Anaphora for Everyone: Pronominal Anaphora Resolution without a Parser

Christopher Kennedy
Board of Studies in Linguistics
University of California
Santa Cruz, CA 95064
kennedy@ling.ucsc.edu

Branimir Boguraev
Advanced Technologies Group
Apple Computer, Inc.
Cupertino, CA 95014
bkb@apple.com

Abstract

We present an algorithm for anaphora resolution which is a modified and extended version of that developed by (Lappin and Leass, 1994). In contrast to that work, our algorithm does not require in-depth, full, syntactic parsing of text. Instead, with minimal compromise in output quality, the modifications enable the resolution process to work from the output of a part of speech tagger, enriched only with annotations of grammatical function of lexical items in the input text stream. Evaluation of the results of our implementation demonstrates that accurate anaphora resolution can be realized within natural language processing frameworks which do not—or cannot—employ robust and reliable parsing components.

1 Overview

(Lappin and Leass, 1994) describe an algorithm for pronominal anaphora resolution with high rate of correct analyses. While one of the strong points of this algorithm is that it operates primarily on syntactic information alone, this also turns out to be a limiting factor for its wide use: current state-of-the-art of practically applicable parsing technology still falls short of robust and reliable delivery of syntactic analysis of real texts to the level of detail and precision that the filters and constraints described by Lappin and Leass assume.

We are particularly interested in a class of text processing applications, capable of delivery of content analysis to a depth involving non-trivial amount of discourse processing, including anaphora resolution. The operational context prohibits us from making any assumptions concerning domain, style, and genre of input; as a result, we have developed a text processing framework which builds its capabilities entirely on the basis of a considerably shallower linguistic analysis of the input stream, thus trading off depth of base level analysis for breadth of coverage.

In this paper, we present work on modifying the Lappin/Leass algorithm in a way which enables it to work off a flat morpho-syntactic analysis of the sentences of a text, while retaining a degree of quality and accuracy in pronominal anaphora resolution comparable to that reported in (Lappin and Leass, 1994). The modifications discussed below make the algorithm available to a wide range of text processing frameworks, which, due to the lack of full syntactic parsing capability, normally would have been unable to use this high precision anaphora resolution tool. The work is additionally important, we feel, as it shows that information about the content and logical structure of a text, in principle a core requirement for higher level semantic and discourse processes, can be effectively approximated by the right mix of constituent analysis and inferences about functional relations.

2 General outline of the algorithm

The base level linguistic analysis for anaphora resolution is the output of a part of speech tagger, augmented with syntactic function annotations for each input token; this kind of analysis is generated by the morphosyntactic tagging system described in (Voutilainen et al., 1992), (Karlsson et al., 1995) (henceforth LINGSOFT). In addition to extremely high levels of accuracy in recall and precision of tag assignment ((Voutilainen et al., 1992) report 99.77% overall recall and 95.54% overall precision, over a variety of text genres, and in comparison with other state-of-the-art tagging systems), the primary motivation for adopting this system is the requirement to develop a robust text processor—with anaphora resolution being just one of its discourse analysis functions—capable of reliably handling arbitrary kinds of input.

The tagger provides a very simple analysis of the structure of the text: for each lexical item in each sentence, it provides a set of values which indicate the morphological, lexical, grammatical and syntactic features of the item in the context in which it appears. In addition, the modified algorithm we present requires annotation of the input text stream by a simple position-identification function which associates an integer with each token in a text sequentially (we will refer to a token's integer value as its *offset*).

As an example, given the text

> "For 1995 the company set up its headquarters in Hall 11, the newest and most prestigious of CeBIT's 23 halls."

the anaphora resolution algorithm would be presented with the following analysis stream. Note, in particular, the grammatical function information (e.g., @SUBJ, @+FMAINV) and the integer values (e.g., "off139") associated with each token.

```
"For/off139" "for" PREP @ADVL
"1995/off140" "1995" NUM CARD @<P
"the/off141" "the" DET CENTRAL ART SG/PL @DN>
"company/off142" "company" N NOM SG/PL @SUBJ
"set/off143" "set" V PAST VFIN @+FMAINV
"up/off144" "up" ADV ADVL @ADVL
"its/off145" "it" PRON GEN SG3 @GN>
"headquarters/off146" "headquarters" N NOM SG/PL @OBJ
"in/off147" "in" PREP @<NOM @ADVL
"Hall/off148" "hall" N NOM SG  @NN>
"11/off149" "11" NUM CARD @<P
"$,/off150" "," PUNCT
"the/off151" "the" DET CENTRAL ART SG/PL @DN>
"newest/off152" "new" A SUP @PCOMPL-O
"and/off153" "and" CC @CC
"most/off154" "much" ADV SUP @AD-A>
"prestigious/off155" "prestigious" A ABS @<P
"of/off156" "of" PREP @<NOM-OF
"CeBIT's/off157" "cebit" N GEN SG @GN>
"23/off158" "23" NUM CARD @QN>
"halls/off159" "hall" N NOM PL @<P
"$./off160" "." PUNCT
```

2.1 Data collection

Although LINGSOFT does not provide specific information about constituent structure, partial constituency—specifically, identification of sequences of tokens as phrasal units—can be inferred from the analysis by running the tagged text through a set of filters, which are stated as regular expressions over metatokens such as the ones illustrated above.

For the purposes of anaphora resolution, the primary data set consists of a complete listing of all noun phrases, reduced to modifier-head sequences. This data set is obtained by means of a phrasal grammar whose patterns characterize the composition of a noun phrase (NP) in terms of possible token sequences. The output of NP identification is a set of token/feature matrix/offset sequences, where offset value is determined by the offset of the first token in the sequence. The offset indicates the position of the NP in the text, and so provides crucial information about precedence relations.

A secondary data set consists of observations about the syntactic contexts in which the NPs identified by the phrasal grammar appear. These observations are derived using a set of patterns designed to detect nominal sequences in two subordinate syntactic environments: containment in an adverbial adjunct and containment in an NP (i.e., containment in a prepositional or clausal complement of a noun, or containment in a relative clause). This is accomplished by running a set of patterns which identify NPs that occur locally to adverbs, relative pronouns, and *noun-preposition* or *noun-complementizer* sequences over the tagged text in conjunction with the basic NP patterns described above. Because the syntactic patterns are stated as regular expressions, misanalyses are inevitable. In practice, however, the extent to which incorrect analyses of syntactic context affect the overall accuracy of the algorithm is not large; we will return to a discussion of this point in section 4.

A third set of patterns identifies and tags occurrences of "expletive" *it*. These patterns target occurrences of the pronoun *it* in certain contexts, e.g., as the subject of members of a specific set of verbs (*seem*, *appear*, etc.), or as the subject of adjectives with clausal complements.

Once the extraction procedures are complete and the results unified, a set of *discourse referents*—abstract objects which represent the participants in the discourse—is generated from the set of NP observations. A particularly convenient implementation of discourse referents is to represent them as objects in the Common Lisp Object System, with slots which encode the following information parameters (where ADJUNCT and EMBED indicate whether a discourse referent was observed in either of the two syntactic contexts discussed above):

TEXT:	text form
TYPE:	referential type (e.g., REF, PRO, RFLX)
AGR:	person, number, gender
GFUN:	grammatical function
ADJUNCT:	T or NIL
EMBED:	T or NIL
POS:	text position

Note that each discourse referent contains information about itself and the context in which it appears, but the only information about its relation to other discourse referents is in the form of precedence relations (as determined by text position). The absence of explicit information about configurational relations marks the crucial difference between our algorithm and the Lappin/Leass algorithm. (Lappin and Leass, 1994) use configurational information in two ways: as a factor in the determination of the salience of a discourse referent (discussed below), and as input to a set of disjoint reference filters. Our implementation seeks to perform exactly the same tasks by inferring hierarchical relations from a less rich base. The modifications and assumptions required to accomplish this goal will be highlighted in the following discussion.

2.2 Anaphora resolution

Once the representation of the text has been recast as a set of discourse referents (ordered by offset value), it is sent to the anaphora resolution algorithm proper. The basic logic of the algorithm parallels that of the Lappin/Leass algorithm. The interpretation procedure involves moving through the text sentence by sentence and interpreting the discourse referents in each sentence from left to right. There are two possible interpretations of a discourse referent: either it is taken to introduce a new participant in the discourse, or it is taken to refer to a previously interpreted discourse referent. *Coreference* is determined by first eliminating from consideration those discourse referents to which an anaphoric expression *cannot* possibly refer, then selecting the optimal antecedent from the candidates that remain, where optimality is determined by a *salience* measure.

In order to present the details of anaphora resolution, we define below our notions—and implementations—of coreference and salience.

2.2.1 Coreference

As in the Lappin and Leass algorithm, the anaphor-antecedent relation is established between two discourse referents (cf. (Heim, 1982), (Kamp, 1981)), while the more general notion of coreference is represented in terms of equivalence classes of anaphorically related discourse referents, which we will refer to as "COREF classes". Thus, the problem of interpreting an anaphoric expression boils down to the problem of establishing an anaphoric link between the anaphor and some previously interpreted discourse referent (possibly another anaphor); a consequence of establishing this link is that the anaphor becomes a member of the COREF class already associated with its antecedent.

In our implementation, COREF classes are represented as objects in the Common Lisp Object System which contain information about the COREF class as a whole, including canonical form (typically determined by the discourse referent which introduces the class), membership, and, most importantly, salience (discussed below).[1] The connection between a discourse referent and its COREF class is mediated through the COREF object as follows: every discourse referent includes an information parameter which is a pointer to a COREF object; discourse referents which have been determined to be coreferential share the same COREF value (and so literally point to the same object). Implementing coreference in this way provides a means of getting from any discourse referent in a COREF class to information about the class as a whole.

2.2.2 Salience

The information parameter of a COREF object most crucial to anaphora resolution is its salience, which is determined by the status of the members of the COREF class it represents with respect to 10 contextual, grammatical, and syntactic constraints. Following (Lappin and Leass, 1994), we will refer to these constraints as "salience factors". Individual salience factors are associated with numerical values; the overall salience, or "salience weight" of a COREF is the sum of the values of the salience factors that are satisfied by some member of the COREF class (note that values may be satisfied at most once by each member of the class). The salience factors used by our algorithm are defined below with their values. Our salience factors mirror those used by (Lappin and Leass, 1994), with the exception of POSS-S, discussed below, and CNTX-S, which is sensitive to the *context* in which a discourse referent appears, where a context is a topically coherent segment of text, as determined by a text-segmentation algorithm which follows (Hearst, 1994).

SENT-S:	100 iff in the current sentence
CNTX-S:	50 iff in the current context
SUBJ-S:	80 iff GFUN = *subject*
EXST-S:	70 iff in an existential construction
POSS-S:	65 iff GFUN = *possessive*
ACC-S:	50 iff GFUN = *direct object*
DAT-S:	40 iff GFUN = *indirect object*
OBLQ-S:	30 iff the complement of a preposition
HEAD-S:	80 iff EMBED = NIL
ARG-S:	50 iff ADJUNCT = NIL

Note that the values of salience factors are arbitrary; what is crucial, as pointed out by (Lappin and Leass, 1994), is the relational structure imposed on the factors by these values. The relative ranking of the factors is justified both linguistically, as a reflection of the role of the functional hierarchy in determining anaphoric relations (cf. (Keenan and Comrie, 1977)), as well as by experimental results—both Lappin and Leass' and our own. For all factors except CNTX-S and POSS-S, we adopt the values derived from a series of experiments described in (Lappin and Leass, 1994) which used different settings to determine the relative importance of each factor as a function of the overall success of the algorithm. Our values for CNTX-S and POSS-S were determined using similar tests.

An important feature of our implementation of salience, following that of Lappin and Leass, is that it is variable: the salience of a COREF class decreases and increases according to the frequency of reference to the class. When an anaphoric link is established between a pronoun and a previously introduced discourse referent, the pronoun is added to the COREF class associated with the discourse referent, its COREF value is set to the COREF value of the antecedent (i.e., to the COREF object which represents the class), and the salience of the COREF object is recalculated according to how the new member satisfies the set of salience factors. This final step raises the overall salience of the COREF, since the new member will minimally satisfy SENT-S and CNTX-S.

Salience is not stable, however: in order to realistically represent the local prominence of discourse referents in a text, a decay function is built into the algorithm, so that salience weight decreases over time. If new members are not added, the salience weight of a COREF eventually reduces to zero. The consequence of this variability in salience is that a very general heuristic for anaphora resolution is established: resolve a pronoun to the most salient candidate antecedent.

2.2.3 Interpretation

As noted above, in terms of overall strategy, the resolution procedure follows that of Lappin and Leass. The first step in interpreting the discourse referents in a new sentence is to decrease the salience weights of the COREF classes that have already been established by a factor of two. Next, the algorithm locates all non-anaphoric discourse referents in the sentence under consideration, generates a new COREF class for each one, and calculates its salience weight according to how the discourse referent satisfies the set of salience factors.

The second step involves the interpretation of lexical anaphors (reflexives and reciprocals). A list of candidate antecedent-anaphor pairs is generated for every lexical anaphor, based on the hypothesis that a lexical anaphor must refer to a coargument. In the absence of configurational information, coarguments are identified using grammatical function information (as determined by LINGSOFT) and precedence relations. A reflexive can have one of three possible grammatical function values: *direct object*, *indirect object*, or *oblique*. In the first case, the closest preceding discourse referent with grammatical function value *subject* is identified as a possible antecedent. In the latter cases, both the closest preceding subject and the closest preceding direct object that is not separated from the anaphor by a subject are identified as possible antecedents. If more than one possible antecedent is located for a lexical anaphor, the one with the highest salience weight is determined to be the actual antecedent. Once an antecedent has been located, the anaphor is added to the COREF class associated with the antecedent, and the salience of the COREF class is recalculated accordingly.

The final step is the interpretation of pronouns. The basic resolution heuristic, as noted above, is quite simple: generate a set of candidate antecedents, then establish coreference with the candidate which has the greatest salience weight (in the event of a tie, the closest candidate is chosen). In order to generate the candidate set, however, those discourse referents with which

[1] The implementation of a COREF object needs to be aware of potential circularities, thus a COREF does not actually contain its member discourse referents, but rather a listing of their offsets.

a pronoun *cannot* refer must be eliminated from consideration. This is accomplished by running the overall candidate pool (the set of interpreted discourse referents whose salience values exceed an arbitrarily set threshold) through two sets of filters: a set of morphological agreement filters, which eliminate from consideration any discourse referent which disagrees in person, number, or gender with the pronoun, and a set of disjoint reference filters.

The determination of disjoint reference represents a significant point of divergence between our algorithm and the Lappin/Leass algorithm, because, as is well known, configurational relations play a prominent role in determining which constituents in a sentence a pronoun may refer to. Three conditions are of particular relevance to the anaphora resolution algorithm:

Condition 1: A pronoun cannot corefer with a coargument.

Condition 2: A pronoun cannot corefer with a nonpronominal constituent which it both commands and precedes.

Condition 3: A pronoun cannot corefer with a constituent which contains it.

In the absence of configurational information, our algorithm relies on inferences from grammatical function and precedence to determine disjoint reference. In practice, even without accurate information about constituent structure, the syntactic filters described below are extremely accurate (see the discussion of this point in section 4).

Condition 1 is implemented by locating all discourse referents with GFUN value *direct object*, *indirect object*, or *oblique* which follow a pronoun with GFUN value *subject* or *direct object*, as long as no subject intervenes (the hypothesis being that a subject indicates the beginning of the next clause). Discourse referents which satisfy these conditions are identified as disjoint.

Condition 2 is implemented by locating for every non-adjunct and non-embedded pronoun the set of non-pronominal discourse referents in its sentence which follow it, and eliminating these as potential antecedents. In effect, the command relation is inferred from precedence and the information provided by the syntactic patterns: an argument which is neither contained in an adjunct nor embedded in another nominal commands those expressions which it precedes.

Condition 3 makes use of the observation that a discourse referent contains every object to its right with a non-nil EMBED value. The algorithm identifies as disjoint a discourse referent and every pronoun which follows it and has a non-nil EMBED value, until a discourse referent with EMBED value NIL is located (marking the end of the containment domain). Condididiton 3 also rules out coreference between a genitive pronoun and the NP it modifies.

After the morphological and syntactic filters have been applied, the set of discourse referents that remain constitute the set of candidate antecedents for the pronoun. The candidate set is subjected to a final evaluation procedure which performs two functions: it decreases the salience of candidates which the pronoun precedes (cataphora is penalized), and it increases the salience of candidates which satisfy either a locality or a parallelism condition (described below), both of which apply to intrasentential candidates.

The locality heuristic is designed to negate the effects of subordination when both candidate and anaphor appear in the same subordinate context, the assumption being that the prominence of a candidate should be determined with respect to the position of the anaphor. This is a point of difference between our algorithm and the one described in (Lappin and Leass, 1994). The salience of a candidate which is determined to be in the same subordinate context as a pronoun (determined as a function of precedence relations and EMBED and ADJUNCT values) is temporarily increased to the level it would have were the candidate *not* in the subordinate context; the level is returned to normal after the anaphor is resolved.

The parallelism heuristic rewards candidates which are such that the pair consisting of the GFUN values of candidate and anaphor are identical to GFUN values of a previously identified anaphor-antecedent pair. This parallelism heuristic differs from a similar one used by the Lappin/Leass algorithm, which rewards candidates whose grammatical function is identical to that of an anaphor.

Once the generation and evaluation of the candidate set is complete, the candidates are ranked according to salience weight, and the candidate with the highest salience weight is determined to be the antecedent of the pronoun under consideration. In the event of a tie, the candidate which most immediately precedes the anaphor is selected as the antededent (where precedence is determined by comparing offset values). The COREF value of the pronoun is set to that of the antecedent, adding it to the the antecedent's COREF class, and the salience of the class is recalculated accordingly.

3 Example output

The larger context from which the sample analysis in the beginning of Section 2 was taken is as follows:

> "...while Apple and its PowerPC partners claimed some prime real estate on the show floor, Apple's most interesting offerings debuted behind the scenes. Gone was the narrow corner booth that Apple shoehorned *its* products into last year. For 1995 the company set up *its* headquarters in Hall 11, the newest and most prestigious of CeBIT's 23 halls."

The anaphora resolution algorithm generates the following analysis for the first italicized pronoun. For each candidate,[2] the annotation in square brackets indicates its offset value, and the number to the right indicates its salience weight at the point of interpretation of the pronoun.

```
ANA: its                [@off/133]
CND: Apple              [@off/131]      432
     Apple              [@off/101]      352
     its                [@off/103]      352
     Apple's            [@off/115]      352
     prime real_estate  [@off/108]      165
     show floor         [@off/112]      155
     year               [@off/137]     310/3
```

The candidate set illustrates several important points. First, the equality in salience weights of the candidates at offsets 101, 103, and 115 is a consequence of

[2] Note that our syntactic filters are quite capable of discarding a number of configurationally inappropriate antecedents, which appear to satisfy the precedence relation.

the fact that these discourse referents are members of the same COREF class. Their unification into a single class indicates both successful anaphora resolution (of the pronoun at offset 103), as well as the operation of higher-level discourse processing designed to identify *all* references to a particular COREF class, not just the anaphoric ones (cf. (Kennedy and Boguraev, 1996)). The higher salience of the optimal candidate—which is also a member of this COREF class—shows the effect of the locality heuristic described in section 2.2.3. Both the pronoun and the candidate appear in the same subordinate context (within a relative clause); as a result, the salience of the candidate (but not of the class to which it belongs) is temporarily boosted to negate the effect of subordination.

An abbreviated candidate set for the second italicized pronoun is given below:

```
ANA: its                    [@off/145]
CND: company                [@off/142]      360
     Apple                  [@off/131]      192
     its                    [@off/133]      192
```

This set is interesting because it illustrates the prominent role of SENT-S in controlling salience: *company* is correctly identified as the antecedent of the pronoun, despite the frequency of mention of members of the COREF class containing *Apple* and *its*, because it occurs in the same sentence as the anaphor. Of course, this example also indicates the need for additional heuristics designed to connect *company* with *Apple*, since these discourse referents clearly make reference to the same object. We are currently working towards this goal; see (Kennedy and Boguraev, 1996) for discussion.

The following text segment illustrates the resolution of intersentential anaphora.

> "Sun's prototype Internet access device uses a 110-Mhz MicroSPARCprocessor, and is diskless. *Its* dimensions are 5.5 inches x 9 inches x 2inches."

```
ANA: Its                        [@off/347]
CND: Internet access device     [@off/335] 180
     MicroSPARCprocessor        [@off/341] 165
     Sun's                      [@off/333] 140
```

The first sentence in this fragment introduces three discourse referents bearing different grammatical functions, none of which appear in subordinate contexts. Since the sentence in which the anaphor occurs does not contain any candidates (the discourse referent introduced by *dimensions* is eliminated from consideration by both the morphological and disjoint reference filters), only those from the previous sentence are considered (each is compatible with the morphological requirements of the anaphor). These are ranked according to salience weight, where the crucial factor is grammatical function value. The result of the ranking is that *Internet access device*—the candidate which satisfies the highest-weighted salience factor, SUBJ-S—is the optimal candidate, and so correctly identified as the antecedent.

4 Evaluation

Quantitative evaluation shows the anaphora resolution algorithm described here to run at a rate of 75% accuracy. The data set on which the evaluation was based consisted of 27 texts, taken from a random selection of genres, including press releases, product announcements, news stories, magazine articles, and other documents existing as World Wide Web pages. Within these texts, we counted 306 third person anaphoric pronouns; of these, 231 were correctly resolved to the discourse referent identified as the antecedent by the first author.[3] This rate of accuracy is clearly comparable to that of the Lappin/Leass algorithm, which (Lappin and Leass, 1994) report as 85%.

Several observations about the results and the comparison with (Lappin and Leass, 1994) are in order. First, and most obviously, some deterioration in quality is to be expected, given the relatively impoverished linguistic base we start with.

Second, it is important to note that this is not just a matter of simple comparison. The results in (Lappin and Leass, 1994) describe the output of the procedure applied to a single text genre: computer manuals. Arguably, this is an example of a particularly well behaved text; in any case, it is not clear how the figure would be normalized over a wide range of text types, some of them not completely 'clean', as is the case with our data.

Third, close analysis of the most common types of error our algorithm currently makes reveals two specific configurations in the input which confuse the procedure and contribute to the error rate: gender mismatch (35% of errors) and certain long range contextual (stylistic) phenomena, best exemplified by text containing quoted passages in-line (14% of errors).

Implementing a gender (dis-)agreement filter is not technically complex; as noted above, the current algorithm contains one. The persistence of gender mismatches in the output simply reflects the lack of a consistent gender slot in the LINGSOFT tagger output. Augmenting the algorithm with a lexical database which includes more detailed gender information will result in improved accuracy.

Ensuring proper interpretation of anaphors both within and outside of quoted text requires, in effect, a method of evaluating quoted speech separately from its surrounding context. Although a complex problem, we feel that this is possible, given that our input data stream embodies a richer notion of position and context, as a result of an independent text segmentation procedure adapted from (Hearst, 1994) (and discussed above in section 2.2.2).

What is worth noting is the small number of errors which can be directly attributed to the absence of configurational information. Of the 75 misinterpreted pronouns, only 2 involved a failure to establish configurationally determined disjoint reference (both of these involved Condition 3), and only an additional several errors could be unambiguously traced to a failure to correctly identify the syntactic context in which a discourse referent appeared (as determined by a misfire of the salience factors sensitive to syntactic context, HEAD-S and ARG-S).

Overall, these considerations lead to two conclusions. First, with the incorporation of more explicit morphological and contextual information, it should

[3] The set of 306 "anaphoric" pronouns excluded 30 occurrences of "expletive" *it* not identified by the expletive patterns (primarily occurrences in object position), as well as 6 occurrences of *it* which referred to a VP or propositional constituent. We are currently refining the existing expletive patterns for improved accuracy.

be possible to increase the overall quality of our output, bringing it much closer in line with Lappin and Leass' results. Again, straight comparison would not be trivial, as e.g. quoted text passages are not a natural part of computer manuals, and are, on the other hand, an extremely common occurrence in the types of text we are dealing with.

Second, and most importantly, the absence of explicit configurational information does *not* result in a substantial degradation in the accuracy of an anaphora resolution algorithm that is otherwise similar to that described in (Lappin and Leass, 1994).

5 Conclusion

Lappin and Leass' algorithm for pronominal anaphora resolution is capable of high accuracy, but requires in-depth, full, syntactic parsing of text. The modifications of that algorithm that we have developed make it available to a larger set of text processing frameworks, as we assume a considerably 'poorer' analysis substrate. While adaptations to the input format and interpretation procedures have necessarily addressed the issues of coping with a less rich level of linguistic analysis, there is only a small compromise in the quality of the results. Our evaluation indicates that the problems with the current implementation do not stem from the absence of a parse, but rather from factors which can be addressed within the constraints imposed by the shallow base analysis. The overall success of the algorithm is important, then, not only for the immediate utility of the particular modifications, but also because the strategy we have developed for circumventing the need for full syntactic analysis is applicable to other interpretation tasks which, like the problem of anaphora resolution, lie in the space of higher level semantic and discourse analysis.

References

Marti Hearst. 1994. Multi-paragraph segmentation of expository text. In *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico. Association for Computational Linguistics, Morristown, New Jersey.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases.* Doctoral dissertation, University of Massachusetts, Amherst.

Hans Kamp. 1981. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof (eds.), *Formal Methods in the Study of Language.* Mathematisch Centrum Tracts, Amsterdam.

Fred Karlsson, Atro Voutilainen, Juha Heikkila, and Arto Antilla. 1995. *Constraint grammar: A language-independent system for parsing free text.* Mouton de Gruyter, Berlin/New York.

Edward Keenan and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8:62–100.

Christopher Kennedy and Branimir Boguraev. 1996. Anaphora in a wider context: Tracking discourse referents. In W. Wahlster (ed.), *12th European Conference on Artificial Intelligence*. John Wiley and Sons, Ltd, London/New York.

Shalom Lappin and Herb Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Atro Voutilainen, Juha Heikkila, and Arto Antilla. 1992. *A constraint grammar of English: A performance-oriented approach.* University of Helsinki, Publication No. 21, Helsinki, Finland.