

# A LANGUAGE-INDEPENDENT ANAPHORA RESOLUTION SYSTEM FOR UNDERSTANDING MULTILINGUAL TEXTS

Chinatsu Aone and Douglas McKee  
Systems Research and Applications (SRA)  
2000 15th Street North  
Arlington, VA 22201  
aonec@sra.com, mckeed@sra.com

## Abstract

This paper describes a new discourse module within our multilingual NLP system. Because of its unique data-driven architecture, the discourse module is language-independent. Moreover, the use of hierarchically organized multiple knowledge sources makes the module robust and trainable using discourse-tagged corpora. Separating discourse phenomena from knowledge sources makes the discourse module easily extensible to additional phenomena.

## 1 Introduction

This paper describes a new discourse module within our multilingual natural language processing system which has been used for understanding texts in English, Spanish and Japanese (cf. [1, 2]).<sup>1</sup> The following design principles underlie the discourse module:

- **Language-independence:** No processing code depends on language-dependent facts.
- **Extensibility:** It is easy to handle additional phenomena.
- **Robustness:** The discourse module does its best even when its input is incomplete or wrong.
- **Trainability:** The performance can be tuned for particular domains and applications.

In the following, we first describe the architecture of the discourse module. Then, we discuss how its performance is evaluated and trained using discourse-tagged corpora. Finally, we compare our approach to other research.

<sup>1</sup>Our system has been used in several data extraction tasks and a prototype machine translation system.

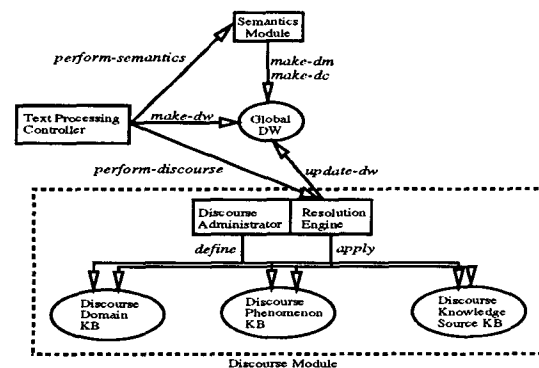


Figure 1: Discourse Architecture

## 2 Discourse Architecture

Our discourse module consists of two discourse processing submodules (the *Discourse Administrator* and the *Resolution Engine*), and three discourse knowledge bases (the *Discourse Knowledge Source KB*, the *Discourse Phenomenon KB*, and the *Discourse Domain KB*). The Discourse Administrator is a development-time tool for defining the three discourse KB's. The Resolution Engine, on the other hand, is the run-time processing module which actually performs anaphora resolution using these discourse KB's.

The Resolution Engine also has access to an external discourse data structure called the *global discourse world*, which is created by the top-level text processing controller. The global discourse world holds syntactic, semantic, rhetorical, and other information about the input text derived by other parts of the system. The architecture is shown in Figure 1.

### 2.1 Discourse Data Structures

There are four major discourse data types within the global discourse world: Discourse World (DW), Dis-

course Clause (DC), Discourse Marker (DM), and File Card (FC), as shown in Figure 2.

The *global discourse world* corresponds to an entire text, and its *sub-discourse worlds* correspond to sub-components of the text such as paragraphs. Discourse worlds form a tree representing a text's structure.

A *discourse clause* is created for each syntactic structure of category S by the semantics module. It can correspond to either a full sentence or a part of a full sentence. Each discourse clause is typed according to its syntactic properties.

A *discourse marker* (cf. Kamp [14], or "discourse entity" in Ayuso [3]) is created for each noun or verb in the input sentence during semantic interpretation. A discourse marker is static in that once it is introduced to the discourse world, the information within it is never changed.

Unlike a discourse marker, a *file card* (cf. Heim [11], "discourse referent" in Karttunen [15], or "discourse entity" in Webber [19]) is dynamic in a sense that it is continually updated as the discourse processing proceeds. While an indefinite discourse marker starts a file card, a definite discourse marker updates an already existing file card corresponding to its antecedent. In this way, a file card keeps track of all its co-referring discourse markers, and accumulates semantic information within them.

## 2.2 Discourse Administrator

Our discourse module is customized at development time by creating and modifying the three discourse KB's using the Discourse Administrator. First, a *discourse domain* is established for a particular NLP application. Next, a set of *discourse phenomena* which should be handled within that domain by the discourse module is chosen (e.g. definite NP, 3rd person pronoun, etc.) because some phenomena may not be necessary to handle for a particular application domain. Then, for each selected discourse phenomenon, a set of *discourse knowledge sources* are chosen which are applied during anaphora resolution, since different discourse phenomena require different sets of knowledge sources.

### 2.2.1 Discourse Knowledge Source KB

The discourse knowledge source KB houses small well-defined anaphora resolution strategies. Each knowledge source (KS) is an object in the hierarchically organized KB, and information in a specific KS can be inherited from a more general KS.

There are three kinds of KS's: a generator, a filter and an orderer. A *generator* is used to generate pos-

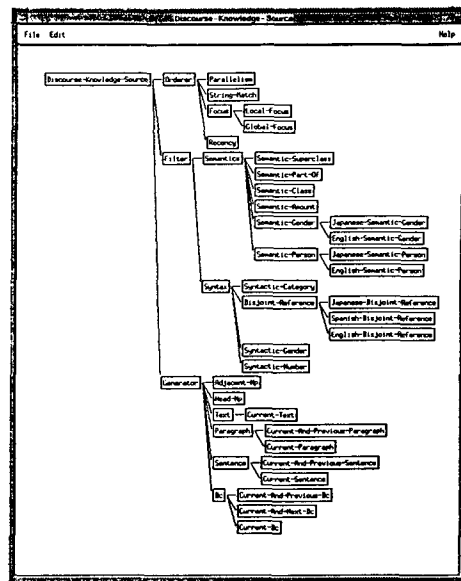


Figure 3: Discourse Knowledge Source KB

sible antecedent hypotheses from the global discourse world. Unlike other discourse systems, we have multiple generators because different discourse phenomena exhibit different antecedent distribution patterns (cf. Guindon *et al.* [10]). A *filter* is used to eliminate impossible hypotheses, while an *orderer* is used to rank possible hypotheses in a preference order. The KS tree is shown in Figure 3.

Each KS contains three slots: *ks-function*, *ks-data*, and *ks-language*. The *ks-function* slot contains a functional definition of the KS. For example, the functional definition of the Syntactic-Gender filter defines when the syntactic gender of an anaphor is compatible with that of an antecedent hypothesis. A *ks-data* slot contains data used by *ks-function*. The separation of data from function is desirable because a parent KS can specify *ks-function* while its sub-KS's inherit the same *ks-function* but specify their own data. For example, in languages like English and Japanese, the syntactic gender of a pronoun imposes a semantic gender restriction on its antecedent. An English pronoun "he", for instance, can never refer to an NP whose semantic gender is female like "Ms. Smith". The top-level Semantic-Gender KS, then, defines only *ks-function*, while its sub-KS's for English and Japanese specify their own *ks-data* and inherit the same *ks-function*. A *ks-language* slot specifies languages if a particular KS is applicable for specific languages.

Most of the KS's are language-independent (e.g. all the generators and the semantic type filters), and even when they are language-specific, the function

(defframe discourse-world (discourse-data-structure)	; DW
date	; date of the text
location	; location where the text is originated
topics	; semantic concepts which correspond to global topics of the text
position	; the corresponding character position in the text
discourse-clauses	; a list of discourse clauses in the current DW
sub-discourse-worlds)	; a list of DWs subordinate to the current one
(defframe discourse-clause (discourse-data-structure)	; DC
discourse-markers	; a list of discourse markers in the current DC
syntax	; an f-structure for the current DC
parse-tree	; a parse tree of this S
semantics	; a semantic (KB) object representing the current DC
position	; the corresponding character position in the text
date	; date of the current DC
location	; location of the current DC
subordinate-discourse-clause	; a DC subordinate to the current DC
coordinate-discourse-clauses)	; coordinate DC's which a conjoined sentence consists of
(defframe discourse-marker (discourse-data-structure)	; DM
position	; the corresponding character position in the text
discourse-clause	; a pointer back to DC
syntax	; an f-structure for the current DM
semantics	; a semantic (KB) object
file card)	; a pointer to the file card
(defframe file-card (discourse-data-structure)	; FC
co-referring-discourse-markers	; a list of co-referring DM's
updated-semantic-info)	; a semantic (KB) object which contains cumulative semantics

Figure 2: Discourse World, Discourse Clause, Discourse Marker, and File Card

definitions are shared. In this way, much of the discourse knowledge source KB is sharable across different languages.

### 2.2.2 Discourse Phenomenon KB

The discourse phenomenon KB contains hierarchically organized discourse phenomenon objects as shown in Figure 4. Each discourse phenomenon object has four slots (dp-definition, dp-main-strategy, dp-backup-strategy, and dp-language) whose values can be inherited. The *dp-definition* of a discourse phenomenon object specifies a definition of the discourse phenomenon so that an anaphoric discourse marker can be classified as one of the discourse phenomena. The *dp-main-strategy* slot specifies, for each phenomenon, a set of KS's to apply to resolve this particular discourse phenomenon. The *dp-backup-strategy* slot, on the other hand, provides a set of backup strategies to use in case the main strategy fails to propose any antecedent hypothesis. The *dp-language* slot specifies languages when the discourse phenomenon is only applicable to certain languages (e.g. Japanese “dou” ellipsis).

When different languages use different sets of KS's for main strategies or backup strategies for the same discourse phenomenon, language specific dp-main-strategy or dp-backup-strategy values are specified. For example, when an anaphor is a 3rd person pronoun in a partitive construction (i.e. 3PRO-Partitive-Parent)<sup>2</sup>, Japanese uses a different generator for the main strategy (Current-and-Previous-DC) than English and Spanish (Current-and-Previous-Sentence).

<sup>2</sup>e.g. “three of them” in English, “tres de ellos” in Spanish, “uchi san-nin” in Japanese

Because the discourse KS's are independent of discourse phenomena, the same discourse KS can be shared by different discourse phenomena. For example, the Semantic-Superclass filter is used by both Definite-NP and Pronoun, and the Recency orderer is used by most discourse phenomena.

### 2.2.3 Discourse Domain KB

The discourse domain KB contains discourse domain objects each of which defines a set of discourse phenomena to handle in a particular domain. Since texts in different domains exhibit different sets of discourse phenomena, and since different applications even within the same domain may not have to handle the same set of discourse phenomena, the discourse domain KB is a way to customize and constrain the workload of the discourse module.

## 2.3 Resolution Engine

The Resolution Engine is the run-time processing module which finds the best antecedent hypothesis for a given anaphor by using data in both the global discourse world and the discourse KB's. The Resolution Engine's basic operations are shown in Figure 5.

### 2.3.1 Finding Antecedents

The Resolution Engine uses the discourse phenomenon KB to classify an anaphor as one of the discourse phenomena (using dp-definition values) and to determine a set of KS's to apply to the anaphor (using dp-main-strategy values). The Engine then applies the generator KS to get an initial set of hypotheses and removes those that do not pass the filter

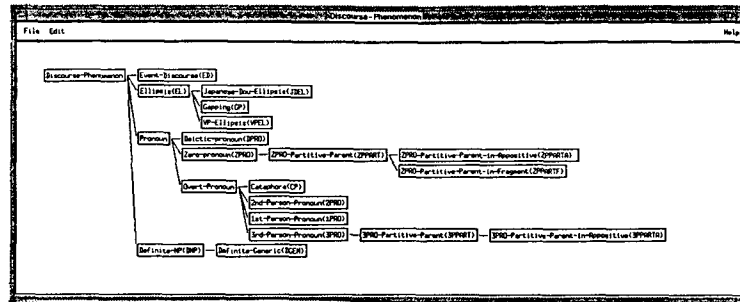


Figure 4: Discourse Phenomenon KB

For each anaphoric discourse marker in the current sentence:

**Find-Antecedent**

Input: anaphor to resolve, global discourse world

**Get-KSs-for-Discourse-Phenomenon**

Input: anaphor to resolve, discourse phenomenon KB

Output: a set of discourse KS's

**Apply-KSs**

Input: anaphor to resolve, global discourse world, discourse KS's

Output: the best hypothesis

Output: the best hypothesis

**Update-Discourse-World**

Input: anaphor, best hypothesis, global discourse world

Output: updated global discourse world

Figure 5: Resolution Engine Operations

KS's. If only one hypothesis remains, it is returned as the anaphor's referent, but there may be more than one hypothesis or none at all.

When there is more than one hypothesis, orderer KS's are invoked. However, when more than one orderer KS could apply to the anaphor, we face the problem of how to combine the preference values returned by these multiple orderers. Some anaphora resolution systems (cf. Carbonell and Brown [6], Rich and LuperFoy [16], Rimon *et al.* [17]) assign scores to antecedent hypotheses, and the hypotheses are ranked according to their scores. Deciding the scores output by the orderers as well as the way the scores are combined requires more research with larger data. In our current system, therefore, when there are multiple hypotheses left, the most "promising" orderer is chosen for each discourse phenomenon. In Section 3, we discuss how we choose such an orderer for each discourse phenomenon by using statistical preference. In the future, we will experiment with ways for each orderer to assign "meaningful" scores to hypotheses.

When there is no hypothesis left after the main strategy for a discourse phenomenon is performed, a series of *backup strategies* specified in the discourse phenomenon KB are invoked. Like the main strat-

egy, a backup strategy specifies which generators, filters, and orderers to use. For example, a backup strategy may choose a new generator which generates more hypotheses, or it may turn off some of the filters used by the main strategy to accept previously rejected hypotheses. How to choose a new generator or how to use only a subset of filters can be determined by training the discourse module on a corpus tagged with discourse relations, which is discussed in Section 3.

Thus, for example, in order to resolve a 3rd person pronoun in a partitive in an appositive (e.g. anaphor ID=1023 in Figure 7), the phenomenon KB specifies the following main strategy for Japanese: generator = Head-NP, filters = {Semantic-Amount, Semantic-Class, Semantic-Superclass}, orderer = Recency. This particular generator is chosen because in almost every example in 50 Japanese texts, this type of anaphora has its antecedent in its head NP. No syntactic filters are used because the anaphor has no useful syntactic information. As a backup strategy, a new generator, Adjacent-NP, is chosen in case the parse fails to create an appositive relation between the antecedent NP ID=1022 and the anaphor.

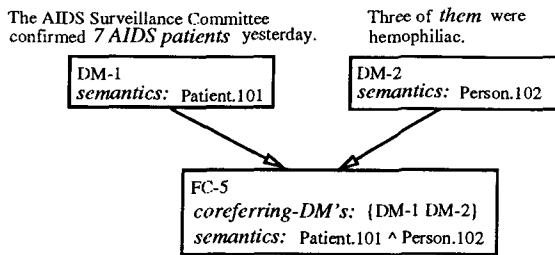


Figure 6: Updating Discourse World

### 2.3.2 Updating the Global Discourse World

After each anaphor resolution, the global discourse world is updated as it would be in File Change Semantics (cf. Heim [11]), and as shown in Figure 6. First, the discourse marker for the anaphor is incorporated into the file card to which its antecedent discourse marker points so that the co-referring discourse markers point to the same file card. Then, the semantics information of the file card is updated so that it reflects the union of the information from all the co-referring discourse markers. In this way, a file card accumulates more information as the discourse processing proceeds.

The motivation for having both discourse markers and file cards is to make the discourse processing a monotonic operation. Thus, the discourse processing does not replace an anaphoric discourse marker with its antecedent discourse marker, but only creates or updates file cards. This is both theoretically and computationally advantageous because the discourse processing can be redone by just retracting the file cards and reusing the same discourse markers.

## 2.4 Advantages of Our Approach

Now that we have described the discourse module in detail, we summarize its unique advantages. First, it is the only working *language-independent* discourse system we are aware of. By “language-independent,” we mean that the discourse module can be used for different languages if discourse knowledge is added for a new language.

Second, since the anaphora resolution algorithm is not hard-coded in the Resolution Engine, but is kept in the discourse KB’s, the discourse module is *extensible* to a new discourse phenomenon by choosing existing discourse KS’s or adding new discourse KS’s which the new phenomenon requires.

Making the discourse module *robust* is another important goal especially when dealing with real-world input, since by the time the input is processed and

passed to the discourse module, the syntactic or semantic information of the input is often not as accurate as one would hope. The discourse module must be able to deal with partial information to make a decision. By dividing such decision-making into multiple discourse KS’s and by letting just the applicable KS’s fire, our discourse module handles partial information robustly.

Robustness of the discourse module is also manifested when the imperfect discourse KB’s or an inaccurate input cause initial anaphor resolution to fail. When the main strategy fails, a set of backup strategies specified in the discourse phenomenon KB provides alternative ways to get the best antecedent hypothesis. Thus, the system tolerates its own insufficiency in the discourse KB’s as well as degraded input in a robust fashion.

## 3 Evaluating and Training the Discourse Module

In order to choose the most effective KS’s for a particular phenomenon, as well as to debug and track progress of the discourse module, we must be able to evaluate the performance of discourse processing. To perform objective evaluation, we compare the results of running our discourse module over a corpus with a set of manually created discourse tags. Examples of discourse-tagged text are shown in Figure 7. The metrics we use for evaluation are detailed in Figure 8.

### 3.1 Evaluating the Discourse Module

We evaluate overall performance by calculating *recall* and *precision* of anaphora resolution results. The higher these measures are, the better the discourse module is working. In addition, we evaluate the discourse performance over new texts, using blackbox evaluation (e.g. scoring the results of a data extraction task.)

To calculate a generator’s *failure rate*, a filter’s *false positive rate*, and an orderer’s *effectiveness*, the algorithms in Figure 9 are used.<sup>3</sup>

### 3.2 Choosing Main Strategies

The uniqueness of our approach to discourse analysis is also shown by the fact that our discourse module can be trained for a particular domain, similar to the ways grammars have been trained (cf. Black

<sup>3</sup>“The remaining antecedent hypotheses” are the hypotheses left after all the filters are applied for an anaphor.

<i>Overall Performance:</i> Recall = $N_c/I$ , Precision = $N_c/N_h$	
$I$	Number of anaphors in input
$N_c$	Number of correct resolutions
$N_h$	Number of resolutions attempted
<i>Filter:</i> Recall = $OP_c/IP_c$ , Precision = $OP_c/OP$	
$IP_c$	Number of correct pairs in input
$IP$	Number of pairs in input
$OP$	Number of pairs output and passed by filter
$OP_c$	Number of correct pairs output by filter
$1 - OP/IP$	Fraction of input pairs filtered out
$1 - OP_c/IP_c$	Fraction of correct answers filtered out (false positive rate)
<i>Generator:</i> Recall = $N_c/I$ , Precision = $N_c/N_h$	
$I$	Number of anaphors in input
$N_h$	Number of hypotheses in input
$N_c$	Number of times correct answer in output
$N_h/I$	Average number of hypotheses
$1 - N_c/I$	Fraction of correct answers not returned (failure rate)
<i>Orderer:</i>	
$I$	Number of anaphors in input
$N_c$	Number of correct answers output first
$N_c/I$	Success rate (effectiveness)

Figure 8: Metrics used for Evaluating and Training Discourse

- For each discourse phenomenon,
  - given anaphor and antecedent pairs in the corpus,
  - calculate how often the *generator* fails to generate the antecedents.
- For each discourse phenomenon,
  - given anaphor and antecedent pairs in the corpus,
  - for each filter,
  - calculate how often the *filter* incorrectly eliminates the antecedents.
- For each anaphor exhibiting a given discourse phenomenon in the corpus,
  - given the remaining antecedent hypotheses for the anaphor,
  - for each applicable orderer,
  - test if the *orderer* chooses the correct antecedent as the best hypothesis.

Figure 9: Algorithms for Evaluating Discourse Knowledge Sources

厚生省のエイズ（後天性免疫不全症候群）サーベイランス委員会（委員長・堀川優一順天堂大名誉教授）は、六日、新たに  
 <DM ID=1000>エイズ患者七人</DM>（<DM ID=1001 Type=3PARTA Ref=1000>うち</DM>死亡三人）と感染者十七人を確認した。  
 [The AIDS Surveillance Committee of the Health and Welfare Ministry (Chairman, Professor Emeritus Junichi Shiokawa), on the 6th, newly confirmed 7 AIDS patients (of them 3 are dead) and 17 infected people.]

<DM ID=1020 Type=DNP Ref=1000>新たに見つかった患者七人</DM>  
 のうち<DM ID=1021>四人</DM>が<DM ID=1022 Type=BE Ref=1021>  
 男性同性愛者</DM>（<DM ID=1023 Type=3PARTA Ref=1021>うち  
 </DM>死亡二人）、<DM ID=1024 Type=ZPARTF Ref=1020></DM>一人  
 が異性間接触の女性、<DM ID=1025 Type=ZPARTF Ref=1020></DM>  
 <DM ID=1026>二人</DM>（<DM ID=1027 Type=JDEL Ref=1026>同  
 </DM>一人）が汚染血液製剤によるもの。

[4 of the 7 newly discovered patients were male homosexuals</DM>  
 (of them</DM> 2 are dead), 1 is heterosexual woman, and 2 (ditto 1)  
 are by contaminated blood product.]

La Comisi'o'n de Te'cnicos del SIDA informo' ayer de que existen <DM ID=2000>196 enfermos de <DM ID=2001>SIDA</DM></DM> en la Comunidad Valenciana. De <DM ID=2002 Type=PRO Ref=2000>ellos </DM>, 147 corresponden a Valencia; 34, a Alicante; y 15, a Castello'n. Mayoritariamente <DM ID=2003 Type=DNP Ref=2001>la enfermedad</DM> afecta a <DM ID=2004 Type=GEN>los hombres</DM>, con 158 casos. Entre <DM ID=2005 Type=DNP Ref=2000>los afectados </DM> se encuentran nueve nin'os menores de 13 an'os.

Figure 7: Discourse Tagged Corpora

[4]). As Walker [18] reports, different discourse algorithms (i.e. Brennan, Friedman and Pollard's centering approach [5] vs. Hobbs' algorithm [12]) perform differently on different types of data. This suggests that different sets of KS's are suitable for different domains.

In order to determine, for each discourse phenomenon, the most effective combination of generators, filters, and orderers, we evaluate overall performance of the discourse module (cf. Section 3.1) at different rate settings. We measure particular generators, filters, and orders for different phenomena to identify promising strategies. We try to minimize the failure rate and the false positive rate while minimizing the average number of hypotheses that the generator suggests and maximizing the number of hypotheses that the filter eliminates. As for orderers, those with highest effectiveness measures are chosen for each phenomenon. The discourse module is "trained" until a set of rate settings at which the overall performance of the discourse module becomes highest is obtained.

Our approach is more general than Dagan and Itai [7], which reports on training their anaphora resolution component so that "it" can be resolved to its correct antecedent using statistical data on lexical relations derived from large corpora. We will certainly incorporate such statistical data into our discourse KS's.

### 3.3 Determining Backup Strategies

If the main strategy for resolving a particular anaphor fails, a backup strategy that includes either a new set of filters or a new generator is attempted. Since backup strategies are employed only when the main strategy does not return a hypothesis, a backup strategy will either contain fewer filters than the main strategy or it will employ a generator that returns more hypotheses.

If the generator has a non-zero failure rate <sup>4</sup>, a new generator with more generating capability is chosen from the generator tree in the knowledge source KB as a backup strategy. Filters that occur in the main strategy but have false positive rates above a certain threshold are not included in the backup strategy.

## 4 Related Work

Our discourse module is similar to Carbonell and Brown [6] and Rich and LuperFoy's [16] work in using multiple KS's rather than a monolithic approach (cf. Grosz, Joshi and Weinstein [9], Grosz and Sidner [8], Hobbs [12], Ingria and Stallard [13]) for anaphora resolution. However, the main difference is that our system can deal with multiple languages as well as multiple discourse phenomena<sup>5</sup> because of our fine-grained and hierarchically organized KS's. Also, our system can be evaluated and tuned at a low level because each KS is independent of discourse phenomena and can be turned off and on for automatic evaluation. This feature is very important because we use our system to process real-world data in different domains for tasks involving text understanding.

## References

- [1] Chinatsu Aone, Hatte Blejer, Sharon Flank, Douglas McKee, and Sandy Shinn. The Murasaki Project: Multilingual Natural Language Understanding. In *Proceedings of the ARPA Human Language Technology Workshop*, 1993.
- [2] Chinatsu Aone, Doug McKee, Sandy Shinn, and Hatte Blejer. SRA: Description of the SOLOMON System as Used for MUC-4. In *Proceedings of Fourth Message Understanding Conference (MUC-4)*, 1992.

<sup>4</sup>Zero failure rate means that the hypotheses generated by a generator always contained the correct antecedent.

<sup>5</sup>Carbonell and Brown's system handles only intersentential 3rd person pronouns and some definite NPs, and Rich and LuperFoy's system handles only 3rd person pronouns.

- [3] Damaris Ayuso. Discourse Entities in JANUS. In *Proceedings of 27th Annual Meeting of the ACL*, 1989.
- [4] Ezra Black, John Lafferty, and Salim Roukos. Development and Evaluation of a Broad-Coverage Probabilistic Grammar of English-Language Computer Manuals. In *Proceedings of 30th Annual Meeting of the ACL*, 1992.
- [5] Susan Brennan, Marilyn Friedman, and Carl Pollard. A Centering Approach to Pronouns. In *Proceedings of 25th Annual Meeting of the ACL*, 1987.
- [6] Jaime G. Carbonell and Ralf D. Brown. Anaphora Resolution: A Multi-Strategy Approach. In *Proceedings of the 12th International Conference on Computational Linguistics*, 1988.
- [7] Ido Dagan and Alon Itai. Automatic Acquisition of Constraints for the Resolution of Anaphora References and Syntactic Ambiguities. In *Proceedings of the 13th International Conference on Computational Linguistics*, 1990.
- [8] Barbara Grosz and Candace L. Sidner. Attentions, Intentions and the Structure of Discourse. *Computational Linguistics*, 12, 1986.
- [9] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Providing a Unified Account of Definite Noun Phrases in Discourse. In *Proceedings of 21st Annual Meeting of the ACL*, 1983.
- [10] Raymonde Guindon, Paul Stadky, Hans Brunner, and Joyce Conner. The Structure of User-Adviser Dialogues: Is there Method in their Madness? In *Proceedings of 24th Annual Meeting of the ACL*, 1986.
- [11] Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts, 1982.
- [12] Jerry R. Hobbs. Pronoun Resolution. Technical Report 76-1, Department of Computer Science, City College, City University of New York, 1976.
- [13] Robert Ingria and David Stallard. A Computational Mechanism for Pronominal Reference. In *Proceedings of 27th Annual Meeting of the ACL*, 1989.
- [14] Hans Kamp. A Theory of Truth and Semantic Representation. In J. Groenendijk et al., editors, *Formal Methods in the Study of Language*. Mathematical Centre, Amsterdam, 1981.
- [15] Lauri Karttunen. Discourse Referents. In J. McCawley, editor, *Syntax and Semantics 7*. Academic Press, New York, 1976.
- [16] Elaine Rich and Susan LuperFoy. An Architecture for Anaphora Resolution. In *Proceedings of the Second Conference on Applied Natural Language Processing*, 1988.
- [17] Mori Rimón, Michael C. McCord, Ulrike Schwall, and Pilar Martínez. Advances in Machine Translation Research in IBM. In *Proceedings of Machine Translation Summit III*, 1991.
- [18] Marilyn A. Walker. Evaluating Discourse Processing Algorithms. In *Proceedings of 27th Annual Meeting of the ACL*, 1989.
- [19] Bonnie Webber. A Formal Approach to Discourse Anaphora. Technical report, Bolt, Beranek, and Newman, 1978.