Identifying Unknown Proper Names in Newswire Text

Inderjeet Mani, T. Richard Macmillan,
Susann Luperfoy, Elaine P. Lusher,
Sharon J. Laskowski

Artificial Intelligence Technical Center
The MITRE Corporation, Mail Stop Z401
7525 Colshire Drive, McLean, Virginia 22102-3481
mani@starbase.mitre.org

Abstract

The identification of unknown proper names in text is a significant challenge for NLP systems operating on unrestricted text. A system which indexes documents according to name references can be useful for information retrieval or as a pre-processor for more knowledge intensive tasks such as database extraction. This paper describes a system which uses text skimming techniques for deriving proper names and their semantic attributes automatically from newswire text, without relying on any listing of name elements. In order to identify new names, the system treats proper names as (potentially) context-dependent linguistic expressions. In addition to using information in the local context, the system exploits a computational model of discourse which identifies individuals based on the way they are described in the text, instead of relying on their description in a pre-existing knowledge base.

1 Introduction

The identification of unknown proper names in text is a significant challenge for NLP systems operating on unrestricted text. A system which indexes documents according to name references can be useful for information retrieval or as a pre-processor for more knowledge intensive tasks such as database extraction. With the growing use of tagged corpora in a variety of language-related research areas, being able to reliably tag proper names is an obvious advantage. In addition, the development of practical techniques for name identification help to shed light on the various uses of proper names in text.

Traditional approaches to unknown proper name identification involve, broadly speaking, the lexical lookup of names or name fragments in a name database. For example, approaches such as [Aone et al., 92], [Aberdeen et al., 92], and [Cowie et al., 92], identify person names by marking off phrases which contain unknown words close to known name elements like first or last names, and (in [Cowie et al., 92]) unknown words close to specific title-words. As the above studies show, name databases such as cross-cultural listings of common first and last names as well as existing geographical gazetteers, are helpful in name recognition. However, approaches based exclusively on unknown words and known name elements can be confused by known common nouns (or other parts of speech) which occur in proper names, even person names. More importantly, such approaches require an initial name element database. Creating such databases can be a labor-intensive task. Furthermore, no matter how large the database one can manually construct, the problem still arises of identifying names which don't happen to be present

in any given name database. The fact that proper names form, lexically speaking, an open class whose elements grow far more rapidly than other open classes, and the fact that they often contain other open-class elements, makes the incompleteness of such databases an obvious problem.

Our approach aims at deriving proper names and their semantic attributes automatically from large corpora, without relying on any listing of name elements. The overall approach is based on two main ideas. Firstly, we hypothesize that for certain genres of text (for example, Wall Street Journal news stories), new references are introduced by information occurring in the immediate syntactic environment of the proper name. (What the precise set of such genres is remains to be determined, but our initial set includes the most common forms of news stories and excludes literary narratives.) Many of these local contextual clues reflect felicity conventions for introducing new names. New names of people (as well as organization names, and to some extent location names) are generally accompanied by honorifics and various appositive phrases which help anchor the new name reference to mutually assumed knowledge. Further contextual clues come from selectional restrictions, for example, given "Kambomambo murdered Zombaluma" (from [Radford, 88]), the verb is the main clue to the hypothesis that the two names are those of people.

Although the idea of exploiting local context to identify semantic attributes in new names is in itself not new (e.g. [Coates-Stephens, 91], [Paik et al., 93]), little attention has been paid in name identification work to the discourse properties of names. Our second, and more general idea is to view proper names as linguistic expressions whose interpretation often depends on the discourse context. For example, in the discourse "U.S. President Bill Clinton....Clinton....Mr. Clinton....President Clinton", the interpretations of "Clinton", "Mr. Clinton" and "President Clinton" are dependent on the prior reference to "U.S. President Bill Clinton", much as "the president", "he" and "himself" are dependent on prior context in the discourse "U.S. President Bill Clinton$_i$....the president$_i$....he$_i$....himself$_i$". The need for text-driven extraction of names presupposes in turn a computational model of discourse which identifies individuals based on the way they are described in the text, instead of relying on their description in a pre-existing knowledge base. The overall discourse representation framework which we use is Luperfoy's three-tiered model [Luperfoy, 91], which in turn is a computational adaptation of Landman's pegs model of NP semantics [Landman 86].

The idea of the three-tiered model is that there are three significant levels of representation: linguistic expressions, Discourse Pegs, and knowledge base objects. A distinctive feature of Discourse Pegs (hereafter referred to as Pegs) as opposed to similar constructs in the literature, like File Cards ([Heim, 81]), Database Objects ([Sidner, 79]), Discourse Referents ([Karttunen, 68]), and Discourse Entities ([Webber, 78], [Dahl and Ball, 90]), is that they describe unique objects with respect to the current discourse, rather than with respect to the underlying belief system or world model. Thus, in an article mentioning Bill Clinton there may be two guises in which he may appear, as Governor Clinton and President Clinton; these would correspond to two distinct pegs. It is important to stress that pegs, as a result, do not correspond to equivalence classes of coreferential mentions; rather, there is one peg for each distinct object under discussion, irrespective of the number of entities in the world of reference. Objects which are distinct in the text may still need to be related to each other for their interpretation; for example, in the discourse "President Bill Clinton... the Clintons....Hilary", the expressions "President Bill Clinton", "the Clintons" and "Hilary" each introduce new pegs, but these pegs are each linked, as

"partial dependents", to the previous one. An interesting subcase of this involves name mergers, e.g. an article describing a joint venture between two companies may use the two individual company names followed by a merged name for the joint venture.

In applying this framework to the unknown name problem, we first distinguish three types of entities: (i) Mentions - these are text segments which are tokens of proper names in text; (ii) Contexts - these are text segments which provide information about syntactic and semantic properties associated with a name; and (iii) Hypotheses - these are hypotheses about individuals and their semantic attributes, associated with a Mention. Given this framework, the goal of unknown name identification is to use the text itself to generate Hypotheses about possible individuals distinguished by a Mention. In a given text context, descriptions from earlier Mentions of a name may be further specified by new information associated with subsequent Mentions of the name (which may take a somewhat different form from previous Mentions). In general, two Hypotheses, each associated with a different Mention, are linked together (by means of a common Peg) whenever they are mutually compatible. Thus, two Mentions, Mention 1 and Mention 2, can be considered to be indirectly anchored together to a common Peg whenever hypothetical information associated with each is mutually compatible. For ease of presentation, we may speak of these coanchored mentions as "coreferential" (when what we really mean is this more specific sense of coanchoring); also, we will use the capitalized word "Coreference" for the process of computing pegs for a mention, a process which may result in either the coanchoring of the mention to one or more existing pegs, or the allocation of a new peg. We describe the Coreference process in more detail in Section 4.

2 Proper Names - Syntactic Forms and Semantic Attributes

We first need to describe more precisely what we mean by proper names. In terms of syntactic categories, proper names are commonly identified as lexical NPs. In the examples in this paper, we use [] to identify an internal proper name constituent of interest. Proper names often occur inside definite NPs, where the proper name can function as the syntactic head ("the [President of France]", "the [Gulf of California]", "the Reagan [White House]", "Iraq's president [Saddam Hussein]", "Lake [George]"), a complement ("the president of [France]"), or an adjunct or attributive NP ("the [Reagan] White House", "the [Bush] administration"). They can also occur with indefinite determiners ("an [Arnold Schwartznegger]", "a [Washington Redskin]", "an [IBM]"). As lexical NPs, proper names have substantial internal structure: they can be formed out of primitive proper name elements ("Oliver North", "Gramm-Rudman" "Villa-Lobos"), other proper names ("Lake George", "the [President of France]", "the [Reagan White House]", "Anne of a Thousand Days") and also out of non-proper names ("the [Savings and Loan] crisis", "General Electric Co.", "Federal Savings and Loan Insurance Corporation", "Committee for the Protection of Public Welfare"). A common resulting form is the open compound proper name ("the [Carter Administration National Energy Conservation Committee]").

Given an occurrence of a proper name in text, we can use the text itself to extract semantic attributes associated with that name. As mentioned earlier, the local context frequently offers valuable clues. Also, for certain varieties of names, such as organization names ("Microelectronics and Computer Technology Corporation") and geographical location names ("Easter Island"), the internal structure of the name can be used to hy-

pothesize various semantic attributes. A study reported in [Amsler, 87] on proper names in the New York Times containing the word "center" (such as "Grand Forks Energy Research Center" and "Boston University's Center for Adaptive Systems") is suggestive of the scope of such techniques. Identifying idiomatic uses is obviously a problem: as [Amsler, 87] points out, "Grand Funk Railroad" is the name of a rock group. In keeping with such an approach, we have developed subgrammars which model the internal syntax and semantics of geographical names, which, in combination with information from the local Context, can be used to guess the type of location.

3 Overall Algorithm

The approach of text skimming is associated with much recent work on data extraction from text (e.g. [Mauldin 89], [Jacobs 88], and many others). In general, this means that different parts of the text can be processed to different depths, with some parts being skipped over lightly. The text skimming approach also implies, in our case, that we lighten the burden of lexical semantics: in contrast to approaches like [Coates-Stephens, 91], we need only represent word meanings for words closely related in meaning to the semantic attributes we are attempting to extract. While we were attracted to such an approach, our work also explores some of the practical tradeoffs associated with text skimming.

The overall algorithm involves first tokenizing the text into sentences and words, then proposing candidate name mentions, and finally allowing various knowledge sources (KSs) to vote on and propose hypotheses about a given mention. Each KS can generate multiple scored hypotheses about a given mention. The KSs are applied in a pre-determined order to a mention, with each KS refining the hypotheses generated by the previous KS. Names which are identified beyond a certain confidence level (a variable recall/precision threshold) are added to a hypothetical lexicon after asking the user about them. Over time, learnt names (or name elements) in the hypothetical lexicon increase the likelihood of recognizing a name mention.

The system assumes a shallow knowledge base representing the specific concepts and attributes to be extracted. For example, a president is either a head-of-state or a corporate-officer, and a person has age, title, gender and occupation; a place may be a continent, country, state, city, etc. The semantic lexicon associated with this knowledge base is a small one, of the order of a few hundred words, consisting of titles, honorifics, location nouns and organizational suffixes extracted from phrases tagged as NP in the Penn Treebank Wall Street Journal (WSJ) corpus. Words associated with these entities are the only ones which currently have any lexical semantics in our system. (A noteable exception comes from our work on place names, which exploits, for comparison purposes, a TIPSTER gazetteer). This small lexicon is complemented by the very large syntactic lexicon derived from the Lancaster-Oslo-Bergen corpus, which is used by our part-of-speech tagger and parser [de Marcken, 90].

A variety of different grammars are used by the system. The simpler kind are regular expression grammars which rely on part-of-speech, some specific key lexical items from our semantic lexicon, and punctuation - these grammars drive a pattern matcher which is an extension of the one described in [Norvig, 92]. Such grammars are used for modeling the internal syntax and semantics of geographical names and person names, and also for locating various Context boundaries - for example, identifying an appositive construction. Further segmentation of the appositive (see Section 3.3) is done by a mix-

ture of pattern-matching of the above kind and NP parsing (into head, pre-modifiers, and post-modifiers) using the MIT Fast Parser [de Marcken, 90] and its associated syntactic grammar. At present, we perform only a rudimentary analysis of organization names, merely hypothesizing whether a mention is a likely organization name or not.

We have used the WSJ as a training corpus. The mode of knowledge engineering has involved building a rudimentary proper name tagger, followed by iterations through a cycle of tagging the corpus with records of Mentions and their occurrence Contexts, examining the tagged corpus to improve the knowledge sources, and retagging. It is envisaged that over time, certain hypothesized individuals will be incorporated into the knowledge base.

3.1 The Mention Generator

Given text which distinguished between upper-case and lower-case, the KS which proposes candidate mentions is based on finding contiguous capitalized words including lower-case function words (e.g. "of", "and", "de", etc.). Only those sentences containing such mentions are processed (partially) by other KSs. This capitalization heuristic recalls all the proper names, but it is slightly imprecise, especially since sentence-initial words are always capitalized in case distinguished text. To eliminate these, a part-of-speech based filter is applied to each sentence-initial candidate sequence, discarding the initial word unless it is from a designated set (a noun, and adjective, a NP, the definite determiner "the", or an unknown word) and excluding isolated definite determiners. In practice, this filter works extremely well. However, mentions may need to be split up later when more knowledge is available, since titles may need to be extracted, and function words like conjunctions and prepositions introduce attachment ambiguities (e.g. "Democratic Sens. Dennnis De Concini and Alan Cranston", "Food and Drug Administration").

Given newswire text which makes no reliable case distinction (e.g. all-uppercase or all-lowercase text), the proposer proposes contiguous sequences of words with categories in the above designated set. The proposals include all the mentions proposed in case-sensitive mode, but the use of shallow processing here is obviously far less precise, generating 3 to 4 times as many mentions. However, incorrect candidates get filtered out eventually, since there are no significant hypotheses about them.

3.2 Knowledge Sources

Each KS can have multiple hypotheses with different confidences. For example, the mention "General Electric Co.", may result in an initial hypothesis that it could be a person, based on interpreting "General" as a title, and other hypotheses that it could be a company or a county, based on the abbreviated suffix "Co.". Each distinct filling of attributes corresponds to a distinct hypothesis. We currently use a somewhat crude thresholding scheme: viewing an attribute-KS as filling a single attribute, the confidence of a particular attribute-KS's hypothesis is a weighted sum of the match strength and the attribute-KS's strength, the latter being based on an initial global ranking followed by later calibration. The KSes are based on simple heuristics, which, except for Coreference, are interesting more in terms of their combined effect than in themselves. For example, Organization? is a KS which trivially determines organizationhood by the presence of certain company suffixes like "Inc.". Honorifics uses the text occurrence of honorifics ("Mr.", "His Holiness", "Lt. Col.") from the small semantic lexicon to make inferences about personhood, as well as gender and job occupation.

The Job-Title and Age KSes extract their data from appositive constructions and premodifying adjective phrases and noun compounds. A job-title (a surface string like "president-for-life") may or may not be in the syntactic or semantic lexicon; if it is present in the semantic lexicon, an effort is made to infer, based on context, the person's job-occupation, as discussed in the next section. Person-Name is a weak KS which segments potential person-names without being able to determine personhood with any confidence. Name-Element upgrades the confidence of names which match learned name elements. Agent-of-Human-Action looks for verbs like "lead", "head", "say", "explain", "think", "admit" in the syntactic context to estimate whether a given mention could be a person, though the assignment of agent role to the mention is only approximate; the frequent use of metonymy involving companies as agents makes this a relatively weak KS. A Short-Name? KS reflects a newspaper honorific convention of not using single-word titleless names in introductory people mentions (as in "Yesterday [Kennedy] said.."). The Location KS uses patterns involving locational category nouns from the semantic lexicon like "town", "sea", "gulf", "north" to flag location mentions like "town of Beit Sahoud".

3.3 Appositives

Appositives are important linguistic devices for introducing new mentions. We limit ourselves to constituents of the form <NP, NP>. These are of the form name-comma-appositive (e.g. "<name>, <ORG>'s top managing director", "<name>, a small Bay Area town"), and appositive-comma-name (e.g. "a top Japanese executive, <name>"). We ignore double appositives, except for simple ones involving age, as in "Osamu Nagayama, 33, senior vice president and chief financial officer of Chugai.". Therefore, given a candidate name mention, the appositive modifier is a NP to the right or the left of the name. (A <NP, NP> constituent can of course be part of an enumerated, conjoined NP; however, if one conjunct is a name, it's likely that the other one may be too. Of course, a <NP, NP> sequence may not be a constituent in the first place).

To identify appositive boundaries, we experimented with both (a) a regular expression grammar tuned to find appositives in the training corpus, and (b) syntactic-grammar based parsing using the MIT Fast Parser. Here we found pattern matching, based on looking for left and right delimiters such as comma and certain parts of speech, to be far more accurate. For example, given "said Chugai's senior vice president for international trade, Osamu Nagayama", the appositive identifier would find "Chugai's senior vice president for international trade". For extracting premodifiers, head and postmodifiers, we have found technique (b) to be somewhat more useful, though attachment errors still occur. The extracted premodifiers and head (or maximal fragment thereof) are then looked up in the semantic lexicon ontology; looking up "senior vice president" would yield corporate-officer or government-official. Hypotheses about "Chugai", based on information from Coreference linking it to an earlier mention of "Chugai Pharmaceutical Corp.", can be used to infer that "Osamu Nagayama" is more likely to be a corporate officer than a government official.

4 Coreference

4.1 Normalized Names

When a new mention is processed by the Coreference KS, pegs from previous mentions seen earlier in the document are considered as candidate coanchored mentions. Obviously, we wish to avoid considering the set of all previous pegs in the discourse. The use of focus information at some level can be used to constrain this set, but that would require in turn strong assumptions about the discourse structure of texts - which could severely limit our applicable domains. Still, it seems unreasonable, given a mention of "Bill Clinton", to consider a peg for "New York City" as a possible antecedent. This suggests we consider only previous mentions which are similar in some way. We do this by indexing each mention by a normalized name, and considering only pegs for mentions which have the same normalized name. This raises the issue of the choice of a normalized name key.

Obviously, there can be considerable variability in the form of a name across different mentions. For example, a mention of "President Clinton" could be followed by "Bill Clinton"; one of "Georgetown University" by "Georgetown"; "the Los Angeles Lakers" by "the Lakers". (See [Carroll, 85] for a discussion of the regularities and numerous irregularities in alternations in name forms, many of which involve metonymic reference). In the training corpus, the heuristic of choosing the last name element in the surface form of a name as a normalized name works well for people. This may reflect the fact that newspapers often impose their own normalization conventions. There are obvious exceptions to the last name element heuristic; for example, in the WSJ, a mention of "Roh Tae Woo" is followed by a co-referential mention of "Mr. Roh". For organization names, our heuristic is to choose all but the last element as the normalized name, but to allow a degree of partial matching. Given a new name mention, upon failure to find a partition cell having previous mentions with the same normalized name, partition cells with neighboring normalized names are searched. (The closeness metric here involves having a high percentage of sequential words in common). Thus the WSJ mentions of "Leaseway Transportation Corp" followed by "Leaseway" would be tied together, as would "Canadian Technical Tape Inc." and "Technical Tape". Of course, at the time of invoking Coreference for a hypothesis associated with a mention, we may or may not have (depending in part on the ordering of knowledge sources) enough information to decide which normalized name heuristic to invoke, in which case we use the last name as a default.

In practice the matching on normalized names works well, except for cases like Mr. Roh above, and in cases of spelling errors. If necessary, the system can use a strategy of iterative widening; if the system fails to find a coreferring mention, in iterative widening mode it attempts to search through the space of all other previous mentions. In this mode, the system can also separately collect and warn about mentions whose names are close to (using the Damerau-Levenshtein similarity metric) but not identical in spelling to the current mention.

4.2 Coreference Algorithm

At each peg site, the system unifies information from Hypotheses associated with the new mention with information accumulated from the other mentions at the peg site. As a rule, successful unification results in coanchoring. The Coreference procedure terminates when all the pegs in the relevant normalized name partition cell have been considered. A failure

of unification, which results from a conflict from a new mention at a peg site, can lead to three possible outcomes: (i) *Ignoring* of the conflict, in which case coanchoring of the new mention to the peg is established; (ii) *Overriding* of earlier information accumulating at the peg in question, in which case coanchoring of the new mention to the peg is established, and coanchoring links from any other conflicting mentions to the peg are broken; or (iii) *Honoring* of the conflict, leading to (a) considering some other peg, or if none remains, (b) the creation of a new peg. The decision whether to *Ignore* or *Override* is based on the relative strength of the hypotheses emanating from different mentions: (i) Conflicts are *Ignored* when the information from the new mention has low confidence. (ii) Conflicts are *Overriden* when (a) (Weak-Opposition-Loses) the conflicting information from the new mention has high confidence and the conflicting information from the old mention has low confidence, or (b) (Strong-Majority-Wins) all the other evidence at the peg (there must be some) strongly confirms the new mention's hypothesis. Strong-Majority-Wins requires that there are at least two old mentions at the peg, with only one old mention giving rise to the conflict, and with all the other old mentions at the peg being compatible with the new mention at a high level of confidence for each attribute. Once a link from a mention is broken, the mention can be relinked to some other peg (either existing, or a new one). (iii) Otherwise, the conflict is *Honored.*

Figure 1 shows an example of Coreference and ambiguity resolution. To simplify the presentation, only one hypothesis is shown per mention, appositives are ignored, and each attribute of each hypothesis is assumed to have the same confidence. (A Mention is identified as a string, with the hypothesis directly below it.)

Assume Mention 1 is discourse-initial; assume further that Person-Name and Age have fired. Coreference on Mention 1 leads to the creation of a new peg, Peg 1, representing the hypothetical entity Bill Clinton. Coreference on Mention 2 leads to a search in the normalized-name partition for Clinton. The system unifies the properties associated with Mention 2 with Mention 1's properties. In this case, since there is no conflict, both mentions are anchored to Peg 1. Mention 3 results in Coreference attempting a link to Peg 1. This leads to a conflict in unification with the properties from one of the other links to Mention 1, arising specifically from the full name and gender information extracted from Mention 1. These are conflicts because they violate a single-valued constraint for these attributes. The conflict with Mention 3 is honored, since there is no disparity in confidence measures. This results in Mention 3 being anchored to a new peg Peg 2, representing a hypothetical entity Hilary Clinton. Mention 4's properties are compatible with both pegs, hence it is coanchored to both, making it ambiguous. Mention 5 leads to a conflict on name at Peg 1. There is no confidence disparity at Peg 1, so the conflict is honored, resulting in a search for some other peg. At Peg 2, there is a conflict on occupation, but since Mention 3 is compatible with Mention 5, by Strong-Majority-Wins, Mention 3 overrides the information from Mention 4. This leads to breaking of the link of the conflicting mention with Peg 2, disambiguating Mention 4.

5 Conclusion

The system has been run on one million words of text (two years of WSJ training corpus as well as the [Kahaner, 91] email corpus). The identification of person names and geographical locations is in place, as well as a rudimentary organization tagger (which does not extract any interesting attributes regarding the organization). The pegs-based

Coreference KS has been implemented, but the breaking of a link from a mention to a peg is not as yet propagated to other pegs. We have not yet implemented a treatment of partial dependents, which involve modeling inter-relationships among pegs. Problems we are currently working on include conjunctions (e.g. is "AVX and Kyocera" a single entity?), the treatment of partial dependents and references to sets (e.g. the discourse "Indira Gandhi....Rajiv Gandhi....the Gandhis"). We are also investigating the applicability of Bayesian inference networks to the overall problem.

Recently, we conducted an empirical evaluation of the system. In a nutshell (details are deferred to a separate paper), the evaluation was carried out on a test set of 42 hand-tagged WSJ articles, using a scoring program we developed. The hand-tagging marked only the type of the tag (person, organization, or location), ignoring attributes. Scores on <precision, recall> varied from <76%, 72%> to <84%, 80%>, depending on whether partial matches (e.g. only a fragment of a name in the program's tag, or a title identified as part of a name) were accepted. We soon expect to more directly evaluate the Coreference KS, but in the meantime we can offer the observation that the Coreference KS has been observed to be extremely effective (apart from the exceptions we mentioned earlier) for name mentions in the WSJ, especially for people mentions.

In conclusion, then, we have found that a treatment of proper names as potentially context-dependent linguistic expressions can be effectively applied to the problem of unknown name identification in newswire text, especially when combined with local-context based text skimming. In addition to determining more precisely the genre limitations of such an approach, one future direction would be to consider porting the system to another language.

References

[Aberdeen et al., 92] J. Aberdeen, J. Burger, D. Connolly, S. Roberts, and M. Vilain, "Description of the Alembic System as used in MUC-4", *Proceedings of the Fourth Message Understanding Conference*, 1992, pp. 215-222.

[Amsler, 87] Robert A. Amsler, "Research Towards the Development of a Lexical Knowledge Base for Natural Language Processing", *SIGIR Forum*, 123, (1-2), 1989.

[Aone et al., 92] C. Aone, D. McKee, S. Shinn, H. Blejer, "Description of the Solomon System as Used for MUC-4", *Proceedings of the Fourth Message Understanding Conference*, 1992, pp. 259-267.

[Carroll, 85] John M. Carroll, "What's in a Name?", Freeman and Company, New York, 1985.

[Coates-Stephens, 91] Sam Coates-Stephens, "Automatic Lexical Acquisition Using Within-Text Descriptions of Proper Nouns", *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*, 1991, pp. 154-169.

[Cowie et al., 92] J. Cowie, L. Guthrie, Y. Wilks, J. Pustejovsky, and S. Waterman, "Description of the Solomon System as Used for MUC-4", *Proceedings of the Fourth Message Understanding Conference*, 1992, pp. 223-232.

[Dahl and Ball, 90] D. Dahl and C.N. Ball, "Reference Resolution in PUNDIT", Technical Report, Unisys, 1987.

[de Marcken, 90] C. G. de Marcken, "Parsing the LOB Corpus", *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 1990, pp. 243-251.

[Heim, 81] I. Heim, The Semantics of Definite and Indefinite Noun Phrases, Ph.D. Dissertation, Department of Linguistics, University of Massachusetts, 1981.

[Jacobs 88] P. Jacobs, "Relation Driven Text Skimming", General Electric Co. Technical Report, 1988.

[Kahaner, 91] The Kahaner email corpus.

[Karttunen, 68] Lauri Karttunen, Discourse Referents, in J. McCawley, (ed.), *Syntax and Semantics*, Academic Press, New York.

[Landman 86] F. Landman, "Pegs and Alecs.", *Linguistics and Philosophy*, 97-155, 1986.

[Luperfoy, 91] Susann Luperfoy, "Discourse Pegs: A Computational Treatment of Context-Dependent Referring Expressions", Ph.D. Dissertation, Department of Linguistics, University of Texas at Austin.

[Mauldin 89] Michael L. Mauldin, "Information Retrieval by Text Skimming", Carnegie Mellon University Technical Report CMU-CS-89-193.

[Norvig, 92] Peter Norvig, "Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp", Morgan Kaufmann, 1992.

[Paik et al., 93] Woojin Paik, Elizabeth D. Liddy, Edmund Yu, and Mary McKenna, "Interpretation of Proper Nouns for Information Retrieval", *Preliminary Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, March 21-24, 1993.

[Radford, 88] Andrew Radford, "Transformational Grammar", Cambridge University Press, 1988.

[Sidner, 79] C. L. Sidner, "Towards a Computational Theory of Definite Anaphora Comprehension in Discourse", Ph.D Thesis, Electrical Engineering and Computer Science, M.I.T., 1979.

[Webber, 78] B. Webber, "A Formal Approach to Discourse Anaphora", Ph.D. Thesis, Department of Applied Mathematics, Harvard University, 1978.

MENTIONS AND HYPOTHESES	PEGS
1. "Bill Clinton, 45" Name: Bill.Clinton Age: 45 Norm: Clinton	[1. Bill.Clinton]
2. "Mr. Clinton" Name: .Clinton Gender: Male Norm: Clinton	[1]
3. "Ms. Hilary Clinton" Name: Hilary.Clinton Gender: Female Norm: Clinton	[2. Hilary.Clinton]
4. "U.S. President Clinton" Name: .Clinton Occupation: HeadofState Norm: Clinton	[2, 1]
5. "First Lady Hilary Clinton" Name: Hilary.Clinton Gender: Female Occupation: FirstLady Norm: Clinton	[2] Leads to breaking of link from Mention 4 to Peg 2.

Figure 1: Coreference and disambiguation