right modification rule. i.e. the right analysis window is modified and the left one is deleted. \{ART, THE:::\} is the left condition window. \{N, ABS:+@def::\} is the right condition window. +@def says that add @def attribute to this node. \{PRE, OF\} and \{BLK\} are two condition windows. \{P30\} indicates priority 30. This rule is applied only when all the conditions are matched and there is no other high priority rule matching the same pattern. After application of the rule the node under left analysis window is deleted (article the) and corresponding attribute is added to the node under right analysis window.
2 Analysis of lexeme *to*

2.1 Introduction

This chapter describes the analysis of the sentences involving lexical item *to*. This lexical item poses the complexities such as, attachment site detection: a fundamental problem, and detection and resolving of empty pronominal: a to-infinitive specific problem. The lexeme *to* in an infinitival clause bring with it problem of an empty pronominal which is in the deep\(^3\) structure of the sentence. Processing of the empty pronominal is important as it helps in accurate semantic representation which leads to better accuracy in information retrieval, question answering and machine translation *etc*. For handling these issues, the exhaustive study of the properties of the lexical item *to* is done with the help of corpus[2][3][8].

Consider sentences (1) and (2).

1. Do not allow the onion to brown.
2. The lights changed from green to brown.

---

\(^3\) Deep structure: the canonical structure of the sentence in linguistics
In (1) and (2) the lexical item to is followed by brown. The phrase ‘to brown’ happens to be ambiguous for a machine, because the phrase in question is a to-prepositional phrase (PP) in (2), and, a to-infinitival clause (IP under X-bar schema [1]) in (1). The lexical element brown falls under three different lexical categories, i.e., adjective, noun and verb. The lexical element to can be a preposition-to leading to prepositional phrase attachment problem or it can be an infinitival-to leading to clause attachment problem.

### 2.2 Detecting the part of speech

Before processing to-prepositional phrase or to-infinitival phrase, it is necessary to identify its type. The first task is to detect the lexical role i.e. part of speech (POS\(^4\)) of the lexeme to.

| - to is followed by a determiner |
| - to is followed by a number |
| - to is followed by a plural noun |
| - to is followed by an adjective |
| - to is followed by a proper noun |
| - to is followed by a pronoun |
| - the matrix verb/noun specifies that it needs a to-preposition complement. |

Table 2.1: Heuristics to detect to-prepositional phrase

There are two possible POS that it can take viz. preposition and infinitive. So with the help of corpus, sentences with some frequently occurring syntactic patterns that contain lexeme to were studied. The syntactic patterns were identified from these sentences which can be used to disambiguate between the two parts of speech. Table 2.1 and table 2.2 describe those as the heuristics.

---

\(^4\) POS: part of speech is functional category of the word. E.g. noun, verb, preposition, adjective etc.
-to is followed by a base verb
-the matrix verb/noun specifies that it needs a to-infinitival complement.
-to is followed by a base verb and preceded by a noun [EVENT] that needs a to-infinitival complement.
-to is followed by adverb followed by base verb.

Table 2.2: Heuristics to detect to-infinitival clause

2.3 Analysis of to-prepositional phrase

Once the part of speech is disambiguated, the next task is to find correct attachment site for the to-prepositional phrase. In the frame \([V-NP_1\text{-}to\text{-NP}_2]\), the preposition to constitutes two problem of (i) determining to-NP\textsubscript{2} attachment site and (ii) determining the semantic relation between NP\textsubscript{2} and V/NP\textsubscript{1}. Consider sentences (3) and (4). In (3) the prepositional phrase ‘to Mary’ attaches to verb give whereas in (4) the prepositional phrase ‘to her essay’ attaches to the noun ‘amendment’.

3. John gave a flower to Mary.
4. She made several minor amendments to her essay.

<table>
<thead>
<tr>
<th>Condition 1</th>
<th>Condition 2</th>
<th>Action</th>
<th>Examples</th>
<th>UNL Relation between V/ NP\textsubscript{1} and NP\textsubscript{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument of V</td>
<td>Argument of NP\textsubscript{1}</td>
<td>NP\textsubscript{2} attaches to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Not TO</td>
<td>Not TO</td>
<td>N\textsubscript{1}</td>
<td>I caught a bus to the coast.</td>
</tr>
<tr>
<td>2</td>
<td>Not TO</td>
<td>TO</td>
<td>N\textsubscript{1}</td>
<td>She made several minor amendments to her essay.</td>
</tr>
<tr>
<td>3</td>
<td>TO</td>
<td>Not TO</td>
<td>V</td>
<td>John gave a flower to Mary.</td>
</tr>
<tr>
<td>4</td>
<td>TO</td>
<td>TO</td>
<td>N\textsubscript{1}</td>
<td>I can’t easily give an answer to your question.</td>
</tr>
</tbody>
</table>

Table 2.3: Four cases of determining attachment site for to-prepositional phrase

This algorithm uses the argument\textsuperscript{6} taking property of the verbs and nouns called the subcategorization\textsuperscript{7} frame. This property along with the proximity property makes the algorithm. It states that

---

\textsuperscript{5} Matrix verb is a verb of the main clause
\textsuperscript{6} Argument: A word/phrase is argument of certain word if it demands it in its syntactic frame.
\textsuperscript{7} Subcategorization frame: A syntactic structure consisting of the arguments of the given word.
Attach NP₂ to the word, V or NP₁ whichever demands it. But if both demands it or both does not demands it, then attach it to the nearest i.e. NP₁.

Whether the verb or noun takes an argument is mentioned along with other information in the lexicon. Depending on the presence or absence of this information in the form of attributes, four possibilities arise. They are showed in the table 2.3.

2.3.1 Semantic relation

Once the attachment site is fixed, the semantic relation can be established between the two related words. Consider the structure [X-P-N] where X can be noun or a verb, P is preposition-to and N is noun phrase. In this frame, the semantic relation between X and N depends on the functional meaning of the preposition-to and semantic properties of the head of the phrase(X) and the object of the preposition(N). Consider again the example in case 3 of table 1.3 as (5).

5. John gave a flower to Mary.

Here, Mary is related to give verb. And the relation between them is goal[4]. This is shown in figure 2.3. This relation has to be decided manually. It is idiosyncratic property of the word which selects the particular type of argument. For verbs, if the given verb falls in some semantic category or class of verbs which are syntactically and semantically similar in behavior then they select similar arguments and semantic roles. This interesting idea can be used to reduce the effort in manual work. Beth Levin’s work[5] has such categorization. It was used extensively during the analysis and deciding the UNL relations in dictionary building.

![Figure 2.3: The UNL graph for the sentence ‘John gave a flower to Mary’](image)

The UNL expression for the sentence in (5) is given below.

```
gtt (give(icl>do), @entry.@past, John(iof>person))
obj (give(icl>do), @entry.@past, flower(icl>flora))
gol (give(icl>do), @entry.@past, Mary(iof>person))
```
2.4 Analysis of to-Infinitival Clause

In the sentences with to-infinitival clause, the subject of the embedded clause is sometimes dropped. The subject which is not realized phonetically and morphologically is indicated as PRO. Consider sentence (6).

6. He promised me [to come for the party].

In this sentence, [to come for the party] is an infinitival clause. But the subject of this infinitive clause is missing. The reading of (6) suggests that the subject is he. i.e. same as subject of the main sentence. This dropped subject is indicated as PRO as shown in (7).

7. He promised me [PRO to come for the party].

Now, consider the sentence (8).

8. He ordered us [to finish the work].

Here, it can be discerned that the missing subject of the embedded clause is us which is also the object of the main sentence. As shown in (9).

9. He ordered us [PRO to finish the work].

It can be observed from (8),(9) that the PRO can be co-indexed with subject of the main sentence or object of the main sentence. Thus depending on the site of co-indexing PRO is said to be subject controlled or object controlled[11]. Also, it is also observed that the subject controlling or object controlling is the idiosyncratic property of verb of the main sentence. It is also noteworthy that [V-N-to-V] frame does not necessarily have a PRO. Consider the example (10). In this sentence the subject of the infinitival clause is John; i.e. the subject is overtly present for the to-infinitival clause and it is not the argument of the verb want of the main sentence. This reading is shown in (11) with explicit bracketing. Since this type of behavior is not common, it is not considered in the analysis.

10. I want John to do this work.
   11. I want [John to do this work]

<table>
<thead>
<tr>
<th>Verb type</th>
<th>PRO controlled by</th>
<th>Frame</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Promise</td>
<td>Subject</td>
<td>V-to-V</td>
<td>They promised [PRO to give a party].</td>
</tr>
<tr>
<td>2 Force</td>
<td>Object</td>
<td>V-N-to-V</td>
<td>They forced Mary [PRO, to come early].</td>
</tr>
<tr>
<td>3 Try</td>
<td>Subject</td>
<td>V-to-V</td>
<td>They tried [PRO, to give a party].</td>
</tr>
<tr>
<td>4 Want</td>
<td>Object</td>
<td>V-N-to-V</td>
<td>I want a pen [PRO, to write a letter].</td>
</tr>
<tr>
<td></td>
<td>Subject</td>
<td>V-to-V</td>
<td>I want [PRO, to write a letter].</td>
</tr>
</tbody>
</table>

Table 2.4: verb-types which has to-infinitival clause as an Argument

In case of to-infinitives which are attached to a noun, have PRO which is not controlled by any word in the sentence. Consider the example in (12). Such PRO is classified as arbitrary PRO. But this type is not discussed further, to constrain the scope of the problem.

12. He makes a real effort to control his emotions.
In certain category of verbs like *promise* as in (9), it selects *to*-infinitival clause with PRO co-indexed with the subject of the main verb. This PRO is said to be subject controlled. In other category of verbs like *order* as in (8), it selects *to*-infinitival clause with PRO co-indexed with the object of the main sentence. This PRO is said to be object controlled. Depending on the syntactic frame and the property of subject/object controlled-ness of PRO, verbs are classified into four types as seen in table 2.4. This property is also incorporated into the lexicon.

### 2.4.1 UNL representation of to-infinitival clause

The UNL has the notion of representing compound concepts: a concept composed of concepts and relations(compound UW\(^8\)). Since the *to*-infinitive clause is conceptualized as IP(inflectional phrase)\(^9\) in X-bar schema, it is represented as a compound UW in UNL. The PRO is the visualized as UW which is referred doubly i.e. in the main sentence and in the clause. Consider the example (13).

13. They promised Mary, [PRO, to give a party].

The graphical representation for (13) along with the UNL expression is given in figure 2.4. The UW *they* is related to *promise* as well as to UW *give*. i.e. *they* is co-referred by *promise* as well as *give*. This co-referencing is reflected in the UNL in a way that the UW *they* is used in relations involving both *give(icl>do)* and *promise(icl>do)*. As seen in figure 2.4, the *to*-infinitival clause is represented as single concept indicated by large eclipse with label :01 and then related to *promise(icl>do)* through *obj* (object) relation.

![Figure 2.4: The proposed UNL graph for the sentence 'They forced Mary to give a party'](image)

The UNL expression equivalent to the graph shown in figure 2.4 is given below.

\(^8\) Compound UW is the set of relations that collectively represents unique concept.

\(^9\) IP is phrase which indicates the sentence with a tense information.
agt (promise(icl>do).@entry.@past, they:0A)
gol (promise(icl>do).@entry.@past, Mary(iof>person))
obj (promise(icl>do), :01)
agt :01(give(icl>do), they:0A)
obj :01(give(icl>do), party(icl>function))

2.4.2 Attachment algorithm

If the syntax of the sentence is considered, then in the frame \([V_1-N-to-V_2]\), there exists a syntactic level clause (to-infinitive) attachment ambiguity. That is, if we consider (13) there are to possibilities of attachment. The to-infinitive can attach to the verb promise or to the noun Mary. The nature of this ambiguity is exactly same as the prepositional phrase attachment ambiguity if the frame \([V_1-N-to-V_2]\) is considered. Thus, the similar strategy is used to solve the to-infinitive clause attachment ambiguity. Considering the fact that to-infinitival clause has more affinity to attach to a verb rather than to a nominal concept, in the 4\(^{th}\) case the attachment preference is given to the \(V_1\) rather than to NP. The modified table is presented as table 2.5.

<table>
<thead>
<tr>
<th>Condition 1</th>
<th>Condition 2</th>
<th>Action</th>
<th>Examples</th>
<th>UNL Relation between (V/\text{NP}_1) and to-inf clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument of (V_1)</td>
<td>Argument of NP</td>
<td>(V_2) attaches to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TO</td>
<td>TO</td>
<td>N</td>
<td>He urged a bipartisan effort to resolve the long standing issue.</td>
</tr>
<tr>
<td>2</td>
<td>TO</td>
<td>Not TO</td>
<td>(V_1)</td>
<td>John ordered Mary to leave.</td>
</tr>
<tr>
<td>3</td>
<td>Not TO</td>
<td>TO</td>
<td>N</td>
<td>A mild form of exercise will increase your ability to relax.</td>
</tr>
<tr>
<td>4</td>
<td>Not TO</td>
<td>Not TO</td>
<td>(V_1)</td>
<td>They gathered a significant crowd to battle for their demand.</td>
</tr>
</tbody>
</table>

*Table 2.5: Four case of to-infinitive clause attachment*

Figure 2.6 shows the sequence of steps involved in the processing of the sentence involving lexeme-to. Once the part of speech is identified the processing of both the prepositional phrase and infinitival phrase is done separately.

The next chapter discusses the practical problem of acquiring different syntactic and semantic attributes through various resources to enrich the dictionary.
Figure 2.6: The overall sequence of steps required for analyzing sentences to UNL.
3 Acquisition of semantic attributes

As it is mentioned in the first chapter, the analysis is based on the various attributes that are assumed to be in the dictionary. But creation of such a dictionary with semantic information is also a challenging job. As seen in the previous chapter, the syntactic and semantic information like argument structure information is also need to be present in the dictionary. This chapter discusses how the organization of attributes is done and how the dictionary is enriched with more attributes.

3.1 Structure of Dictionary

In addition to this, the dictionary entry also needs attributes which provides Building dictionary with manual efforts is cumbersome and time-consuming. Different attempts have been made to acquire the information from various resources available in public domain. Also, an attempt is made to organize the attributes in the form of ontology using WordNet.

3.2 Building semantic hierarchy of attributes

Each dictionary entry is furnished with the set of semantic attributes which define the concept represented by it. To find the semantic relations between two syntactically related words, it is necessary to know the meaning of the connecting words i.e. the set of attributes associated with both the connecting words. This meaning is encoded in the dictionary in the form of semantic attributes. For instance, consider the word `banana` in (1). Its attributes suggests it is an inanimate physical thing, a fruit and an edible thing.

1. `{banana}{}"banana\{icl>fruit\}"(N,PHSCL,INANI,FRUIT,EDBL)<h,0,0>;

These attributes provides generic to specific information about the word’s meaning. So it is discernible from this fact that the semantic features needs some hierarchical organization through which the appropriate features can be picked up for the given word. If not, it would become complicated and error prone to coin and choose attributes.

3.2.1 WordNet and noun ontology

WordNet[6] is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical
concept. Different relations like hypernymy, meronymy, entailment, holonymy link the synonym sets.

The WordNet, repository of concepts(synset) and relations between them has underlying ontology respectively for nominal concepts, verbal concepts, adjectival concepts and adverbial concepts. The noun ontology is deepest and comprehensive. abstraction, state, event, group etc. are the top level concepts in noun ontology. Any given word, i.e. concept in WordNet is related to one or more than one such top level node.

Figure 3.1 depicts snippet of noun ontology for the word water. Here, one can observe that water is connected to entity: the top level concept in the noun hierarchy. The notable thing about ontology is that one concept can have more than one ancestor at any given level. In the figure 3.1, \{water\} is linked to a \{binary compound\} as well as \{liquid\}. \{Fluid\}, the ancestor of \{liquid\} and \{chemical compound\}, the ancestor of \{binary compound\} are both related to \{substance, matter\} concept.

We have used this ontology to organize the semantic attributes in our system. The hypernymy relation is the one which gives a superordinate-subordinate relationship between parent and its child node. Figure 3.2 shows part of the top 2 levels in nominal concepts category. The complete hierarchy is given in appendix1. Each line in the figure 3.2 corresponds to a concept (synset) in the language. The relationship from parent to child is hyponymy relation and from child to parent is hypernymy relation. Besides hypernymy & hyponymy-relationship there is other ontology based on meronomy-holonomy relationship.
3.3 Subcategorization

Consider the dictionary entry for the word give considering the following usage in (3).

2. He gave a gift to her.

In (2), give takes one NP as its first argument and a prepositional phrase as its second argument. This is specified in the lexicon through the attribute \#_TO_A2. Additionally, the UNL relation is specified \#_TO_A2_GOL. Thus dictionary entry for gave will be as given in (3).

3. \[\text{gave}\] "give(icl>do)"(VRB,VOA,VOA-PHSCI,#_TO_A2,#_TO_A2_GOL,PAST) <E,0,0>;

In the similar fashion, nouns and adjectives are given the attributes indicating the subcategorization[12].

3.3.1 Automatic acquisition of subcategorization frame information

The subcategorization information for the words is gathered through two lexical resources: WordNet[6] and Oxford advanced learner’s dictionary(OALD)[7]. Both the resources are widely used for many NLP related tasks. OALD provides the partial syntactic frames against each word. These frames were mapped to UW attributes indicating the preposition phrase and infinitival clause arguments. For example, consider the entry for the noun effort in OALD as shown in figure 3.3. The meaning specifies that effort takes ‘to do sth’ type of frame which says that the noun effort takes to-infinitival phrase as its argument.

Consider one instance of verb i.e. force. Figure 3.4 shows the part of the entry for the verb force from OALD. One of the usages of it describes the sentence frame ‘[VN to
The frame says that force takes to-infinitival clause as its second argument and noun as its first argument.

<table>
<thead>
<tr>
<th>force verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>make sb do sth</td>
</tr>
<tr>
<td>1 [often passive] ~ sb (into sth / into doing sth) to make sb do sth that they do not want to do</td>
</tr>
<tr>
<td>SYN COMPEL [VN, VN to inf] The President was forced into resigning.</td>
</tr>
<tr>
<td>• The President was forced to resign.</td>
</tr>
<tr>
<td>• [VN to inf] I was forced to take a taxi because the last bus had left.</td>
</tr>
<tr>
<td>• She forced herself to be polite to them.</td>
</tr>
<tr>
<td>• [VN] He didn't force me—I wanted to go.</td>
</tr>
<tr>
<td>• Ill health forced him into early retirement.</td>
</tr>
<tr>
<td>• (spoken, humorous) ‘I shouldn’t really have any more.’ ‘Go on—force yourself!’</td>
</tr>
<tr>
<td>• Public pressure managed to force a change in the government’s position.</td>
</tr>
</tbody>
</table>

Figure 3.4: The part of OALD entry for the verb ‘force’

The WordNet provides sentence frames against each sense of the verb, which also includes example sentence particular to the verb. These sentences are actually prepared with the fixed number of template sentences. Sentence in (4) shows one such template sentence for verb promise with a placeholder for the verb.

4. They ____ him to write the letter.

From all the sentence frames, those sentence frames which have a to-infinitival clause are identified. Then, for given verb it can be easily decided if it takes to-infinitival clause as an argument or not. The frames also specify if the argument is first argument of the verb or second.

3.4 Extracting PRO information from Penn Treebank

As discussed in analysis, having Subject-controlled PRO or Object-controlled PRO is the idiosyncratic property of the verb of the matrix clause. According to our knowledge, at present no lexical database records this kind of information in any explicit form.

This section explains how a syntactically annotated corpus can be used to extract such semantic information. Penn Treebank[8] is syntactic annotated corpus. It gives best parse for each sentence along with providing consistent treatment to many linguistics phenomenon such as subject movement, empty and null elements, logical subject of passive constructs, subject marking for infinitival clauses, in the form of functional tags and semantic labels.

A typical sentence involving to-infinitival clause from Penn Treebank is shown in figure 3.5. It shows parse structure with functional tags NP, VP, S etc. and semantic tags
SUB(subject), DIR(direction). The *-1 in ‘NP-SBJ *-I’ is the trace\(^{10}\) of the subject of the to-infinitival clause and it is co-indexed with NP-SBJ-I which is the noun investors.

```
( (S
   (NP-SBJ-1 investors)
   (VP continue
      (S (NP-SBJ) *-1)
      (VP to
        (VP pour
          (NP cash)
          (PP-DIR into
            (NP money funds))))))
```

*Figure 3.5: A sentence from Penn Treebank showing parse structure*

Algorithm to extract PRO is described as follows. This information is accumulated in the repository from where ultimately the attributes for dictionary are picked up.

```
For each sentence,
   1. If the sentence contains to under VP node (i.e. to-infinitive)
   2. Find its subject NP, if it is not empty continue with next sentence.
   3. Find the verb to which to-infinitival clause is attached.
   4. a. Find its subject NP.
      b. If it is referred by same index as that of the empty element in step 2,
         mark the verb as having 'Subject Controlled PRO'.
   5. a. Else, find the object NP for the verb in step 3. (This step will detect and
         consider passive appropriately)
      b. If it is 'referred' by same index as that of the empty element in step 2,
         mark the verb as having 'Object Controlled PRO'.
   6. Find the root of the inflected verb with the help of WordNet.
   7. End.
```

*Figure 3.6: Algorithm to extract PRO information from the Penn tree-bank database.*

\(^{10}\)Trace: It is an empty position created after movement of certain word from canonical form of a sentence. Trace theory assumes that if an element X has been moved in the course of a derivation, it has left a trace in its original position.
4 UNL Generation

In this chapter, the implementation of rules is discussed focusing the phenomenon discussed in chapter one. The first section explains the overall structure of the system.

4.1 Analysis System

![Analysis module diagram]

The UNL framework consists of two parts: analysis and generation. The analysis module converts the natural language input sentence to the UNL expression and the generation module converts the UNL expression to the natural language sentence. Figure 3.1 shows the constituents of the analysis module. The Enconverter is a language independent analyzer. It processes input sentence to produce UNL expression. It uses the dictionary and the rule base of the particular language (i.e. English) to accomplish the required task. Enconversion system consists of three kinds of rules in the context of the problem.

1. Detecting preposition-*to* from infinitival-*to*.
2. Finding the correct attachment site.
3. Establishing appropriate semantic relation.

Besides this, implementation of PRO involves three steps which are discussed in section 4.1.3 in detail.
4.1.1 Look-ahead

Primarily, look-ahead rules are required to distinguish between preposition-to and infinitival-to. This work involves disambiguation of lexical element to. In Enconverter it is done with the help head-movement (look ahead) [9] rules. A few actions are also encoded within the other type of rules. In (1) a rule demonstrates how the heuristics in the table 1.1 is transformed into the rules.

1. :{:::}{{^TO_INF_NEXT: +TO_INF_NEXT::}( #TO, TO_INF)( BLK)( VRB, V_1)P40;

This rule states that

IF

The left analysis window (indicated by {}) is on any word

AND

The right analysis window is on a word which does not have a \textit{TO-INF\_NEXT} (indicating absence (^) of that property) \textit{i.e.} look ahead is not performed yet.

THEN

Select the next sequence of words such that they will satisfy the conditions as -

- Pick the word \textit{to} corresponding infinitival-to (indicated by attributes \#TO and \textit{TO-INF})

AND

- Pick a space (indicated as \textit{BLK})

AND

- Pick a verb which is in its simple form (indicated by \textit{V_1})

AND

- Add the property \textit{TO-INF\_NEXT} to the word in the right analysis window (indicated by +)

The priority of this rule is 40 which should be between 0 (lowest) and 255 (highest). The priority is used in case of rule conflict.

4.1.2 Rules for preposition-to

Let us consider an example of a rule to resolve attachment. The rule in (2) decides when to shift right to take care of case 4 in Table 1.3 for prepositional phrase attachment.

2. R\{VRB, \_TO\_AR2:::\}{N, \_TO:::}( PRE, \#TO)P60;

This states that

IF

- the left analysis window is on a \textit{VRB} which takes a \textit{to-pp} as the second argument (indicated by \_\_\_TO\_AR2)

AND

- the right analysis window is on a \textit{NOUN} which takes a \textit{to-pp} as an argument (indicated by \_\_\_TO)

AND

- the preposition-to follows the noun (indicated by \textit{(PRE,\#TO)})

THEN
Shift right (indicated by $R$ at the start of the rule) by one word anticipating noun attachment to the to-pp.

Taking another example, where a UNL relation is created, the rule in (3) sets up $gol$ (goal) relation between $V$ and $NP_2$ and deletes the node corresponding to $NP_2$.

3. $\langle$VRB, #_TO_AR2, #_TO_AR2_gol:::}{N,TORES,PRERES::gol::}P25;

This states that

IF

the left analysis window is on a verb which takes a to-pp as the second argument which should be linked with the $gol$ relation (indicated by #_TO_AR2_gol)

AND

the right analysis window is on a noun for which the preceding preposition has been processed and deleted. (indicated by TORES, PRERES)

THEN

set up the $gol$ relation between $V$ and $N_2$.

The above is relation-setting rule as indicated by $<$ at the start of the rule.

4.1.3 Rules for to-infinitive clauses

The steps taken for the attachment and producing a relation for to-preposition clause are same as explained in context of to-preposition. The difference lies in generation of PRO and generation of compound UW for the embedded clause.

Before resolving the relation between the infinitive clause and the matrix verb, the empty pronominal has to be generated and linked to the verb within the clause. *i.e.* the complete processing of PRO involves 3 steps as follows.

1. Produce a PRO element in UNL with appropriate relation. (Enconverter)
2. Relate it to the verb of the infinitive clause semantically. (Enconverter)
3. Substitute a referred UW in the place of PRO. (Post editor)

The first two steps are carried out with the help of Enconverter. This is explained with rules for each step in (4) and (5) respectively.

4. $\langle$VRB, SUB_PRO:::)$[[SUB_PRO]}::N, SUB_PRO, #INSERTED::}{VRB,TO_INFRES,^PRORES}P30;

The rule in (4), inserts a $SUB_PRO$ (property that indicates subject controlled PRO) between matrix verb and verb of to-infinitive clause.

This states that

IF

the left analysis window is on a verb which has a $SUB_PRO$ property

AND

the right analysis window is on a verb which is predicate of the to-infinitive clause (lexical element to is already deleted during analysis)

THEN

insert the node with UW $SUB_PRO$ between left and right analysis window (indicated by "$\langle\$"). Also add noun($N$), and subject PRO($SUB_PRO$) attributes to this new node.
The rule in (5), creates a relation between the PRO (indicated by N, SUB_PRO) and the verb of the embedded to-infinitive clause (indicated by VRB, VOA).

5. \(> (VRB) \{N, SUB_PRO::agt:\} \{VRB, VOA, TO_INFRES::prores, +SUB_PRORES::\}P40;\)

4.1.3.1 Post-processing

After the Enconverter produces UNL, the PRO is actually replaced with the appropriate UW in the UNL (the UW which is co-indexed with the PRO). The algorithm for the same is shown in figure 3.2.

For the given UNL expression,
1. Search for UNL relation that contains one of the UWs named SUB_PRO or OBJ_PRO.
2. Let its scope id be i.
3. Find the verbal UW at the upper scope level that is related with the compound UW (having scope id i).
4. In case of SUB_PRO
   a. Look for the second UW in the agt or aoj relation along with verbal UW (from step 3)
   b. Replace SUB_PRO with the UW found in (3.a)
5. In case of OBJ_PRO
   a. Look for the second UW in the obj relation along with verbal UW (from step 2)
   b. Replace OBJ_PRO with the UW found in (4.a)
6. End.

Figure 4.2: Algorithm to replace PRO with actual UW.

It must be noted that this implementation only detects and produces correct referencing for UWs where the ‘referenced UW’ is subject or object of the matrix clause as discussed in analysis.

4.2 Evaluation

This section describes the creation of test-data, preparation of dictionary entries, and UNL generation process. It also discusses the results and the errors.

4.2.1 Preparation of test sentences

The testing is done on the sentences obtained from Penn tree-bank corpus[8], Oxford advanced learner’s dictionary[7] (OALD). Though all the simple, complex and compound sentences were analyzed for the creation of rule base, testing is done only on simple sentences. Because processing of complex and compound sentences involves handling of constructs which are at present, beyond the scope of Enconverter. All compound sentence i.e. the sentences which contains two independent clauses joined by a coordinator are not considered. All complex sentences except those involving to infinitive clauses are also not considered. For instance, if...then, because..., ...so constructs are filtered out. Also only declarative sentences are selected, all other sentences like
interrogative, exclamatory sentences, sentences with punctuations within sentence like comma, semicolon, and colon were also filtered out.

4.2.2 Creation of dictionary entries

The dictionary was prepared for all the sentences selected as above. Document specific dictionary generator [13] is used to generate the test dictionary. It is an automated tool for generating English-UW dictionary using WordNet. It assumes that only one sense of the word is present in the document. This generated dictionary has mistakes which need to be corrected. For instance, it contains mistakes in headwords, UW restrictions (i.e. sense), and attributes (i.e. problem of underspecified or overspecified attributes, irrelevant attributes, wrong stem). Besides, some of the missing senses of some words, proper nouns are added at the end.

After completing the correction work, the extra semantic attributes are appended to the appropriate dictionary entries. These are the attributes which are discussed in chapter three. They provide syntacto-semantic information such as argument structure information, semantic relation for arguments, type of pro for to-infinite clauses. This is done with the help of a script which uses the dictionary and repository of word with the extra features.

4.2.3 Testing

Once the dictionary for content words is ready, it is appended with the functional words dictionary to form a complete dictionary. The sentences were then run through Enconverter which has rules as discussed in chapter 3. The result for 200 sentences tested is summarized in the table 4.1.

<table>
<thead>
<tr>
<th>To</th>
<th>Preposition sense</th>
<th>Infinitive sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of sentences(200)</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Number of sentences where correct sense of to is detected</td>
<td>97</td>
<td>93</td>
</tr>
<tr>
<td>Number of sentences with correct attachment/UNL</td>
<td>80</td>
<td>72</td>
</tr>
</tbody>
</table>

Table 4.1: The results from testing of sentences with lexeme to

4.2.3.1 Result analysis

The errors are because of the dictionary (the inconsistent attributes) and the sentences or because of the inadequacy of the rules. The following examples will illustrate the various factors.

Consider sentence (6), which demonstrates wrong POS tag identification. Here, the verb sense of gamble is picked up from dictionary giving wrong POS for to too. It treats the word to as to-infinitival marker instead of preposition to. In (7), the noun refund is not the object of the matrix verb order but it is the subject of infinitive clause as given in (7a). The system treats it as an object of verb order which leads to wrong UNL generation.

6. He turned to gambling.
7. Judge ordered the refunds to begin in May.
7a. Judge ordered [the refunds to begin in May.]

In (8), the PRO (referred to ‘the industry’ as shown in (8a)) should relate to verb continue with aoj relation. But instead, the system produces the default agt relation between the two because of the inadequacy of the rules to handle aoj relation in infinitive clause. This leads to wrong UNL generation.

8. Government helped the industry to continue.
8a. Government helped [the industry] [PRO to continue].

Constructions like (9) fell as these types of sentences are not handled by the system. These types of constructions involve two verbs in sequence and the second verb act as matrix verb (raising verbs).

9. She felt moved to address the crowd.

10. They signed an agreement to acquire additional revenue.
10a. They signed [an agreement [PRO to acquire additional revenue.]]

In (10), there is an error in attachment. Ideally, the infinitival clause should attach to noun agreement and not to verb sign. But since the noun agreement does not specify a to-infinitival clause as an argument, the to-infinitival clause gets attached to verb sign and not to the noun agreement.

4.3 Conclusion

The automatic acquisition of dictionary attributes is the effective idea of creating and enriching dictionary, as manually creating dictionary is slow and prone to error. From this work, it becomes evident that the detailed analysis when incorporated into the system improves the accuracy.

4.4 Future work

The work done till now is for six most common prepositions and the to-infinitival ‘to’. This can be scaled in the same way for other prepositions. One of the main hindrances in this task is creation of dictionary rich with syntacto-semantic attributes. The dictionary creation is still a bottleneck in the system. This work is a significant progress in that direction. However, dictionary enhancement work needs to be automatized further, without degrading the quality.
References

16. Hara T and Wiebe J.: Classification of Preposition semantic roles using class-based lexical associations. New Mexico State University, NM.
### Appendix 1

The following hierarchy is the top 2 levels in WordNet 2.0 noun ontology.

<table>
<thead>
<tr>
<th>abstraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute</td>
</tr>
<tr>
<td>measure, quantity, amount</td>
</tr>
<tr>
<td>relation</td>
</tr>
<tr>
<td>set</td>
</tr>
<tr>
<td>space</td>
</tr>
<tr>
<td>time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>act, human action, human activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
</tr>
<tr>
<td>activity</td>
</tr>
<tr>
<td>assumption</td>
</tr>
<tr>
<td>communication, communicating</td>
</tr>
<tr>
<td>distribution</td>
</tr>
<tr>
<td>group action</td>
</tr>
<tr>
<td>hindrance, interference</td>
</tr>
<tr>
<td>inactivity</td>
</tr>
<tr>
<td>judgment, judgement, assessment</td>
</tr>
<tr>
<td>nonaccomplishment, nonachievement</td>
</tr>
<tr>
<td>production</td>
</tr>
<tr>
<td>rejection</td>
</tr>
<tr>
<td>residency, residence, abidance</td>
</tr>
<tr>
<td>speech act</td>
</tr>
<tr>
<td>stay</td>
</tr>
<tr>
<td>stop, stoppage</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>causal agent, cause, causal agency</td>
</tr>
<tr>
<td>enclosure, natural enclosure</td>
</tr>
<tr>
<td>expanse</td>
</tr>
<tr>
<td>location</td>
</tr>
<tr>
<td>object, physical object</td>
</tr>
<tr>
<td>sky</td>
</tr>
<tr>
<td>substance, matter</td>
</tr>
<tr>
<td>thing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>event</th>
</tr>
</thead>
<tbody>
<tr>
<td>group action</td>
</tr>
<tr>
<td>happening, occurrence, natural event</td>
</tr>
<tr>
<td>miracle</td>
</tr>
<tr>
<td>social event</td>
</tr>
</tbody>
</table>

**group, grouping**
- arrangement
- association
- biological group
- citizenry, people
- collection, aggregation, accumulation, assemblage
- community, biotic community
- ethnic group, ethnos
- halogen
- kingdom
- multitude, masses, mass, hoi polloi, people
- people
- population
- race
- rare earth, rare-earth element, lanthanoid, lanthanide, lanthanon
- social group
- subgroup
- system, scheme

**phenomenon**
- consequence, effect, outcome, result, event, issue, upshot
- luck, fortune
- luck, fortune, chance, hazard
- natural phenomenon
- process

**possession**
- assets
- liabilities
- property, belongings, holding, material possession
- transferred property, transferred possession

**psychological feature**
- cognition, knowledge, noesis
- feeling
- motivation, motive, need

**state**
- action, activity, activeness
- being, beingness, existence
- cognitive state, state of mind
- condition
- condition, status
- conflict
- damnation, eternal damnation
- death
- degree, level, stage, point
- dependence, dependance, dependency
disorder
freedom
hostility, enmity, antagonism
illumination
immaturity, immaturity
imperfection, imperfectness
inaction, inactivity, inactiveness
integrity, unity, wholeness
maturity, matureness
merchantability
motion
motionlessness, stillness
nonbeing
obligation
office, power
omnipotence
omniscience
order
perfection, flawlessness, ne plus ultra
physiological state, physiological condition
readiness, preparedness, preparation
relationship
representation, delegacy, agency
situation, state of affairs
skillfulness
status, position
temporary state
union, unification
wild, natural state, state of nature