Proposal for A Framework for the High-Precision Identification of Linguistic Relationships

Claire Cardie
Cornell University
Ithaca NY, 14853
cardie@cs.cornell.edu

Scott Mardis
Cornell University
Ithaca NY, 14853
mardis@cs.cornell.edu

May 28, 1997

Abstract

Current research in Information Retrieval and Information Extraction demands high-precision syntactic and semantic information from natural language text. We propose a plan for developing a framework to identify, with high-precision, the linguistic relationships between pairs of words in natural language text. Related research is reviewed and preliminary results are given. In our plan we outline the linguistic identification task: a new way to evaluate NLP systems.

1 Description of Project

Currently there is much interest in automatic processing of large volumes of natural language text; organizing, indexing, and summarizing it. Within this area there is a need for accurate searching of large corpora. Most work in this area is performed by researchers in the field of Information Retrieval (IR). Researchers in this field have identified a new area, High-precision IR, that focuses on accuracy by sacrificing the number of documents retrieved. Toward this end, they are seeking out more specific information present both in the queries and corpora on which to base searches. Some of this information is in the form of linguistic relationships. That is, rather than rely solely on the statistical distribution and ordering information of the lexical items, there is a need to utilize structural syntactic and semantic information in matching queries with documents.

Extracting this syntactic and semantic information requires natural language processing of both the queries and the document collections. High-precision is achieved by matching relationships found in the query with those found in documents. As an example, a query text might contain the phrase, "senate bills rejected by the President." This short text has many useful linguistic relationships that might be utilized by a High-Precision IR (HPIR) system. For example, there is an act of "rejecting" here with both a theme and an agent identified. "The President" is the agent and "senate bills" are the objects of the action (i.e., the theme). This represents at least two useful relationships that we might look for in searched text. We should prefer, for example, a document where we can positively identify an instance of "bills" being "rejected" over documents where we cannot identify such a relationship. Similarly we can prefer documents in which we identify "The President" "rejecting" something over those that do not have such a relationship. This provides a significantly larger degree of confidence in the quality of the retrieved document over matching schemes that rely solely on word distributions.

1.1 Project Goal

It is the goal of our research to develop a general framework for identifying linguistic relationships in natural language text with high-precision. Our goal is to focus on the deep semantic relationships of text, which may be reflected in perhaps many different syntactic forms. The IR community refers to these varied forms as "term invariants". As an example, consider that all of the following phrases have the "bill-reject" relationship mentioned above:

> The *bill* was *rejected* by the senate by a vote of 33-69.
>
> Most *bills* *rejected* by the White House are rewritten and find their way into other bills as amendments.
>
> At noon, White House press-secretary Mike McCurry told reporters that the President planned to *reject* without consideration any *bill* that did not include provisions for ....

Here, the same relationship is present though it appears in three different syntactic forms. Initially, we will concentrate on accurately identifying relationships in particular syntactic forms. Later, we will use these syntactic relationships to identify the various ways in which common underlying semantic relationships are represented. In the same manner as HPIR researchers, we will consider the trade-off between precision and recall, relying on the processing of large amounts of text to compensate for the possibly low rates of recall.

The application of high-precision linguistic relationship identification is not limited to HPIR systems. Kupiec [13] describes the Murax system, which searches an encyclopedia for answers to simple questions. This system utilizes an IR system for part of its retrieval process but then uses more specific and detailed syntactic information to try to verify the presence of linguistic relationships. An example given by Kupiec is, "Who was the Pulitzer Prize-winning novelist that ran for mayor of New York City?" This query has several relationships that can be independently verified, possibly in different articles in the encyclopedia. The correct answer, for example, must identify a person that has "run for mayor of New York City." The deep relationships in "mayor of New York City", "run for mayor", and "<person x> run-for <office>" all contribute to locating the correct answer. Our research is directed specifically at methods

Salvadoran president Alfredo Cristiani implicated four officers, including one colonel, and five members of the armed forces in the assassination of six jesuit priests and two women on 16 November at the Central American University.

1.	DATE	16 November 90
2.	LOCATION	El Salvador: Central American University
3.	TYPE	attack
4.	STAGE OF EXECUTION	accomplished
5.	INCIDENT CATEGORY	terrorist act
6.	PERP: INDIVIDUAL ID	"four officers", "one colonel", "five members of the armed forces"
7.	PERP: ORGANIZATION ID	"armed forces"
8.	PERP: CONFIDENCE	reported as fact
9.	HUM TGT: DESCRIPTION	"Jesuits", "women"
10.	HUM TGT: TYPE	civilian: "Jesuits" civilian: "women"
11.	HUM TGT: NUMBER	6: "Jesuits" 2: "women"
12.	EFFECT OF INCIDENT	death: "Jesuits" death: "women"

Figure 1: IE Example

and techniques that will locate such information with high-precision. The recall loss that we suffer in favor of precision might be compensated for by using multiple encyclopedias or other information sources.

1.2 Information Extraction

Another possible area of application for our research is in the field of Information Extraction (IE). Information Extraction involves the development of systems that are designed to pick detailed domain specific information from large corpora. The IE task, which is quite similar to our present research differs in its specificity. Typically, the information required is quite detailed, involving filling out many entries in a form that describes a complete event such as corporate takeover or terrorist attack [18, 19]. As an example, Figure 1 shows sample text with a properly completed frame describing the terrorist attack described in the text. Such systems are designed for a particular domain and a given set of information frames and therefore typically require a large knowledge engineering effort. As with IR, there is also a need for systems that perform high-precision IE. Our efforts, which seek to locate individual relationships with high- precision should be of immediate use to IE systems that are trying to collect such fine information into larger frame-like structures as shown in Figure 1.

1.3 Linguistic Relationships

So far we have yet to adequately define the term *linguistic relationship* as we mean it. By this term we refer to a wide variety of relationships between words that we might identify from lexical and syntactic information. We have given a few examples so far: "rejected bills", "mayor of New York City", "run for mayor". We think of these examples as instances of the more general relationships **action/object**, **noun/argument**, and **action/for-object**. When we refer to *linguistic relationship* we actually mean both the general and specific examples. In order to best outline the full extent of what we mean by *linguistic relationship*, we will try to define the most general types of relationships and then describe how the more specific types are derived from these.

A term that closely approximates what we mean by *linguistic relationship* is *thematic role*. *Thematic role*, as a concept, does not meet our needs but, for a number of reasons, it is a good basis for comparison. For example, in the sentence, "Marianne rode her bike to school," there is a verb "rode" which takes three arguments, "Marianne", "her bike", and "school". Each of these fills a particular thematic role for the verb "rode"; "Marianne" fills the **agent** role, "her bike" fills the **theme** role and "school" fills the **goal** role. It is the *argument structure* of the predicate "rode" that defines which roles the predicate has and the requirements on what fills each role. Since we are interested primarily, if not solely, in binary relationships, we can think of each of these roles in different terms: the **agent/predicate**, **theme/predicate** and **goal/predicate** relationships.

There are other relationships that we can define in terms of thematic roles that we also wish to include in our definition. For example, arguments of a common predicate, such as **actor/theme**, **theme/goal**, and **them/experiencer**. These relationships are not represented by any one role, but instead by two roles with a common predicate.

We also want to include relationships that are typically referred to by the term *adjuncts*. This refers to optional, or additional information attached to a predicate. An example is the **predicate/location** relationship, which is often represented in prepositional phrases (e.g., "ate dinner at home", "slept on the couch"). Indeed, any kind of prepositional phrase relationship we would include in our concept *linguistic relationship*.

Also included are **modifier/modifiee** relationships represented by examples like "disarmament treaty", "rejected bills", and "former CEO". At this point we start to diverge from the thematic role concept significantly.

Ultimately, the terms of thematic role relationships (agent, predicate, theme, etc.) will be too limiting as we truly wish to consider relationships both more general and specific. For example, we cannot define all our relationships in terms of predicates because some do not involve predicates (e.g., **modifier/modifiee**). Also we might want to be more specific and refer to only certain kinds of predicates: verbal predicates, nominal predicates, actions, titles (e.g., mayor), mental processes, etc. Indeed, we want to be able to specify various syntactic and semantic restrictions on both the related words and the relationship itself. We might define relationships such as a **noun/of-object** or **person/title**.

Thus far we have defined a wide class of binary relationships. As was just mentioned, we might wish to specify a relationship using syntactic (e.g., prepositional phrase using "in") or semantic (e.g., animate nouns, actions, etc.) restrictions. We might further restrict a particular relationship by specifying one of its words (or possibly just a morphological root). An **action/object** relationship constrained in this way might become *killed*/**theme** or **actor**/*bomb*-$_{VERB}$. Relationships of this sort are ideal for use with Information Extraction tasks where there are key triggering words known in advance.

For use with IR applications we like to include in our definition of *linguistic relationship* instances where both words have been tightly constrained (e.g., *research/in-Europe*, *cause*-$_{NOUN}$/*of-disease*, *kill-/in-NewYorkCity*, etc.). These represent the most specific forms of the relationships we wish to consider.

The exact boundary between what is and is not a *linguistic relationship* is ill-defined and probably fluid. The relationships that we will study are those that we believe will be of use in the applications we outlined at the beginning of the section. A succinct but slightly inadequate summary of use of the term might be as follows:

> A *linguistic relationship* is a relationship between open-class words in sentences or sentence fragments where the words and the relationships are constrained by general syntactic and semantic restrictions. The bulk of these are closely tied to noun and verb argument structure.

2 Approach

To reiterate, our goal is to develop a general framework for identifying linguistic relationships with high precision in natural language text. Currently, there is no such framework in existence. Information Extraction systems represent the most similar previous work but their focus has been on domain-specific relationships with little being done on high-precision extraction. We will start with a simple system architecture consisting mostly of standard algorithms from NLP (e.g., taggers, parsers, chunkers, etc.). We will evaluate these algorithms in terms of their contribution to identifying linguistics relationships, and then specialize the algorithms for our task. Feedback from experiments may be used to modify the architecture as well.

Our initial efforts will be based on the system architecture outlined in Figure 2. We have left off data structures such as grammars and lexicons to focus in on the processing elements of the system. The first step in processing text is assigning part-of-speech tags to the words of the input. These part-of-speech tags are used (in conjunction with the input words) as input to a parser that tries to determine some and possibly all of the syntactic structure. The output of the parser is used to extract whatever detailed information is needed from the text.

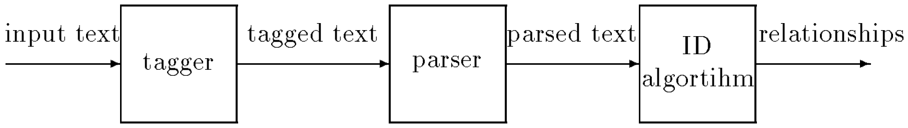

Figure 2: Initial Framework Architecture

We will begin with established methods of tagging and parsing and add algorithms to extract from the output of these two stages the linguistic relationship information that we are interested in. For our initial experiments we have chosen two sharply different parsing methods, probabilistic context free grammars (PCFG) and partial-parsing. These two methods represent very different approaches to parsing and should give us a good initial look at how the parsing technique will affect our linguistic relationship identification task.

By studying the correlation between the tagger & parser performance and the overall performance in identifying relationships, we hope to isolate areas for specializing the taggers, parsers, and relationship identification algorithms for higher-precision identification. At our disposal for such improvements (and for the identification algorithms themselves) are machine learning (ML) algorithms, statistical NLP algorithms, and the use of on-line syntactic and semantic information such as is available in on-line dictionaries or semantic networks such as WORDNET[17].

Machine Learning algorithms in particular seem suited to our task. We have a wide variety of relationships to consider and also a wide variety of syntactic and semantic evidence needed to identify them. The framework, therefore, will need to be flexible in terms of identifying which information to use in making decision for a particular relationship. Information will be required from both the local and global textual contexts. Machine Learning algorithms and techniques such as Case-base Learning, Decision Trees, and Neural nets (not a symbolic method) all share a common flexibility and allow the use of a wide source of information types.

3 Related Work

In this section we survey some of the more important NLP research related to this project. There are two basic approaches that we will cover: Full parsing plus extraction and partial-parsing plus attachment. Both of these could form the basis of a system to do high-precision relationship identification. With a full parse approach, the relationships would be extracted from the parse tree generated by the parser. Using a partial-parser, the relationships would be assembled from bracketed phrase chunks with special conditions on words and phrases that intervene between words involved in the relationship.

In this section we also cover two specific research efforts that bear some close relationship to our work. Both of these projects incorporate some elements of both traditional parsing and partial-parsing. From the IE literature, we have selected the pattern-based work of the Autoslog system [20], in conjunction with the CIRCUS project at the University of Massachusetts at Amherst, as being most relevant. From the parsing literature, we outline recent work by Collins [10] where a probabilistic parser is built based on relationships between lexical heads. This work distinguishes itself from other parsing research in that its particular statistical model is closely related to our linguistic relationship task.

3.1 Parsing

Parsing techniques vary widely based upon their theoretical underpinnings. Some researchers approach the parsing problem from the point of view that language is a set statistical regularities that may be learned [11, 5]. The n-gram model is the logical extreme of this view. Many others base their work on more traditional grammatical models of language.

Most of the grammatical approaches come from linguistic theory and are typically based upon models of human linguistic performance. A wide variety of such models (government & binding, lexical functional grammar, head-based phrase structure grammar) have been used as the basis for parsing systems but all typically provide inadequate coverage of the language for real-world applications. Some phenomena not adequately addressed by these grammars are prepositional phrase attachment, coordination, lexical ambiguity, appositives, and idiom processing.

It is this set of problems that motivates statistical approaches to parsing because the information that has been most helpful in performing prepositional-phrase attachment, lexical disambiguation, etc. can be extracted in statistical form from text corpora. The most successful parsers have often incorporated both the grammatical and statistical approaches by using a rule-based grammar with statistics of rule and lexeme usage.

One widely used parsing technique is based on the probabilistic context free grammar (PCFG). A PCFG is a context-free grammar (CFG) where a probability is added to each rule; for each rule, its probability is the conditional probability that the non-terminal from the left-hand-side of the rule is expanded into the right-hand-side. These probabilities can encode many of the statistical regularities of structure usage in language. Additionally, lexical probabilities may be incorporated to model lexical dependencies like verb sub-categorization or prepositional-phrase attachment preferences. PCFG statistics are easy to gather from bracketed corpora such as the Penn TreeBank.

Another successful parser incorporating many statistical regularities of language but without an explicit statistical model is that of Brill[7]. Brill's parser produced results comparable to many PCFG-based parsers through the use of transformation rules. The parser begins with a sentence bracketed in some regular way, such as all right-branching, and then applies an ordered set of transformations. Each transformation is fairly simple (e.g., delete a right bracket to the right of all determiners) and the whole set can be learned using a relatively small training corpus (relative to PCFG methods).

3.2 Partial Parsing

Partial parsing is a strategy aimed at resolving some of the difficulties found in applying traditional parsers to successively larger collections of unrestricted text. Traditional parsers are not well equipped to handle noisy or ungrammatical text (common in transcribed speech); they do not handle long or ambiguous texts well; and it is often difficult to recover from mistakes caused by incompleteness of the grammar. Partial parsing seeks to avoid these difficulties by sacrificing the goal of a complete and accurate parse. A good overview of partial-parsing techniques is given by Abney[3].

By only conducting partial analyses, partial parsers are better equipped to handle a wider variety of constructions, including ungrammatical ones, typically by ignoring anything for which it does not have a good model. Many partial parsers try to recover only the core of noun phrases. Others try to recover the non-recursive core of other major phrase types (e.g., verb phrases, adjectival phrases). The non-recursive parts of these phrases are termed chunks. The advantage of analyzing only these non-recursive parts is that they are typically identifiable using only local information and there are no attachment ambiguities because no attachment is done. If required, a second subsystem, often using largely semantic clues, can make attachment decisions. This division of labor, between forming chunks and computing attachments may play some part in human sentence processing [1, 2]. A number of approaches to partial parsing have been tried. One simple method involves dividing words into two classes: stop-words (typically function words) and non-stop-words [21]. Chunks are sequences of non-stop-words and the sequences of stop- words are termed chinks. Another method, by Church [9], involves finding the appropriate noun-phrase brackets by inserting open and close brackets into a part-of-speech sequence. Church used a stochastic model based on the local parts-of-speech.

A more traditional parser, that functions as a partial parser, is the Fidditch parser [12]. Fidditch tries to compute a complete parse but is able to *punt* when it is unable to correctly determine the proper attachment for a phrase. The Fidditch grammar is fairly complex and hand-coded.

Cascades of finite-state machines have also been used in partial parsing[4]. Such a cascade is an ordered sequence of finite-state-machines (FSM) each defined by a set of regular expressions and where each FSM operates on the output of the previous FSM. Early FSMs typically correspond to the familiar noun-phrase chunks. Later ones may try to recognize basic phrase structures. Such cascades can be compiled to be quite fast.

3.3 Conceptual Sentence Analysis

The work by Ellen Riloff [20] in conjunction with the CIRCUS system [15, 14] is perhaps the most relevant of recent work in conceptual sentence analysis. The

Name:	target-subject-passive-verb-bombed
Trigger:	bombed
Variable slots:	(target (*subject* 1))
Constraints:	(class (phys-target *subject*))
Constant Slots:	(type bombing)
Enabling Conditions:	((passive))

Figure 3: A Concept Node

linguistic patterns used by Riloff in the course of her work are a subset of the linguistic relationships that we are exploring in our research.

The CIRCUS system is the sentence analyzer for one of the many information extraction (IE) systems which participated in the Message Understanding Conferences (MUC) [18, 19]. CIRCUS is based upon a dictionary of *concept nodes* (semantic case frames) that are triggered by lexical items. Figure 3 shows a simple concept node that finds targets of bombing incidents in the terrorism domain. These dictionary entries, once activated by the triggering lexical item, make syntactic and semantic predictions which can be evaluated as more text is processed. Each dictionary entry, concept node, outlines the key information needed in the frame and where to retrieve it from the local linguistic context. Prior to filling out a case frame, the system must identify instances of the concept in the text. This task is similar to our linguistic identification task in that it is looking for particular words and relationships between them. Riloff's work [20], in particular, dealt with the automatic creation of the concept nodes and implicitly with the generalization of linguistic relations and their identification in text.

Riloff's primary interest was in automatically creating the dictionary entries from marked-up text. Her system was given a text that had been annotated with all the information relevant to a particular semantic case frame. Riloff's system (Autoslog) needed to identify the linguistic (primarily syntactic) context in which the information occurred and generate dictionary entries that CIRCUS could use to find similar instances of the concept in novel texts. Riloff identified 13 *conceptual anchor point heuristics* which are used to identify instances of the concepts in text. These heuristics (see Table 1) are linguistic patterns and represent syntactic relationships which CIRCUS could identify in novel texts. Autoslog generated candidate concept nodes by processing the marked up text. The candidate list was then edited by hand to remove bad extraction patterns. As was mentioned before, these linguistic patterns are syntactic relationships and are a subset of the relationships that are of interest to us. The CIRCUS/Autoslog system was designed to extract detailed semantic information. The Autoslog system essentially maps that detailed information onto the syntactic relationships in Table 1.

The concept nodes produced by Autoslog are specializations of the heuristics of Table 1 where one of the syntactic items (the triggering word) has been fixed.

Linguistic Pattern	Example
<subject> passive-verb	<victim> was murdered
<subject> active-verb	<perpetrator> bombed
<subject> verb infinitive	<perpetrator> attempted to kill
<subject> auxiliary noun	<victim> was victim
passive-verb <direct-object>	killed <victim>
active-verb <direct-object>	bombed <target>
infinitive <direct-object>	to kill <victim>
verb infinitive <direct-object>	threatened to attack <target>
gerund <direct-object>	killing <victim>
noun auxiliary <direct object>	fatality was <victim>
noun preposition <noun-phrase>	bomb against <target>
active-verb prep. <noun-phrase>	killed with <instrument>
passive-verb prep. <noun-phrase>	was aimed at <target>

Table 1: Autoslog Conceptual Anchor Point Heuristics

This type of relationship, where some lexical item is fixed, is of interest to us as it too is a type of relationship we wish to work with in our framework. For example, an Autoslog concept node might be used to find all sentences of the form "<person> was killed". This instantiation, where the word killed is fixed is much more specific than the general **subject/passive-verb** relationship. In our framework we hope to address specializations of this sort producing increased precision for very narrowly specified relationships.

Another facet of the CIRCUS system that we believe will be of importance in a high-precision system is that the system does not produce full parses. As a conceptual sentence analyzer, it does not have a separately identifiable parser per se. It is able to identify portions of noun-phrases and clauses and as such is more like partial parsing system. We believe that such partial parsing approaches are likely to yield better long term results for our task because of their speed and their robustness.

3.4 Collins' Statistical Parser

The statistical parser of Collins [10] is built on a model that emphasizes the identification of relationships that are of interest to us. The parser combines partial-parsing (noun phrase bracketing) and statistical parsing. The Collins statistical model is based on the probabilities between head-words in parse trees. In so doing, it explicitly represents the parse probabilities in term of basic syntactic relationships of the lexical heads. Collins defines a mapping from parse trees to sets of dependencies, on which he defines his statistical model. Some of the details of the model are as follows. First, a set of rules defines a

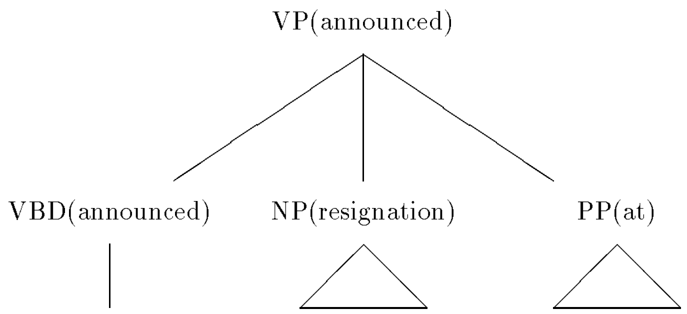

Figure 4: A Collins-type Parse Tree

head-child for each node in the tree. The lexical head of the head-child of each node becomes the lexical head of the parent node. Also associated with each node is a set of dependencies derived in the following way. For each non-head child, a dependency is added to the set where the dependency is identified by a triple consisting of the non-head-child non-terminal, the parent non-terminal, and the head-child non-terminal. To clarify, consider the following example. Figure 4 shows the internal node of some parse tree. Each node is labeled with its lexical head. From this tree, the two dependencies (there are two non-head children) produced are <NP,VP,VBD> and <PP,VP,VBD>.

In this manner, each constituent with n children produces $n-1$ dependencies. The statistical model is then based on the conditional probability of a the relationship given a pair of lexical heads. Formally, Collins defines the function $F(R| < word1, tag1 >, < word2, tag2 >)$ as the probability that the word $< word1, tag1 >$ modifies the word $< word2, tag2 >$ with relationship R, given that those two words appear in the same sentence. Collins computes a maximum likelihood estimate of this function from a collection of parsed text and uses the estimate to compute a parse probability for a proposed sentence. Overall, the computation is like that of PCFG parsing but uses a different statistical model.

Using the above model (with some modification to handle the sparsity of the maximum likelihood estimate, and a word proximity bias), Collins is able to achieve parsing results as good as the SPATTER parser [16], which was the best published at the time of Collins' writing. There are two important ways in which the Collins work bears on our line of work. First of all, the statistical model explicitly models relationships that are of interest to us such as syntactic verb-object roles. Second, by finding the most probably parse in this model, his parser should be more likely to get the syntactic head relationships correct. That is, the parser should maximize the probability of getting the relationships correct, rather than what most traditional parsers metrics achieve: maximizing the number of correctly labeled subtrees or minimizing the number of bracket crossings. In one sense, the aim of Collins work is a reversal of ours in that he is using relationship probabilities to do parsing whereas we are using parsing to accurately identify relationships.

The Collins approach to parsing may provide a good basis for finding relationships in text but as it is, it does not go far enough for our needs. First of all, it models purely syntactic information and as we have outlined before, our eventual goal is semantic relationships. Second, his model uses only syntactically local information such as the relationship of subjects to verbs and noun-objects to prepositions. It cannot model subject-object relationships or the relationships between verbs and the object of modifying prepositional- phrases. For example, it cannot directly represent the relationship between syntactic subject and object in the sentence "Thomas *Jefferson* was the *Vice-President* of the United States under John Adams." Even with these deficits, the Collins parser represents an important combination of lexical dependencies and parsing which we hope to exploit. Evaluating its performance on our task is an important next step for our research.

4 Preliminary Investigation

As was stated before, the basis of our work will be an understanding of how the current best-in-class components support the identification of linguistic relationships. We will do this by evaluating various NLP components in terms of their performance on our identification task.

We have begun this effort with two experiments; two experiments that directly correspond to the two approaches we discussed at the beginning of Section 3. They are 1) Parse & extract; and 2) Partial-parse & attach. For simplicity we have begun with a simple syntactic relationship, the verb-object relationship. This will allow us to quickly assess the general efficacy of varied techniques and it is also one for which we have a wealth of test data in the form of the bracketed Penn TreeBank corpus. Examples of the relationship we test are here:

> The screen *shows* two distorted, unrecognizable *photos*, presumably of two politicians.
>
> The campaign has *blamed* these reporting *problems* on computer errors.

Because we are interested in finding the relationships between lexical items, our task is defined in terms of the lexical heads; in this case, the head verb of the verb phrase and the head noun of the object noun phrase. In the examples given, <shows, photos> and <blamed, problems> are the word pairs that we wish to identify.

Since we want to evaluate the different approaches to parsing, our two experiments vary in their strategic approach. One experiment used a traditional best-first PCFG parser and the second used partial parsing. We describe the detailed design and preliminary results of these experiments in the following subsections. These two experiments represent widely differing approaches to

	# Sentences	Precision	Recall
PCFG	374	0.477	0.565
Partial Parsing	676	0.513	0.510

Table 2: Preliminary results

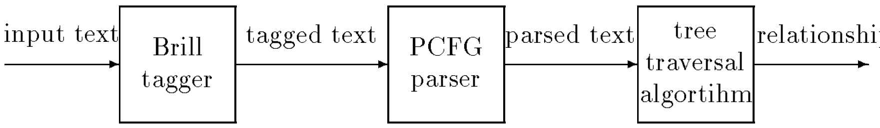

Figure 5: PCFG Experiment Architecture

parsing and will hopefully give us some insight as to what will be the most useful parsing model to pursue.

Table 2 shows preliminary results for the two experiments. Superficially, the two methods produce nearly equivalent results. Detailed analysis will show some similarities but also some key indications that the partial-parsing approach may be closer to the ultimate parsing model for our task. A summary and analysis of the results follow detailed descriptions of the experiments.

4.1 Experiment 1

The structure of the first experiment is depicted in Figure 5. Initially, all the words (includes punctuation lexemes) are assigned part-of-speech tags using the Brill transformational tagger [6]. The training text and tag-set are from the Penn TreeBank. The second step was to produce a complete parse on test texts using a best-first PCFG parser based on that presented in Allen[5]. We chose a PCFG parser because a) it enjoys wide success as a general parser and b) output parses are associated with probabilities, which we might be able to use for confidence factors in obtaining high-precision. In order to increase the speed of the parser to a point where we could process reasonably large sentences, we used a figure-of-merit based on the geometric mean of the sub-constituent probabilities rather than a simple product. Typically, a basic PCFG parser using a simple product of probabilities behaves very much like a breadth-first search algorithm because of the way in which the constituent probabilities shrink as the depth of the tree grows. The geometric mean has been used before and was suggested to us by Charniak [8]. We trained our PCFG on 280 files (approx. 150,000 words) from the Penn TreeBank, a set disjoint from the test sentences.

Once a sentence has been parsed, a parse-tree traversal algorithm searches the tree for the verb-object pairs that we are identifying. We used the same algorithm as was used to extract the reference set of verb-object pairs from the parsed TreeBank corpus. We then compare the extracted set of relationship

Metric	Our PCFG parser	Brill[1]	Charniak[2]	Collins[3]
Labeled Precision	71.0 %		81.6 %	84.9 %
Labeled Recall	67.4 %		80.0 %	84.9 %
Crossing Brackets Rate	80.5 %	83.8 %	87.8 %	

Table 3: PCFG Parser Evaluation Results

pairs against the reference set, computing both precision and recall. In order to complete the experiment in a reasonable amount of time, we tested only sentences with fewer than 30 tags. Parses could not be obtained for 3 of the 374 sentences parsed.

In order to validate our results it was necessary to show that our parser was sufficiently good to warrant its use. Toward that end we conducted a small evaluation of the parser alone and compared it with other parsers. Our evaluation consisted of computing parses for 131 sentences from the Penn Tree-Bank and comparing the results to the hand-corrected parser. Table 3 shows the results using the same grammar that we used in our PCFG experiment. The other columns show results for other parsers in similar evaluations on sentences of the same size. It is clear our parser is not among the best but we believe that its performance was sufficient to justify using it for our initial experiments. We performed no special tuning of the grammar and our training set size is significantly smaller than top-performing PCFG parsers. These factor probably contributed greatly to the difference in parser results. Throughout the course of our research we plan to assess the relationship between parser performance, as is indicated by simple evaluations such as this, and the contribution of the parser to identifying linguistic relationships.

4.2 Experiment 2

The second experiment (see Figure 6) used only partial-parsing information to locate verb-object pairs. After computing part-of-speech tags, just as in the PCFG experiment, noun and prepositional phrases were identified in each sentence using a partial parser. To find the noun and prepositional phrases, we used a simple parser with a conventional (non- probabilistic) grammar of non-recursive rules derived from the TreeBank. The parser worked left-to-right bottom-up, selecting the longest sequence that matched the rules in the grammar. Once these chunks had been identified, verb-object pairs were derived in

[1]Brill gives only results in terms of Accuracy = Crossing Brackets Rate, 83.8% is for sentences up to length 25 which is the shortest limit reported here but the longest tested by Brill.

[2]Charniak results are for sentences up to length 40 (similar to our evaluation) with a reduced grammar (comparable to ours)

[3]Collins does not give the same type of Crossing Brackets statistic. His results are the latest, and best.

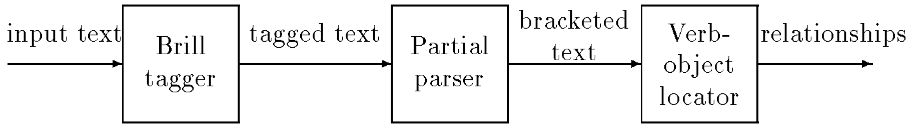

Figure 6: Partial Parser Experiment Architecture

a very simple way. The pair, a verb and a noun were selected when the noun was the last word in a noun phrase and the verb preceded the noun phrase without any "disallowed" intervening words. Disallowed words were those that were tagged as anything other than adverbs or particles.

The test set size, 676 sentences, includes sentences with more than 30 words or tokens. These we did not test with the PCFG parser because of its slow speed.

4.3 Comparing the results

As the Table 2 shows, there is not a significant difference in the overall results. Because the primary focus of this research is in obtaining high-precision information, further experiments will need to outline in more detail the relationship between precision and recall for the various parsing methods. Still, even with the short experiments we have done, we can look in detail at the mistakes each method made and outline a course for improving each as well as get some idea of where we are likely to have success in obtaining truly high precision.

We analyzed, in detail, the results for one file (approx. 50 sentences) from the Penn TreeBank. Some linguistic structures cause problems for both approaches. The misanalysis of quotative inversion structures causes the greatest number of errors for both experiments, accounting for 8 of the 23 incorrectly produced pairs for the PCFG approach and for 14 of 42 of incorrectly produced pairs for the partial-parsing approach.[4] Typical quotative inversion structures are shown here:

> "Now," says Joseph Napolitan, "the idea is to attack first ..."
>
> "Remember Pinocchio?" says a female voice.

Both algorithms incorrectly identified <says, Napolitan> and <says, voice> as verb-object pairs for these sentences. In general, noun-phrases that immediately follow verbs were interpreted in both experiments as objects. This also caused trouble for sentences that involved Extended-Case-Marking (ECM) such as:

[4] Note that the number of pairs differs widely between the two experiments even for a common file because a high-percentage of sentences in the TreeBank WSJ corpus are longer than the maximum size we tested in the PCFG experiment.

> The ad prompted Mr. Coleman, the former Virginia attorney general, to launch a series of advertisements ...
> Rabin did not want The President to invite Hussein to the meeting.

In each case the subject of the sentential complement is identified as the object of the main verb (i.e., <prompted, Coleman> and <want, President>).

These, and other problems, are a result of the unsophisticated model of verbal complements entailed in both models. One significant class of information missing from these algorithms is verb subcategorization information. Both algorithms would be improved by using detailed information, possibly specialized to individual corpora, regarding the verbs' potential complement structures and their relative frequency of usage. In the course of our research we will investigate how to incorporate verb-subcategorization information into our algorithms and its net effect on the identification of linguistic relationships.

Front-end processing errors also contributed to errors in extracting the verb-object relationships. Tagging mistakes caused inaccuracies for both algorithms as did the tagging procedure itself. Hyphenated words (e.g., "close-up", "cop-killer") are not tagged as single words but rather as three separate lexemes (e.g., **cop**/NN **-**/>HY **killer**/NN). Although we are not concerned ourselves with improving the tagging process, we would like to investigate the relationship between various tagging errors and the final precision of relationship identification.

Some classes of errors are more strongly evident in one experiment than the other. The partial parsing experiment showed numerous errors due to post-nominal gerundival complements such as:

> The attacks began when Democratic Rep. James Florio aired *an ad featuring a drawing of Pinocchio ...*

The algorithm incorrectly paired "aired" with "drawing" rather than with "ad" because of its simple expectation of the head-noun appearing last in a noun phrase (and the particular noun phrase found by our noun phrase parser, "an ad featuring a drawing"). More internal structure for the noun phrases will be needed to correct such mistakes.

A second phenomenon, isolated to the partial-parsing experiment, involves incorporating multiple noun-phrases together into single noun phrases:

> Mr. Wilder did introduce [NP such legislation 17 years] ago.
> There is [NP no evidence the barrels] were a serious threat.

The bracketing above shows the incorrect noun phrases found by the partial parser that prevent the correct head-noun from being identified. These errors are caused by a wide variety of syntactic structures such as sentential complement, temporal adjuncts, and small clauses. It is likely that several different approaches may be needed for the various roots causes of this inappropriate noun phrase combination. One initial idea would be to try to trim rules that lead to such inappropriate "glomming" from the NP grammar. It may suffice to use a better, more thoroughly proven noun-phrase chunker from other researchers [9, 3].

The PCFG failed in some cases because of attachment decisions. Since the PCFG parser had to construct a full parse, it was forced to make decisions with inadequate information. Additionally, for the PCFG experiment, there were a number of errors that had no single obvious problem. The output of the parser was so different from the correct parse that it is unclear what was the problem and what might be done to the parser to improve the results.

Despite the overall similarity of the results, there is some reason to believe the partial parser may be more appropriate for our framework. First of all, the relative speed of the two parsers indicates that the PCFG parser is unlikely to be able to process large corpora. Processing a large volume of text is one way we can compensate for the low recall rates typically associated with improved precision. Second, the results for individual sentences are more transparent for the partial-parsing approach than for the PCFG approach. That is, most individual errors in the partial-parsing experiment are traceable to a few potentially solvable problems but the PCFG experiment shows that some of its problems may be well buried in the statistics. For this reason, we believe that the symbolic information and statistics required for our framework will be more obvious if we use a partial parser. Lastly, we believe that the partial parsing approach will be easier to modify and extend with machine learning algorithms than will the PCFG parser. This may be key in allowing us to use our framework in wider types of text domains and literary styles.

5 Short-Term Plan

While we cannot articulate a complete and detailed plan that will take us from the present to our goal, we can outline the direction in which we are heading, the plan for collecting more information, and the way we will use our experimental results toward developing the framework we desire. The following list identifies particular experiments, or at least, areas of inquiry that we will look at in the upcoming months.

- Collins parser evaluation
- Additional relationships
- Precision/Recall parameterization

Following detailed discussions of each these items, we outline the overriding process that we will use to bring us closer to the framework that we seek.

5.1 Collins Parser

The bigram dependency parser of Collins focuses on low-level syntactic relationships using a new statistical model. Additionally, it incorporates partial parsing methods in that the task of finding the non-recursive core of noun phrases is performed separately. Because the Collins model focuses directly on lexical head relationships, we believe that this parser deserves its own evaluation, to be compared with the work already presented.

5.2 More relationships

The experiments conducted so far focus on a single relationship, that of the verb-object heads. To fully develop our framework we will have to greatly expand the range of relationships we test. We need to look at other relationships that have local syntactic clues such as subject-verb relationships and also at relationships that are neither as local nor as strongly syntactic in their resolution such as prepositional phrase relationships and relationships of predication. The following list identifies a few specific relationships that we may study in the coming months.

syntactic-subject/passive-verb a continuation of the study of verb-object forms.

noun/of-object a simple introduction to prepositional phrase study.

noun-or-verb/in-object "in" is a preposition with often ambiguous attachment whose resolution will require a larger context and semantic information.

modifier/modifiee a largely syntactic relationship with importance to IR systems.

agent/action involves subjects of both noun and verb phrases; this relationship will require some semantic restrictions.

5.3 Parameterization

Another important analysis for us will involve modifying our existing work so that we can see how precision and recall trade-off with one another. This will involve parameterizing the algorithms to weaken recall to see how high we can push the precision. While our focus is clearly on high-precision identification, there may also be need of higher-recall identification that yields more modest precision. If our work can be parameterized to return relationships only if a certain confidence level is reached, IR systems and other applications might ask for successively lower precision results after the results (or lack of results) of the high-precision identification are processed.

5.4 Toward a Framework

Clearly, we can conduct experiments such as we have outlined but the question may be asked, "How do these things contribute to progress toward the desired framework?" There is more than one answer. In one view, we are conducting an evaluation of known NLP techniques that vary along many dimensions. The following list identifies a view:

- Fast versus slow
- Stochastic versus inductive
- Use of global versus local information
- Optimization of global versus local metrics

By conducting numerous experiments that vary along these dimensions we aim to eliminate areas of this "method space" and focus in on what is most effective for our task.

A second way in which these experiments carry us forward is that they can help us to isolate missing information. Currently we use only simple syntactic information from a single corpus in identifying relationships. As the experiments have already shown, other information would be useful, such as verb-subcategorization patterns and statistics. Semantic class information may be helpful in identifying relationships involving prepositional phrase attachments. Through our experiments we hope to find and categorize the type of information we need and then extract it from on-line sources as best we can. Existing text collections are one source of information, as are existing packages such as on-line dictionaries and semantic nets.

Lastly we will use the experiments as a means of seeing the variability of our task across collections and text domains. A general framework has to be flexible in order to be applied widely and for this, we will need to understand how the relationships and their clues vary across collections. This is important in two distinct ways. First, it allows us to isolate the most consistent evidence for each kind of relationships. Second, because we have a variety of algorithms from the NLP and Machine Learning literature at our disposal, detailed knowledge of the available linguistic clues help us select the most appropriate algorithms.

References

[1] Steven Abney. Parsing By Chunks. In Robert Berwick and Steven Abney and Carol Tenny, editor, *Principle-Based Parsing*. Kluwer Academic Publishers, 1991.

[2] Steven Abney. Chunks and dependencies: Bringing processing evidence to bear on syntax. In Jennifer Cole and Georgia Green and Jerry Morgan, editor, *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164. CSLI, 1995.

[3] Steven Abney. Part-of-Speech Tagging and Partial Parsing. In Ken Church and Steve Young and Gerrit Bloothooft, editor, *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, Dordrecht, 1996.

[4] Steven Abney. Partial Parsing via Finite-State Cascades. In John Carroll, editor, *Workshop on Robust Parsing*, pages 8–15, 1996.

[5] James Allen. *Natural Language Understanding.* Benjamin/Cummings, Second edition, 1995.

[6] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing.* Association for Computational Linguistics, 1992.

[7] Eric Brill. Automatic Grammar Induction and Parsing Free Text: A Transformation-Based Approach. In *Proceedings of the 31th Annual Meeting of the ACL.* Association for Computational Linguistics, 1993.

[8] Eugene Charniak. Tree-bank Grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996.

[9] Ken Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143. Association for Computational Linguistics, 1988.

[10] Michael John Collins. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 184–191, San Francisco, CA, 1996. Morgan Kaufmann.

[11] R. Garside, G. Leech, and G. Sampson. *Computational Analysis of English: A Corpus-based Approach.* Longman, London, U.K., 1987.

[12] D. Hindle. Acquiring Disambiguation Rules from Text. In *Proceedings of the 27th Annual Meeting of the ACL*, pages 118–125, Vancouver, 1989. Association for Computational Linguistics.

[13] Julian Kupiec. Murax: A Robust Linguistic Approach for Question Answering Using an On-line Encyclopedia. In *SIGIR-93*, Philadelphia, PA, 1993. Association for Computing Machinery, Inc.

[14] W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. University of Massachusetts: Description of the CIRCUS System as Used in MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, San Mateo, CA, 1992. Morgan Kaufmann.

[15] W. Lehnert, C. Cardie, D. Fisher, E. Riloff, and R. Williams. University of Massachusetts: Description of the CIRCUS System as Used in MUC-3. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, pages 223–233, San Mateo, CA, 1991. Morgan Kaufmann.

[16] David M. Magerman. Statistical Decision-Tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 276–283. Association for Computational Linguistics, 1995.

[17] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–245, 1990.

[18] *Proceedings of the Third Message Understanding Conference (MUC-3)*. Morgan Kaufmann, San Mateo, CA, 1991.

[19] *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann, San Mateo, CA, 1992.

[20] Ellen Riloff. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816, Washington, DC, 1993. AAAI Press / MIT Press.

[21] Ian C. Ross and John W. Tukey. Introduction to these volumes. In *Index to Statistics and Probability*. The R&D Press, Los Altos, CA, 1975.