Partial Parsing: A Report on Work in Progress

Ralph Weischedel, Damaris Ayuso, R. Bobrow, Sean Boisen, Robert Ingria, Jeff Palmucci

BBN Systems and Technologies
10 Moulton St.
Cambridge, MA 02138

Abstract

This paper reports a handful of experiments designed to test the feasibility of applying well-known partial parsing techniques to the problem of automatic data base update from an open-ended source of messages. and the feasiblity of automatically learning semantic knowledge from annotated examples. The challenges arise from the incompleteness of any lexicon, sentences that average over 20 words in length, and the lack of a complete semantics.

Introduction

Traditionally natural language processing (NLP) has focussed on obtaining complete syntactic analyses of all input and on semantic analysis based on handcrafted knowledge. However, grammars are incomplete; text often contains new words; and there are errors in text. Furthermore, as research activities tackle broader domains and if the research results are to scale up to realistic applications, handcrafting knowledge must give way to automatic knowledge base construction.

An alternative to traditional parsers is represented in FIDDITCH (Hindle, 1983), MITFP (deMarcken 1990), and CASS (Abney, 1990). Instead of requiring complete parses, a forest is frequently produced, each tree in the forest representing a non-overlapping fragment of the input. However, algorithms for finding the semantics of the whole from the disjoint fragments have not previously been developed nor evaluated.

We are comparing several differing algorithms from various sites to evaluate both the effectiveness of such a strategy in correctly predicting fragments and the effectiveness of syntactic/semantic algorithms for combining fragments. One question our research is trying to answer is how well the linguistic expression of entities and the relational structures among them can be recovered for data base update without determining global syntactic structure and without full information regarding the vocabulary items. A second question is how well an algorithm to learn lexical semantic knowledge from examples will perform.

Application Context

For message processing, insistence on complete syntactic analysis is usually unnecessary, since much of the input isn't directly relevant to updating a data base, routing a message, or prioritizing it.

An example article from the Third Message Understanding Conference (MUC-3), illustrates how complete analysis is unnecessary; the first sixteen paragraphs relate the results of a summit between the presidents of Peru and Bolivia. Those paragraphs would not add anything to the MUC-3 data base on terrorist acts. However, the final two sentences of the article mention, almost incidentally, that a bomb exploded near the summit, and therefore, do provide data to be added to the terrorism data base.

Even at the sentential level, one is not likely to be able to reliably compute a full semantic interpretation. For instance, in the sentence below from the aforementioned article, only the material in italics actually contributes to the desired, pre-defined data base update.

A BOMB EXPLODED TODAY AT DAWN *IN THE PERUVIAN TOWN OF YUNGUYO,* NEAR THE LAKE, VERY NEAR WHERE THE PRESIDENTIAL SUMMIT WAS TO TAKE PLACE.

In a task such as MUC-3 the goal is to identify pre-defined classes of entities, e.g., terrorist events, and dates, and the relationships among them, e.g., the perpetrator of a given terrorist act. Below, we have listed the first seven of nineteen pre-specified classes of data to be extracted from the messages of MUC-3.

0. MESSAGE ID: identifier
1. TEMPLATE ID: identifier
2. DATE OF INCIDENT: date
3. TYPE OF INCIDENT: set element e.g., KIDNAPPING, ATTEMPTED KIDNAPPING, KIDNAPPING THREAT,...
4. CATEGORY OF INCIDENT: set element, e.g, TERRORIST ACT, STATE-SPONSORED VIOLENCE
5. PERPETRATOR: ID OF INDIV(S): a string
6. PERPETRATOR: ID OF ORG(S): a string

System Architecture Assumptions

For the purposes of this discussion, we assume a fairly standard system architecture as shown in Figure 1 below. We further assume that a domain model exists for the pre-specified data to be extracted. That is, every class of entities of importance is specified in a frame representation indicating subclass-superclass relationships and all other important binary relationships among them.

Processing sentences to the point of finding semantic interpretations is the topic of this paper. Pilot experiments are reported here on alternative algorithms to find interpretable fragments even when no global syntactic or semantic analysis can be found. We are particularly exploring probabilistic models for this processing, and have described experiments with various probabilistic models elsewhere (Ayuso, et al, 1990; Meteer, et al., 1991).

The discourse component has two roles. One is resolving references. The second role is to use hypotheses regarding what domain-specific events are being described in each paragraph or article. Particular events correspond to templates. Once a template has been hypothesized, and once all text has been processed, if the template requires (or expects) certain information that has not yet been found, the discourse processor looks for values of the right semantic type and plausible within the discourse structure of the article. This process will be described elsewhere.

The output templates consist of three types of fields: set fill (a pre-specified finite set of alternatives), string (a literal, uninterpreted string from the input), and denumerably infinite entities (integers, dates, identifiers, etc.).

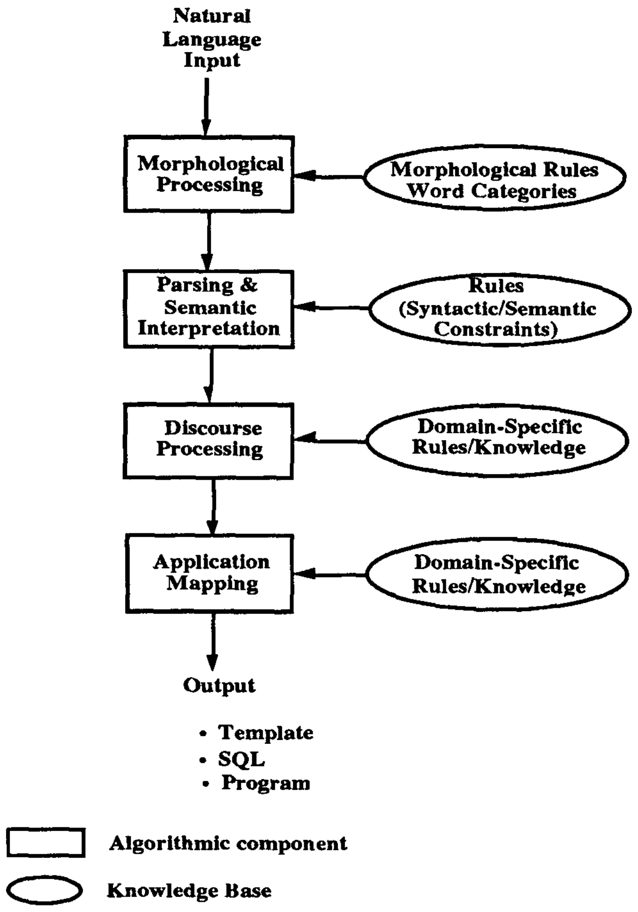
Natural Language Input
Morphological Processing
Morphological Rules Word Categories
Parsing & Semantic Interpretation
Rules (Syntactic/Semantic Constraints)
Discourse Processing
Domain-Specific Rules/Knowledge
Application Mapping
Domain-Specific Rules/Knowledge
Output
• Template
• SQL
• Program
Algorithmic component
Knowledge Base

Figure 1

Finding Core Noun Phrases

In a task such as MUC-3 one fundamental application goal is to identify pre-defined classes of entities, e.g., dates, locations, individuals, and organizations of primary interest in the domain. Normally these entities appear as noun phrases in the text. Therefore, a basic concern is to reliably identify noun phrases that denote entities of interest, even if neither full syntactic nor full semantic analysis is possible.

Two of our experiments have focussed on the identification of core noun phrases, a primary way of expressing entities in text. A core NP is defined syntactically as the maximal simple noun phrase, i.e., the largest one containing no post-modifiers. Here are some examples of core NPs (marked by italics) within their full noun phrases:

a joint venture with the Chinese government to build an automobile-parts assembly plant

a $50.9 million loss from discontinued operations in the third quarter because of the proposed sale

Such complex, full NPs require too many linguistic decisions to be directly processed without detailed syntactic and semantic knowledge about each word, an assumption which need not be true for open-ended text.

We tested two differing algorithms on text from the *Wall Street Journal* (WSJ). Using BBN's part of speech tagger (POST), tagged text was parsed using the full unification grammar of Delphi to find only core NPs, 695 in 100 sentences. Hand-scoring of the results indicated that 85% of the core NPs were identified correctly. Subsequent analysis suggested that half the errors could be removed with only a little additional work, suggesting that over 90% performance is achievable.

In a related test, we explored the bracketings produced by Church's PARTS program (Church, 1988). We extracted 200 sentences of WSJ text by taking every tenth sentence from a collection of manually corrected parse trees (data from the TREEBANK Project at the University of Pennsylvania). We evaluated the NP bracketings in these 200 sentences by hand, and tried to classify the errors. Of 1226 phrases in the 200 sentences, 131 were errors, for a 10.7% error rate. The errors were classified by hand as follows:

- Two consecutive but unrelated phrases grouped as one: 10
- Phrase consisted of a single word, which was not an NP: 70
- Missed phrases (those that should have been bracketed but were not): 12
- Ellided head (e.g. part of a conjoined premodifier to an NP): 4
- Missed premodifiers: 4
- Head of phrase was verb form that was missed: 4
- Other: 27

The 90% success rate in both tests suggests that identification of core NPs can be achieved using only local information and with minimal knowledge of the words. Next we consider the issue of what semantics should be assigned and how reliably that can be accomplished.

Semantics of Core Noun Phrases

In trying to extract pre-specified data from open-ended text such as a newswire, it is clear that full semantic interpretation of such texts is not on the horizon. However, our hypothesis is that it need not be for automatic data base update. The type of information to be extracted permits some partial understanding. For semantic processing, minimally, for each noun phrase (NP), one would like to identify the class in the domain model that is the smallest pre-defined class containing the NPs denotation. For each clause, one would like to identify the corresponding event class or state of affairs denoted.

Our pilot experiment focussed on the reliability of identifying the minimal class for each noun phrase.

Assigning a semantic class to a core noun phrase can be handled via some structural rules. Usually the semantic class of the head word is correct for the semantic class not only of the core noun phrase but also of the complete noun phrase it is part of. Additional rules cover exceptions, such as "set of ...". These heuristics correctly predicted the semantic class of the whole noun phrase 99% of the time in the sample of over 1000 noun phrases from WSJ that were correctly predicted by Church's PARTS program.

Furthermore, even some of the NP's whose left boundary was not predicted correctly by PARTS, nevertheless were assigned the correct semantic class. One consequence of this is that the correct semantic class of a complex noun phrase can be predicted even if some of the words in the noun phrase are unknown and even if its full structure is unknown. Thus, fully correct identification of core noun phrase boundaries and noun phrase boundaries may not be necessary to accurately produce data base updates.

Finding Relations/Combining Fragments

Though finding the entities of interest is fundamental to the task, finding relationships of interest among them is also critical. For instance, in MUC-3 one must identify terrorist events in any of nine Latin American countries, the perpetrators of the event, the victims, if any, the date, the location, any structural damage, and so on.

The experiments reported above were run by mid-summer, 1990. In fall, 1990, a more complete alternative, the MIT Fast Parser (MITFP), became available to us. It finds fragments using a stochastic part of speech algorithm and a nearly deterministic parser. It produces fragments averaging 3-4 words in length. An example output follows.

```
(S (NP (DETERMINER "A") (N "BOMB"))
  (VP (AUX (NP (MONTH "TODAY"))
        (PP (PREP "AT")
          (NP (N "DAWN"))))
    (VP (V "EXPLODED"))))
(PP
 (PP (PREP "IN")
   (NP (NP (DETERMINER "THE")
        (N "PERUVIAN")
        (N "TOWN"))
      (PP (PREP "OF")
        (NP (N "YUNGUYO")))))
 (PUNCT ","))
(PP (PP (PREP "NEAR")
     (NP (DETERMINER "THE")
       (N "LAKE")))
  (PUNCT ","))
(ADJP (DEGREESPEC "VERY")
    (ADJP (ADJ "NEAR")))
(ADV "WHERE")
(NP (DETERMINER "THE")
  (ADJP (ADJ "PRESIDENTIAL"))
  (N "SUMMIT"))
(VP (AUX) (VP (V "WAS")))
(VP (AUX "TO")
  (VP (V "TAKE")
    (NP (N "PLACE"))))
(PUNCT ".")
```

Figure 2 Example Output from MITFP

Certain sequences of fragments appear frequently, as illustrated in the tables below. One frequently occurring pair is an S followed by a PP (prepositional phrase). Since there is more than one way the parser could attach the PP, and syntactic grounds alone for attaching the PP would yield poor performance, semantic preferences applied by a post-process that combines fragments are called for.

Pair		Occurrences
S	PP	104
NP	VP	89
VP	VP	72
S	VP	65
PP	PP	62
PP	NP	58
NP	PP	56
VP	PP	54
PP	VP	48
NP	NP	34

Table 1 Most Frequently Occurring Pairs (In 2500 Pairs)

	Triple		Occurrences
NP	PUNCT	NP	53
VP	PUNCT	S	20
S	PUNCT	S	19
NP	PUNCT	S	19
S	PUNCT	NP	17
VP	PUNCT	N	12
NP	PUNCT	PP	10
NP	PUNCT	VP	9

Table 2 Frequently Occurring Fragment Pairs Surrounding Punctuation

In our approach, the first step is to compute a semantic interpretation for each fragment found without assuming that the meaning of each word is known. For instance, as described above, the semantic class for any noun phrase can be computed provided the head noun has semantics in the domain.

Based on the data above, a reasonable approach is an algorithm that moves left-to-right through the set of fragments produced by MITFP, deciding to attach fragments (or not) based on semantic criteria. To avoid requiring a complete, global analysis, a window two constituents wide is used to find patterns of possible relations among phrases. For example, an S followed by a PP invokes an action of finding all points along the "right edge" of the S tree where a PP could attach, applying the fragment combining patterns at each such spot, and ranking the alternatives.

As evident in Table 2, MITFP frequently does not attach punctuation. This is to be expected, since punctuation is used in many ways, and there is no deterministic basis grounds for attaching the constituent following the punctuation to the constituent preceding it. Therefore, if the pair being examined by the combining algorithms ends in punctuation, the algorithm looks at the constituent following it, trying to combine it with the constituent left of the punctuation.

A similar case is when the pair ends in a conjunction. Here the algorithm tries to combine the constituent to the right of the conjunction with that on the left of the conjunction.

Learning Semantic Information

Since the norm will be that there are several ways to combine a pair of fragments, we plan to test several alternative heuristics for ranking the alternatives. Probabilistic methods seem particularly powerful and appropriate. Thus far, we have tested this hypothesis on propositional phrase attachment.

Such semantic knowledge called *selection restrictions* or *case frames* governs what phrases make sense with a particular verb or noun (what arguments go with a particular verb or noun). Traditionally such semantic knowledge is handcrafted, though some software aids exist to enable greater productivity (Ayuso et al., 1987; Bates, 1989; Grishman et al., 1986; Weischedel, et al., 1989).

Instead of handrafting this semantic knowledge, our goal is to learn that knowledge from examples, using a three step process:

1. Simple manual semantic annotation,
2. Supervised training based on parsed sentences,
3. Estimation of probabilities.

Simple Manual Semantic Annotation

Given a sample of text, we annotate each noun, verb, and proper noun in the sample with the semantic class corresponding to it in the domain model. For instance, *dawn* would be annotated <time>, *explode* would be <explosion event>, and *Yunguyo* would be <city>. For our experiment, 560 nouns and 170 verbs were defined in this way. We estimate that this semantic annotation proceeded at about 90 words/hour.

Supervised Training

From the TREEBANK project at the University of Pennsylvania, we used 20,000 words of MUC-3 texts that had been bracketed according to major syntactic category. The bracketed constituents for the sentence used in Figure 2 appears in Figure 3 below.

```
( (S
        (NP a bomb)
        (VP exploded
            today
            (PP at
                 (NP dawn) )
            (PP in
                 (NP the Peruvian town
                      (PP of
                           (NP yunguyo) ) ) )
            ,
            (PP near
                 (NP the lake) )
            ,
            (SBAR (WHPP very
                          near
                           (WHADVP where) )
                    (S (NP the presidential summit)
                       (VP was
                            (S (NP *)
                               to
                               (VP take
                                    (NP place) ) ) ) ) ) ) )
        .)
```

Figure 3 Example of TREEBANK Analysis

From the example one can clearly infer that bombs can explode, or more properly, that *bomb* can be the logical subject of *explode*, that *at dawn* can modify *explode*, etc. Naturally good generalizations based on the instances are more valuable than the instances themselves.

Since we have a hierarchical domain model, and since the manual semantic annotation states the relationship between lexical items and concepts in the domain model, we can use the domain model hierarchy as a given set of categories for generalization. However, the critical issue is selecting the right level of generalization given the set of examples in the supervised training set.

We have chosen a known statistical procedure (Katz, 1987) that selects the minimum level of generalization such that there is sufficient data in the training set to support discrimination of cases of attaching phrases (arguments) to their head. This leads us to the next topic, estimation of probabilities from the supervised training set.

Estimation of Probabilities

The case relation, or selection restriction, to be learned is of the form X P O, where X is a head word or its semantic class; P is a case, e.g., logical subject, logical object, a preposition, etc.; and O is a head word or its semantic class.

One factor in the probability that O attaches to X with case P is $p'(X \mid P, O)$, an estimate of the likelihood of X given P and O. We chose to model a second factor $p(d)_1$, the probability of an attachment where d words separate the head word X from the phrase to be attached (intuitively, the notion of attachment distance).

Since a 20,000 word corpus is not much data, we used a generalization algorithm (Katz, 1987) to automatically move up the hierarchical domain model from X to its parent, and from O to its parent.

The Experiment

By examining the table of triples X P O that were learned, it was clear that meaningful information was induced from the examples. For instance, [<attack> *against* <building>] and [<attack> *against* <residence>] were learned, which correspond to two cases of importance in the MUC domain.

However, we ran a far more meaningful evaluation of what was learned by measuring how effective the learned information would be at predicting 166 prepositional phrase attachments that were not made by the MITFP. For example, in Figure 1, fragment 2 could be attached syntactically to fragment 1 at three places: modifying *dawn*, modifying *today*, or modifying *explode*.

Closest attachment, a purely syntactic constraint, worked quite effectively, having a 25% error rate. Using the semantic probabilities alone p' (X | P, O) had poorer performance, a 34% error rate. However, the richer probability model p' (X | P, O) * p(d) outperformed both the purely semantic model and the purely syntactic model (closest attachment), yielding an 18% error rate.

As a consequence, useful semantic information was learned by the training algorithm.

However, the degree of reduction of error rate should not be taken as the final word, for the following reasons:

- 20,000 words of training data is much less than one would want. An additional 70,000 words of training data should soon be available through TREEBANK.

- Since many of the head words in the 20,000 word corpus are not of import in the MUC-3 domain, their semantic type is vague, i.e., <unknown event>, <unknown entity>, etc.

Related Work

In addition to the work discussed earlier on tools to increase the portability of natural language systems, another recent paper (Hindle and Rooth, 1990) is directly related to our goal of inferring case frame information from examples.

Hindle and Rooth focussed only on prepositional phrase attachment using a probabilistic model, whereas our work applies to all case relations. Their work used an unsupervised training corpus of 13 million words to judge the strength of prepositional affinity to verbs, e.g., how likely it is for *to* to attach to the word *go*, for *from* to attach to the word *leave*, or for *to* to attach to the word *flight*. This lexical affinity is measured independent of the object of the preposition. By contrast, we are exploring induction of semantic relations from supervised training, where very little training may be available. Furthermore, we are looking at triples of head word (or semantic class), syntactic case, and head word (or semantic class).

In Hindle and Rooth's test, they evaluated their probability model in the limited case of verb - noun phrase - prepositional phrase. Therefore, no model at all would be at least 50% accurate. In our test, many of the test cases involved three or more possible attachment points fro the prepositional phrase, providing a more realistic test.

An interesting next step would be to combine these two probabilistic models (perhaps via linear weights) in order to get the benefit of domain-specific knowledge, as we have explored, and the benefits of domain-independent knowledge, as Hindle and Rooth have explored.

Conclusions

Two traditional approaches to applying natural language processing techniques are complete syntactic analysis and script-based analysis. In Proteus (Grishman, 1989), complete syntactic analysis is applied. If no complete analysis of a sentence can be found, the largest S found anchored at the left end is analyzed, ignoring whatever occurs to the right. A second alternative is script-based analysis e.g., as represented in FRUMP (de Jong, 1979). This technique emphasizes semantic and domain expectations, minimizing dependence on syntactic analysis.

In our approach to open-ended text processing, there are three steps:

1. Probabilistically based syntactic analysis produces a forest of non-overlapping fragments, if no single tree can be found.
2. A semantic interpreter assigns semantic representations to the trees of the forest.
3. Fragments are combined using a probability model reflexting both syntactic and semantic preferences.

However, the most innovative aspect of our approach is the automatic induction of semantic knowledge from annotated examples. The use of probabilistic models offers the induction procedure a decision criterion for making generalizations from the corpus of examples.

The partial parsing approach offers an alternative. By finding fragments based only on syntactic knowledge, and by starting a new fragment when a constituent cannot be deterministically attached, one has some partial analysis of the whole input. How to compute semantic analysis for any constituent is well understood in any compositional semantics. An algorithm that combines the semantically interpreted fragments seems to gain the power of semantically guided analysis without sacrificing syntactic analysis. Fragments that cannot be combined can still be employed with discourse processing and script-based expectations to identify the entities and relations among them for data base update.

Our pilot experiments indicate that the approach to text processing and the induction algorithm are both feasible and promising.

Acknowledgements

The work reported here was supported by the Advanced Research Projects Agency and was monitored by the Rome Air

Development Center under Contract No. F30602-87-D-0093. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

References

Abney, S., Rapid Incremental Parsing with Repair, Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research: Electronic Text Research,. UW Centre for the New Oxford English Dictionary and Text Research, 1990, 1-9 .

Ayuso, D. et al., Towards Understanding Text with a Very Large Vocabulary.. Proceedings of the Speech and Natural Language Workshop, Morgan Kaufman Publishers, Inc., 1990, 354-358.

Ayuso, D.M., Shaked, V., and Weischedel, R.M. An Environment for Acquiring Semantic Information. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 32-40. ACL, 1987.

Bates, M. Rapid Porting of the Parlance™ Natural Language Interface. *Proceedings of the Speech and Natural Language Workshop,* Morgan Kaufmann Publishers, Inc., pages 83-88, 1989.

Church, K. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. Proceedings of the Second Conference on Applied Natural Language Processing, ACL, 1988, 136-143.

Church, K., Gale, W.A., Hanks, P., and Hindle, D. Parsing, Word Associations and Typical Predicate-Argument Relations. *Proceedings of the Speech and Natural Language Workshop* Oct. 1989, 75-81.

De Jong, G.F. Skimming Stories in Real Time: An Experiment in Integrated Understanding. Yale University, Research Report No. 158, May 1979.

de Marcken, C.G. Parsing the LOB Corpus. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics* 1990, 243-251.

Grishman, R., and Sterling, J. Preference Semantics for Message Understanding. *Proceedings of the Speech and Natural Language Workshop* Oct. 1989, 71-74.

Grishman, R., Hirschman, L., Nhan, N.T., Discovery Procedures for Sublanguage Selectional Patterns: Initial Experiments, *Computational Linguistics* , 1986, 205-215.

Hindle, D., and Rooth, M. Structural Ambiguity and Lexical Relations. Proceedings of the Speech and Natural Language Workshop, Morgan Kaufman Publishers, Inc., 1990, 257-262.

Katz, S.M. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. ASSP-35 No. 3, March 1987.

Meteer, M., Schwartz, R., and Weischedel, R. Empirical Studies in Part of Speech Labelling., *Proceedings of the Speech and Natural Language Workshop,* 1991, this volume.

Stallard, D. Unification-Based Semantic Interpretation in the BBN Spoken Language System. *Proceedings of the Speech and Natural Language Workshop* Oct. 1989, 39-46.

Weischedel, R.M., Bobrow, R., Ayuso, D.M., and Ramshaw, L. Portability in the Janus Natural Language Interface. In *Speech and Natural Language,* pages 112-117. Morgan Kaufmann Publishers Inc., Oct. 1989.