Semantically Relatable Sets: Building Blocks for Representing Semantics

Rajat Kumar Mohanty
Dept. of Computer Science and Engineering
Indian Institute of Technology Bombay
rkm@cse.iitb.ac.in

Anupama Dutta
Dept. of Computer Science and Engineering
Indian Institute of Technology Bombay
anupama@cse.iitb.ac.in

Pushpak Bhattacharyya
Dept. of Computer Science and Engineering
Indian Institute of Technology Bombay
pb@cse.iitb.ac.in

Abstract

Motivated by the fact that *ultimately, automatic language analysis is constituent detection and attachment resolution*, we present our work on the problem of generating and linking *semantically relatable sets (SRS)* as a via media to automatic sentence analysis leading to semantics extraction. These sets are of the form <*entity*$_1$ *entity*$_2$> or <*entity*$_1$ *function-word entity*$_2$> *or <function-word entity*>, where the *entities* can be single words or more complex sentence parts (such as an embedded clause). The challenge lies in finding the components of these sets, which involves solving prepositional phrase (PP) and clause attachment problems, and empty pronominal (PRO) determination. Use is made of (i) the parse tree of the sentence, (ii) the subcategorization frames of lexical items, (iii) the lexical properties of the words and (iv) the lexical resources like the *WordNet* and the *Oxford Advanced Learners' Dictionary (OALD)*. The components within the sets and the sets themselves are linked using the *semantic relations* of an interlingua for machine translation called the *Universal Networking Language (UNL)*. The work forms part of a UNL based MT system, where the source language is analysed *into* semantic graphs and target language is generated *from* these graphs. The system has been tested on the *Penn Treebank*, and the results indicate the effectiveness of our approach.

Keywords: Semantically Relatable Sets, Syntactic and Semantic Constituents, Interlingua Based MT, Parse Trees, Lexical Properties, Argument Structure, Penn Treebank.

1 Introduction

Analysis of sentences with a view to semantics extraction involves detecting *semantic* constituents of the sentence. These constituents are words that are semantically related and not necessarily *adjacent*. Systems for detecting chunks and n-grams do a meaningful but limited job of constituent determination (Lafferty *et.al*., 2001; Sha and Pereira, 2003). *Chunks* are supposed to consist of words that are adjacent to each other. They are thus *shallow* components of the sentence.

We look upon sentence analysis as a two stage process of determining:

a. Which words can form semantic constituents which we call *Semantically Relatable Sets (SRS)* and what after all are the SRSs of the given sentence; this needs solving various kinds of attachment problems.
b. What semantic relations can link the words in an SRS and the SRSs themselves.

Section 2, which follows this section, elucidates the concept of SRS through various examples. It should be noted that these SRSs are not necessarily chunks or words dominated by a non-terminal in the parse tree. *They are a group of entities which demand semantic relations or speech act attributes when the semantic representation of the sentence is ultimately produced.*

The linguistic insight for the paper is obtained from the following related works: Chomsky (1981), Grimshaw (1990), Jackendoff (1990), Kim (2002), Levin (1993), Mohanty *et. al*.(2004, 2005). In NLP, PP-attachment is a classical problem, that has been studied by several researchers, such as, Brill and Resnik (1994), Boland and Blodgett (2003), Kordoni (2003), Hindle and Rooth (1993), Dorr (1994), Niemann (2003), Ratnaparkhi et. al. (1994), Alda and Patrick (2003); among others.

Our work is ultimately an exercise in knowledge representation; the knowledge representation problem has been extensively discussed in the classical treatises by Dorr (1992), Schank (1972), Sowa (2000) and Woods (1985). Inerlingua representations have been studied in the machine translation literature (Hutchins and Somers 1992). One of the early noteworthy interlingua based MT systems is Atlas-II (Uchida, 1989); the comparison of the interlingua approach to the more widespread transfer approach is done in Boitet (1988); the consequence of language divergence on interlingua has been recently studied in Dave et. al. (2002).

The road map of the paper is as follows. Having elucidated SRS in Section 2, we discuss in Section 3 how *attachment* and related problems need to be solved before the SRSs are found. Section 4 shows why the parse tree is the correct starting point for finding the SRSs. Section 5 discusses the implementation of the system. Section 6 is on evaluation which describes the UNL generation as the task at hand using the SRS theory. Section 7 concludes the paper.

2 Semantically Relatable Sets (SRS)

Consider the sentence:

```
(1)  The man bought a new car in
     June.
```

This sentence contains five content words - *man, bought, new, car, June* - and three function words - *the, a, in*. In order to obtain the semantic representation of (1), we need the following sets:

```
(2)  a. {man, bought}
     b. {bought, car}
     c. {bought, in, June}
     d. {new, car}
     e. {the, man}
     f. {a, car}
```

The words within these sets have to be related and the sets themselves need linking.

We postulate that a sentence needs to be broken into sets of at most three forms, as shown in (3).

```
(3)  a. {CW, CW}
     b. {CW, FW, CW}
     c. {FW, CW}
```

The notation FW stands for function words; CW stands either for a *content word* or for a *clause*. These sets are called *Semantically Relatable Sets (SRS)* and are defined below.

Definition: A **semantically relatable set (SRS)** of a sentence is a group of *unordered* words in the sentence (not necessarily consecutive) that appear in the semantic graph of the sentence as linked nodes or nodes with speech act labels.

Thus, SRSs are the building blocks of the semantic graph of the sentence, as illustrated in Figure 1. They roughly correspond to chunks, but differ from them crucially in the fact that the components need not be consecutive. The latter fact needs solving the attachment problem.

SRSs can be used to represent different kinds of constituents as illustrated below. Consider the sentence (4).

```
(4)  The boy saw the girl in the
     office.
```

The sets, *{The, boy}, {boy, saw}* and *{the, office}* are three SRSs which are generated from semantically connected words in the sentence. The sets *{saw, girl}* and *{saw, in, office}* illustrate the fact that SRSs can span across the sentence to bring together semantically related non-consecutive entities like *saw* and *office*.

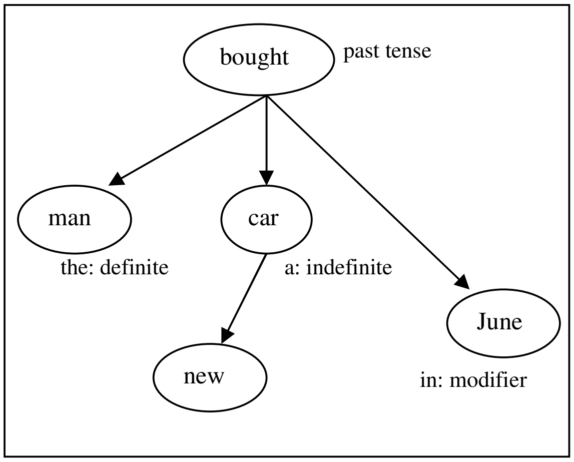

Figure 1: Semantic graph of the sentence (1)

```
(5) The boy said that he was
    reading a novel.
```

In sentence (5), the embedded clause *he was reading a novel* is denoted in the SRS representation by the term *SCOPE*. The SRS for the words within the clause such as *{he, reading}* are marked as being under *SCOPE*, as illustrated in (6). The semantic relation between the embedded clause and the words in the main clause is depicted through the SRS *{said, that, SCOPE}*.

```
(6)  a. {the boy}
     b. {boy, said}
     c. {said, that, SCOPE}
     d. SCOPE:{he, reading}
     e. SCOPE:{reading, novel}
     f. SCOPE:{a, novel}
     g. SCOPE:{was, reading}
```

The phrase *John and Mary* in sentence (7) represents a compound concept and is hence, marked under *SCOPE.*

```
(7) John and Mary went to school.
```

The linking of this phrase to the rest of the sentence is indicated by (8a).

```
(8)  a.{SCOPE, went}
     b.SCOPE:{John, and, Mary}
     c.{went, to, school}
```

These examples illustrate different cases of SRS construction leading to the semantics of a sentence.

3 SRS and Attachment Problems

Since the components of SRSs straddle word boundaries, the constructions of SRSs often need solving different kinds of attachment problems.

3.1 PP Attachment

We focus our attention on the particular frame [V-NP_1–P-NP_2], for which the prepositional phrase attachment sites under various conditions are enumerated, as shown in Table 1. The descriptions are self explanatory.

Conditions	Sub-conditions	Attachment Point
[PP] is subcategorized by the verb [V]	[NP_2] is licensed by a preposition [P]	[NP_2] is attached to the verb [V] *(e.g., He forwarded the mail to the minister)*
[PP] is subcategorized by the noun in [NP_1]	[NP_2] is licensed by a preposition [P]	[NP_2] is attached to the noun in [NP_1] *(e.g., John published six articles on machine translation)*
[PP] is neither subcategorized by the verb [V] nor by the noun in [NP_1]	[NP_2] refers to [PLACE] *feature*	[NP_2] is attached to the verb [V] *(e.g., I saw Mary in her office; The girls met him on different days)*
	[NP_2] refers to [TIME] *feature*	

Table 1: PP-attachment conditions for the frame [V-NP_1-P-NP_2]

Assuming that the PP-attachment is resolved using these heuristics, the sentence in (9) can be broken into SRSs as shown in (10).

```
(9)John published an article in
   June.
```

```
(10)(John, published)-----(CW,CW)
    (published, article)-(CW,CW)
    (published,in,June)-(CW,FW,CW)
    (an, article)--------(F,CW)
```

3.2 Infinitival Clause

Theoretically, to-infinitival clauses have an empty pronominal, called PRO, which is covertly present as the grammatical subject of the clause. Detection of the PRO elements in a to-infinitival clause, and subsequent resolution of the co-indexing of the PRO element are needed for SRS generation.

(11) I forced him_i $[\text{PRO}]_i$ to watch this movie.

In (11), the PRO element is co-indexed with *him.* The SRSs generated for the sentence in (11) are given in (12).

```
(12){I, forced} -----------{CW,CW}
    {forced, him}---------{CW,CW}
    {forced, SCOPE}-------{CW,CW}
   SCOPE:{him,to,watch}-{CW,FW,CW}
   SCOPE:{watch,movie}---{CW,CW}
   SCOPE:{this,movie}----{FW,CW}
```

In (12), the entire to-infinitival clause appears under a SCOPE and this is referred to in the SRS *(forced, SCOPE)*. The entity *him* acts as the object of the matrix verb *forced* as well as the entity participating in the SRS *SCOPE:{him, to, watch}* by virtue of its co-indexing with the PRO element.

3.3 Complex Sentences

Embedded clausal constructs need to be resolved for SRS generation.

```
(13) Mary claimed that she had
     composed a poem.
```

In (13), the matrix verb *claim* subcategorizes a *that-clause* and takes the clause *she had composed a poem* as its complement. The word that connects these two concepts is the complementizer *that* and the SRS for this part of the sentence is *{claim, that, SCOPE}*, where SCOPE covers the entire embedded clause. The SRSs are as in (14).

```
(14){Mary, claimed}--------{CW,CW}
    {claimed,that,SCOPE}--------
                       --{CW,FW,CW}
    SCOPE:{she,composed}--{CW,CW}
    SCOPE:{composed,poem}--{CW,CW}
    SCOPE:{a, poem}--------{FW,CW}
    SCOPE:{had, composed}--{FW,CW}
```

In (15), the relative clause *that John solved* modifies the preceding noun *problem.*

```
(15) The problem [that John solved]
     was easy.
```

The lexical item *that* plays the role of a relative pronoun and not that of a complementizer. The modifier relation between the clause and the noun *problem* is represented by the SRS *{problem, SCOPE}.* The SRSs generated for the sentence are given in (16).

```
(16){SCOPE,was,easy}-----{CW,FW,CW}
    SCOPE:{John, solved}--{CW,CW}
    SCOPE:{that, solved}--{CW,CW}
    {problem, SCOPE}------{CW,CW}
```

```
(17) John ignored the fact that
     Mary was unhappy.
```

In (17), the abstract noun *fact* subcategorizes an appositive clause, *i.e., Mary was unhappy,* as its complement. Since this clause is introduced by the complementizer *that*, the SRS for the clause attachment relation is *{fact, that, SCOPE}.* The complete set of SRS is given in (18).

```
(18) {John, ignored}------{CW,CW}
     {ignored, fact}------{CW,CW}
     {the, fact}}---------{FW,CW}
     {fact,that,SCOPE}}--{CW,FW,CW}
     SCOPE:{Mary,was,unhappy}------
                      --{CW,FW,CW}
```

4 Mapping from Syntax to Semantics

We use a probabilistic parser (Charniak, 2004) and lexical resources like WordNet 2.0 (Miller, 2003) and OALD (Hornby, 2001) to generate the SRSs.

In a parse tree, the tags *NP, VP, ADJP* and *ADVP* indicate the presence of content words while the tags *PP* (prepositional phrase), *IN* (preposition) and *DT* (determiner) denote function words. Consider the sentence (19).

```
(19)  John has bought a car.
```

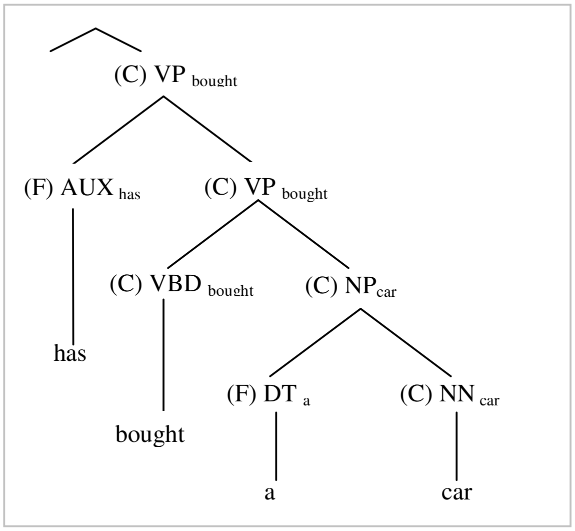

Figure 2: Parse Tree for *has bought a car*

The partial parse tree for this sentence is shown in Figure 2 with the (C) and (F) tags denoting content and function words and the subscripts indicating the head words. It is observed that most SRSs are constituted of the headwords of sibling nodes. For (19), *bought* and *car* being siblings form an SRS of the form {CW,CW}, *i.e.*, *{bought, car}*. Since *has* and *a* are FWs, they attach to their sibling CWs *bought* and *car* to form *{has, bought}* and *{a, car}*.

4.1 Attachment Resolution for PPs

In the parse tree, the PPs are often shown wrongly attached. Using the *noun class* of the preceding nouns, the time and place features of the noun within the PP and the subcategorization information provided for the preceding nouns and verbs, we achieve *resilience to attachment failures* of the parser. Consider the sentence (20) and its partial parse tree in Figure 3.

```
(20)  John has published an article
      on linguistics.
```

The nodes under the PP *on linguistics* have their headwords as *on*, a FW and *linguistics*, a CW. This combination can be attached to a preceding CW like *article* or *published* to obtain a {CW, FW, CW} set. Using the heuristics presented in Table 1 and subcategorization information for *published* and *article*, we obtain the SRS *{article, on, linguistics}*.

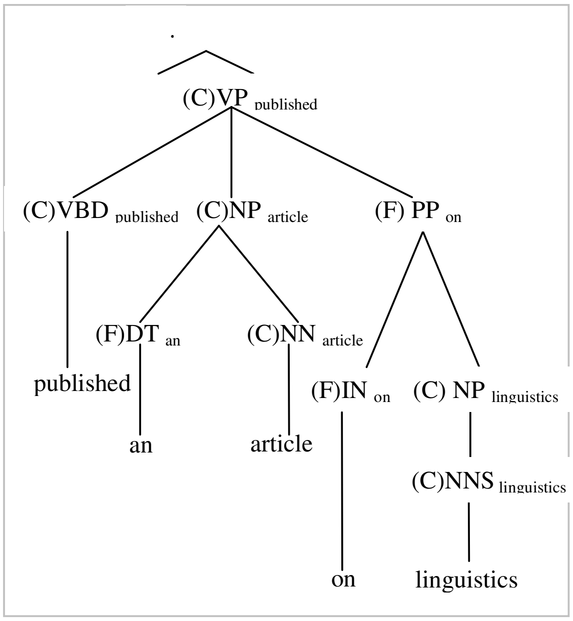

Figure 3: Parse tree for *published an article on linguistics*

4.2 To-infinitival clause

The partial parse tree for the to-infinitival clause in sentence (21) is shown in Figure 4.

```
(21) I  forced  him  to  watch  this
   movie.
```

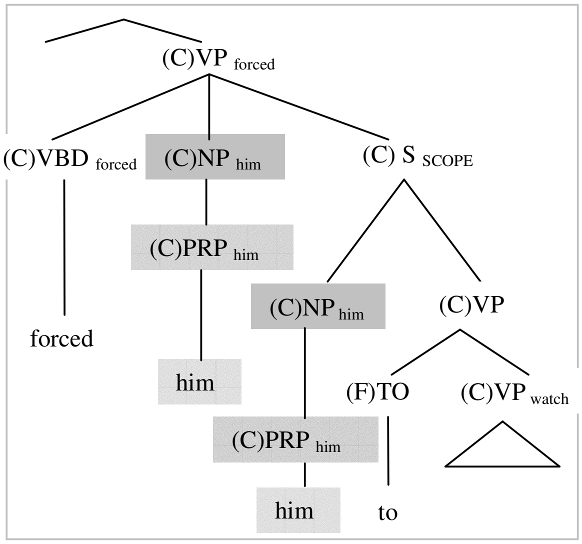

Figure 4: Partial parse tree for the to-infinitival clause in (21)

The TO node under the VP node indicates that the VP heads a to-infinitival clause. The fact that the PRO element *him* is semantically the object of *forced* is not depicted in the parse tree. Hence, the following modifications are done to the parse tree as shown in Figure 4:

a. The clause boundary is the VP node, which is labeled with the head *SCOPE* to indicate that it is a compound concept. Its tag is also modified to *TO*, a FW tag, indicating that it heads a to-infinitival clause,
b. The duplication and insertion of the NP node (depicted by shaded nodes in Figure 4) with head *him* as a sibling of the VBD node with head *forced* is done to bring out the existence of a semantic relation between *force* and *him*.

4.3 Linking of Clauses

In the parse tree of a complex sentence, the embedded clause boundary is correctly marked through an SBAR node. Consider sentence (22) and its partial parse tree in Figure 5.

```
(22) John said that he was reading
     a novel.
```

In the parse tree, the SBAR node has the complementizer, *that* and the S node as its children. The head of the S node is marked as *SCOPE*, since it takes an entire sentence as its subtree. The CW *said* to which the entire SBAR structure is attached, is taken as the first CW in the {CW, FW, CW} set to be generated. This leads to the generation of the SRS *{said, that, SCOPE}*. Adverbial clauses also have similar parse tree structures except that the subordinating conjunctions are different from *that*.

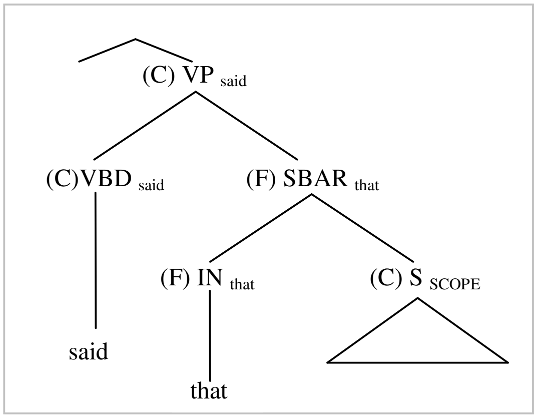

Figure 5: Partial Parse tree for the complement clause in (22)

Appositive and relative clauses too are marked by SBAR nodes and the attachment of the SBAR nodes is to an NP. In our approach, except for clausal constructs which attach to nouns, we take the parser's attachment of the SBAR node as the correct one. Extra analysis is done only for the case of noun attachments wherein the subcategorization information for nouns like *fact* and *boy* is used to distinguish between the appositive clause and relative clause cases.

The relative pronoun *that* and complementizer *that* are not differentiated in the parse tree and both appear with the tag IN. The subcategorization details of nouns are used to distinguish between these two cases.

5 Implementation

A high-level overview of the SRS Generator system is presented in Figure 6. The two important blocks in the system are (a) the module which determine the heads of the nodes and identifies clause boundaries for the creation of scope, using a bottom-up strategy and (b) the SRS generator module which uses an attachment resolver for generating the correct sets.

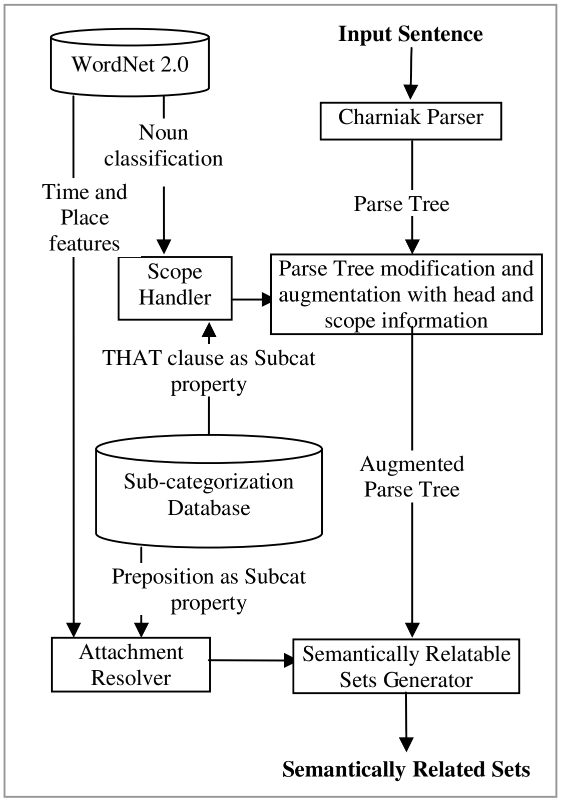
Input Sentence
WordNet 2.0
Charniak Parser
Noun classification
Time and Place features
Parse Tree
Scope Handler
Parse Tree modification and augmentation with head and scope information
THAT clause as Subcat property
Sub-categorization Database
Augmented Parse Tree
Preposition as Subcat property
Attachment Resolver
Semantically Relatable Sets Generator
Semantically Related Sets

Figure 6: Overview of the SRS Generator

5.1 Strategy

The *head determination* module uses a bottom-up strategy to determine the headword for every node in the parse tree. This head information is crucial in obtaining the SRSs, since wrong head information may end up getting propagated all the way up the tree. This module processes the children of every node starting from the rightmost child and checks the head information already specified against the node's tag to determine the head of the node. Some special cases are highlighted here:

a. In case of an SBAR node, the scope handler module is invoked to decide the kind of clause, scope creation points and heads for the nodes in the locality of the SBAR node.
b. A VP node is checked for the following cases:
   i. In the to-infinitival clause case, PRO insertion is done according to whether the PRO element is displayed within the S node by the parse tree or is missing completely.
   ii. If the copula, *be*, is the head of a VP and is followed by an adjectival predicate, the head of the adjectival phrase is taken to be the head of the predicate. *E.g., She is famous*
   iii. Phrasal verbs, when detected, cause modification of the tree which results in deletion of the particle following the verb. *E.g., look up*
c. NP nodes are checked for of-PP cases and conjunctions under them, which lead to scope creation.

The *scope handler* module performs modification on the parse trees by insertion of nodes in to-infinitival cases and adjustment of the tag and head information in case of SBAR nodes.

The *Semantically Relatable Sets Generator* module performs a breadth-first search on the parse tree and performs the following processing at every node N_1 of the tree. Depending on whether N_1 is a CW or a FW, several checks are performed as stated below. In the algorithm presented below, the example words applicable at every point are given in braces along with their tags. In this discussion, the S nodes which dominate entire clauses (main or embedded) are treated as CWs and SBAR and TO nodes are treated as FWs. The actual algorithm is now presented.

Algorithm

```
If the node $N_1$ is a CW (new/JJ, published/VBD, fact/NN, boy/NN, John/NNP) perform the following checks:
a) If the sibling $N_2$ of $N_1$ is a CW (car/NN, article/NN, SCOPE/S)
       Then create {CW,CW} ({new, car}, {published, article}, {boy, SCOPE})
b) If the sibling $N_2$ is a FW (in/PP, that/SBAR, and/CC)
       Then, check if $N_2$ has a child FW, $N_3$ (in/IN, that/IN) and a child CW, $N_4$ (June/NN, SCOPE/S)
        i. If yes,
              Then use attachment resolver to decide the CW to which $N_3$ and $N_4$ attach.
              Create{CW,FW,CW} ({published, in, June}, {fact, that, SCOPE})
       ii. If no,
              Then check if next sibling $N_5$ of $N_1$ is a CW (Mary/NN)
              If yes,
                     Create {CW,FW,CW} ({John, and, Mary})

If the node $N_1$ is a FW (the/DT, is/AUX, to/TO), perform the following checks:
a) If the parent node is a CW (boy/NP, famous/VP)
          Check if sibling is an adjective.
          i. If yes, (famous/JJ)
                 Then, create {CW,FW,CW} ({She, is, famous})
         ii. If no,     (boy/NN)
                 Then, create {FW,CW} ({the, boy}, {has, bought})
b) If the parent node $N_6$ is a FW (to/TO) and the sibling node $N_7$ is a CW (learn/VB)
              Use attachment resolver to decide on the preceding CW to which $N_6$ and $N_7$ can attach.
              Create {CW,FW,CW} ({exciting, to, learn})
```

The *attachment resolver* module takes a CW1, a FW and a CW2 as input and checks the time and place features of CW2, the noun class of CW1 and the subcategorization information for the CW1 and FW pair, to decide the attachment. If none of these yield any deterministic results, we fall back on the attachment indicated by the parser.

6 Evaluation

We used the Penn Treebank (LDC, 1995) as the testbed. The un-annotated sentences - which are actually from the WSJ corpus (Charniak et.al. 1987) - were passed through the SRS generator (*cf.* Section 5). The results were compared with the Treebank's annotated sentences. We hasten to add that we take only those cases where the Treebank shows correct semantic grouping. Simultaneously was tested the correctness of UNL generation.

6.1 Experiments and Top Level Statistics

The statistics presented in Table 2 shows that the sentences used for testing had a considerable number of PPs, to-infinitival clauses in particular and embedded clauses in general (as indicated by the number of S nodes).

General Statistics	
Total no. of Sentences Tested	#1745
Total no. of S nodes	#6789
Total no. of to-infinitival clauses	#403
Total no. of PPs	#4456

Table 2: General Statistics

The SRSs generated by our system were compared with the SRS-like sets derived from the Penn Treebank's parse trees. The comparison (*See Appendix I*) of these outputs gave the recall and precision figures reported in Figure 7, where recall and precision are defined as given in (23) and (24).

$$(23)\quad \text{Recall} = \frac{\#\,\text{matched SRS}}{\#\,\text{SRS expected by the Treebank}}$$

$$(24)\quad \text{Precision} = \frac{\#\,\text{matched SRS}}{\#\,\text{SRS output by our system}}$$

Figure 8 gives the recall and precision figures for some of the language constructs handled in our system.

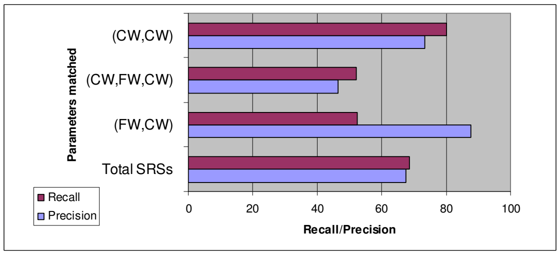

Figure 7: Results for SRS generation parameters

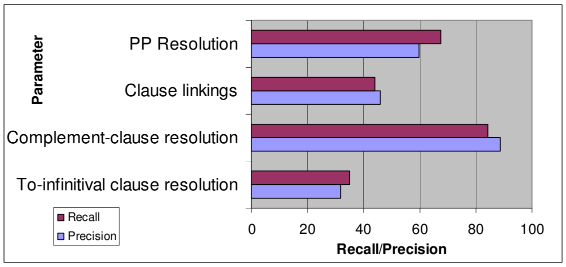

Figure 8: Results for different sentence constructs.

The system is found to perform well (Recall: 67.52%, Precision: 68.49%) on an overall basis as the SRS generation results show. The complement clause resolution results are also good. The low values for some of the other parameters were analysed and it was found that the reasons are:

a. There are constructs and Penn Treebank tags which are not handled by our system, and
b. The differences in conventions in the parse tree formats of the Charniak parser and Penn Treebank.

Manual checking (*see Appendix I*) of the output also revealed that the generated SRSs tallied with the semantics of the sentences. The robustness of our approach stems from the fact that even if the system is unable to handle a particular construct, it gives partially correct SRSs.

7 Conclusion and Future work

In this paper we have reported a system that attempts to reach at the semantic representation by first solving an essential problem. This problem is the determination of *Semantically Relatable Sets (SRS)* which are basically *semantic constituents* composed of content words (not necessarily contiguous), function words and clauses. The classical attachment, co-indexing and empty PRO determination problems need to be solved on the way. The results establish the efficacy and promise of the approach and hint at improvements achievable through (i) more thorough exploitation of lexical properties and argument frames and (ii) comparing directly against corpora of semantic graphs.

As mentioned in the abstract, the work reported is part of an MT effort involving interlingua. The specific interlingua used is called *Universal* Networking Language (UNL) (Uchida, 1999; UNDL, 2003). The SRS theory as outlined above has been tested by actually producing the UNL-graphs of sentences, and the results are found to be good.

References

Alda, M. and S. A. Patrick. 2003. A Conceptual structure for prepositions denoting instrumentality. *ACL-SIGSEM Proceedings of the first workshop on the syntax and semantics of prepositions and computational linguistics applications,* Toulouse.

Brill, E., and R. Resnik. 1994. A Rule based approach to Prepositional Phrase Attachment disambiguation. *Proc. of the fifteenth International conference on computational linguistics*. Kyoto.

Charniak, Eugene, Don Blaheta, Niyu Ge, Keith Hall, John Hale and Mark Johnson. *WSJ Corpus Release 1*. LDC.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris, Dordrecht.

Dorr, Bonnie. 1992.. The use of lexical semantics in Interlingua Machine Translation, *Machine Translation*, 7.

Sha, Fei and Fernando Pereira. 2003. Shallow Parsing with Conditional Random Field. Proc. of Human Language Technology- NAACL 2003, Edmonton, Canada.

Grimshaw, Jane. 1990. *Argument Structure*. The MIT Press, Cambridge, Mass.

Hindle, D. and M. Rooth. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics,* 19(1).

Hornby, A. S.: *Oxford Advanced Learner's Dictionary of Current English*. Oxford University Press, Oxford.(2001)

Jackendoff, Ray. 1990. *Semantic Structures*. The MIT Press, Cambridge.

Kordoni, Valia. 2003. A Robust Deep Analysis of Indirect Prepositional Arguments. *Proc. of ACL-SIGSEM workshop on preposition* 2003, Toulouse.

Lafferty, J., A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. of ICML-01, 2001.

LDC, 1995. Penn Treebank Release II. Linguistic Data Consortium.

Levin, Beth. 1993. *English verb Classes and Alternation*. The University of Chicago Press, Chicago.

Miller, George. 2003. Wordnet 2.0. http://wordnet.princeton.edu/

Mohanty, Rajat K., Ashish F. Almeida and Pushpak Bhattacharyya. 2004. The Complexity of *OF* in English. *Proceedings of International Conference on Natural Language Processing* (ICON-2004), Hyderabad, India.

Mohanty, Rajat K., Ashish F. Almeida and Pushpak Bhattacharyya. 2005. Prepositional Phrase Attachment and Interlingua. In *Research on Computing Science*, J. Cardenosa, A. Gelbukh & E. Tovar, eds., Instituto Politecnico Nacional, Mexico.

Niemann, Michael. 2003. Determining PP attachment through Semantic Associations and Preferences. MS.

Ratnaparkhi, Adwait. 1998. Statistical Models for Unsupervised Prepositional Phrase Attachment. Proc. of COLING-ACL 1998. http://www.cis.upenn.edu/~adwait /statnlp.html

Uchida, Hiroshi, M. Zhu, and T. Della. Senta. 1999. UNL: A Gift for a Millennium.The United Nations University, Tokyo. http://www.undl.org/publications/gm/top.htm

UNDL Foundation. 2003. The Universal Networking Language (UNL) specifications version 3.2. (2003) http://www.unlc.undl.org

UNU/ IAS. 2003. EnConverter Specification Version 3.3. UNL Center, United Nations University/ Institute of Advanced Studies, Tokyo.

Appendix I Testing Example

A. Sentence: `A form of asbestos once used to make Kent cigarette filters has` *`caused a high percentage of cancer deaths among a group of workers`* `exposed to it more than 30 years ago, researchers reported.`

B. Penn Tree: Partial tree for the chunk *`caused a high percentage of cancer deaths among a group of workers`*:

```
…
(VP (VBN caused)
  (NP
      (NP (DT a) (JJ high) (NN percentage) )
        (PP (IN of)
            (NP (NN cancer) (NNS deaths) ))
        (PP-LOC (IN among)
            (NP
               (NP (DT a) (NN group) )
                (PP (IN of)
                  (NP
                    (NP (NNS workers) )
                     . . .
```

C. SRSs derived from the treebank :

```
(1)   {caused, deaths}
(2)   {a, percentage}
(3)   {high, percentage}
(4)   {percentage, of, deaths}
(5)   {percentage, among, workers}
(6)   {cancer, deaths}
(7)   {a, group}
(8)   {group, of, workers}
```

D. Obtained output from our SRS generator system:

```
(1)   {caused, deaths}
(2)   {a, percentage}
(3)   {high, percentage}
(4)   {percentage, of, deaths}
(5)   {deaths, among, workers}
(6)   {cancer, deaths}
(7)   {a, group}
(8)   {group, of, workers}
```

Manual evaluation of the chunk of the sentence (A), *`caused a high percentage of cancer deaths among a group of workers`*, reveals that our system generates all eight correct SRSs (shown in C), whereas there are only seven correct SRSs (shown in B) which are derived from the Penn Tree. Accordingly, in the process of SRS matching (between Penn tree derived SRSs and the SRSs generated by our system), only seven SRSs match. Although our SRSs generator gives 100% precision and recall for the above chunk, the SRSs in (C5) and (D5) do not match in the automatic process of evaluation leading to low recall and precision.