Originally presented at the SIGLEX workshop on "Acquisition of Lexical Knowledge from Text", June 21, 1993 and appears in the proceedings of that workshop pgs. 32-43 (available from the ACL). This slight revision of that presentation appears in Corpus Processing for Lexical Acquisition, MIT Press, 1996, edited by Boguraev and Pustejovsky, pages 21-37.

Internal and External Evidence in the Identification and Semantic Categorization of Proper Names

David D. McDonald [1]

Brandeis University

Abstract

We describe the Proper Name recognition and classification Facility ("PNF") of the SPARSER natural language understanding system. PNF has been used successfully in the analysis of the names found it unrestricted texts in several sublanguages taken from online news sources. It makes its categorizations on the basis of 'external' evidence from the context of the phrases adjacent to the name as well as the standard 'internal' evidence within the sequence of words that make up the name. A semantic model of each name and its components is maintained and used for subsequent reference.

We describe PNF's operations of delimiting, classifying, and semantically recording the structure of a name; we situate PNF with respect to the related parsing mechanisms within Sparser; and finally we work through an extended example that is typical of the sorts of text we have applied PNF to.

[1] Author's address: 14 Brantwood Road, Arlington, MA 02174-8004 davidmcdonald@alum.mit.edu

Proper names are the Rodney Dangerfield of linguistics.

They don't get no respect.

1 Introduction

Within theoretical linguistics, proper names are relegated to the 'periphery' of the language. Unlike the 'core' phenomena of long-distance movement, case marking, argument structure, and the like, there is an assumption that the study of proper names will yield no deep insights into the nature of language or illustrate principles with broad application. When thought about at all, proper names, like numbers, dates, ages, etc. are imagined to be easy to analyze, or, in a computational context, to generate or understand.

People who actually work with proper names know better. The accurate identification and semantic categorization of names has proved to be anything but easy, yet understanding names, and their patterns of initial and subsequent reference, is central to the analysis of the extended, unrestricted texts that have become the focus of research in the natural language processing community.

From the study of these texts (principally newspaper articles or specialized sets of messages) we know that proper names exhibit an enormous diversity, but that they also have a systematic and compositional structure that can be captured in a grammar. This grammar is more lexical and less syntactic, and its links to semantics are far tighter than is the case for the so-called core grammar, but it is still a principled structure with a generative capacity that allows never-before-seen instances of proper names to be reliably recognized and semantically understood.

This paper will argue that this grammar must be context sensitive, and that the semantic interpretation of proper names should be mediated by semantic structures that denote names, per se, with only an indirect link to the individuals being named. In the next section we introduce this notion of context sensitive analysis for proper names and the motivations behind it. In section three we go through our procedure for analyzing names and provide examples of the kinds of complexities that can be encountered. In section four we discuss the setting for this analysis as part of the Sparser language understanding system, and finally in section five we step through a fairly complex example.

2. Internal versus External Evidence

The requirement that a grammar of proper names must be context-sensitive derives from the fact that the classification of a name involves two complementary kinds of evidence, which we will term 'internal' and 'external'. *Internal evidence* is taken from within the sequence of words that comprise the name. This can be definitive criteria, such as the presence of known 'incorporation terms' that indicate companies ("Ltd.", "G.m.b.H."); or it can be heuristic criteria such as abbreviations or known first names, which often indicate people. Name-internal evidence is the only criteria considered in virtually all of the name recognition systems that are reported as

part of state of the art information extraction systems (see e.g. Rau 1991, Alshawi 1992, Cowie et al. 1992), most of which depend on large (~20,000 word) gazetteers and lists of known names for their relatively high performance.

By contrast, *external evidence* is provided by the context in which a name appears. The basis for this evidence is the obvious observation that names are just ways to refer to individuals of specific types (people, churches, rock groups, etc.), and that these types have characteristic properties and participate in characteristic events. The presence of these properties or events in syntactic relation with a proper name can be used to provide confirming or criterial evidence for a name's category. External evidence is analyzed in PNF in terms of substitution contexts, and operationalized in terms of context-sensitive rewrite rules.

External evidence is necessary for high accuracy performance. One obvious reason is that the predefined word lists so often used as internal evidence can never be complete. Another is that in many instances, especially those involving subsequent references, external evidence will override internal evidence. In the final analysis it is always the way a phrase is used—the attributions and predications it is part of—that make it a proper name of a given sort. Without the consideration of external evidence, this definitive criteria is missed, resulting in mistakes and confusion in the state of the parser.[2]

An additional reason for using external evidence, and one with considerable engineering utility from the point of view of the grammar writer, is that the inclusion of external evidence into the mix of available analysis tools reduces the demands on the judgments one requires of internal evidence. The internal analysis can get away with a weaker (less specific) categorization about which it can be more certain, and the categorization can then be refined as external evidence becomes available. Lacking definitive internal evidence, one can initially label a segment simply as a 'name', and then later strengthen the judgment when, e.g., the segment is found to be adjacent to an age phrase or a title, whereupon context-sensitive rewrite rules are triggered to re-label it as a person and to initiate the appropriate semantic processes.

This kind of staged analysis is a requirement when the conclusions from internal evidence are ambiguous. It is not uncommon, for example, for the names of a person and of a company in the same news article to share a word, as when the company is named after its founder. A subsequent reference using just that word cannot be definitively categorized on internal evidence alone, and must wait for the application of external evidence from the context. In the event that the context is inadequate, as when it involves a predication not in the grammar, the further analysis of such 'name' segments can be left to default judgments by statistical heuristics operating after a first

---

[2] Relying solely on name lists has led to some funny errors, for example mistaking the food company *Sara Lee* for a person. Even some external evidence such as a title can be inadequate if it is considered apart from the wider context of use, as in *General Mills*—both of which are actual mistakes made by an otherwise quite reasonable program some years ago (Masand & Duffey 1985).

pass by the parser, and the stronger categorizations then tested for coherency as the parse is resumed.[3]

To make this discussion concrete, we will ground the remainder of this paper in a description of "PNF", the proper name facility of the SPARSER natural language understanding system, paying particular attention to how PNF uses external evidence and deploys its semantic model of names and their referents to handle ambiguities such as the ones noted just above. In a blind test of an earlier implementation on "Who's News" articles from the Wall Street Journal, PNF performed at nearly 100% when the name appeared in a sentence in the sublanguage for which a full grammar had been prepared. We are currently testing a new implementation on a more diverse set of texts.

Space will not permit a comparison of this algorithm with other approaches to proper names beyond occasional remarks and references. As far as we know this is the only treatment of proper names that uses context-sensitive rewrite rules for the analysis of external evidence, however the FUNES system of Sam Coates-Stephens (1992) is very similar to this work in making essential use of external evidence, and Coates-Stephens's extensive research into proper names is an important contribution to the field; we have adopted some of his terminology as noted below.

3 An overview of the procedure: Delimit, Classify, Record

The goal of the Proper Name Facility in Sparser is to form and interpret full phrasal constituents—noun phrases—that fit into the rest of a text's parse and contribute to the analysis of the entire text just like any other kind of constituent. That is to say that PNF is a component in a larger natural language comprehension system, and not a standalone facility intended for name spotting, indexing, or other tasks based on skimming. This integration is essential to the way the PNF makes its decisions; it would not operate with anything like the same level of performance if it were independent, since there would then be no source of external evidence.

To analyze an instance of a proper name for use by a full natural language comprehension system we must

(1) *delimit* the sequence of words that make up the name, i.e. identify its boundaries;

(2) *classify* the resulting constituent based on the kind of individual it names; and

(3) *record* the name and the individual it denotes in the discourse model as our interpretation of the constituent's meaning.

For other parts of Sparser's grammar, these three actions are done with one integrated mechanism much as they would be in most other systems. Constituents are

---

[3] In the case of a company and a person with the same name, a well edited publication is unlikely to use the ambiguous word to refer to the founder without prefixing it with "Mr." or "Ms." as needs be, so a word with both person and company denotations but without external evidence can be assumed to be referring to the company.

initiated bottom up by the terminal rules of the lexicalized grammar, and compositions of adjacent constituents are checked for and nonterminal nodes introduced as the grammar dictates.[4] The grammar's production rules delimit and classify (label) constituents in one action: the classifications are given by the productions' lefthand sides, and the new constituents' boundaries by the sequence of daughter constituents on the rules' righthand sides, with the new constituent's denotation given by an interpretation function included directly with the rule and applied as the rule completes.

This normal mode of operation, however, has not proved workable for proper names. The reason has to do with the central problem with names from the point of view of a grammar, namely that in unrestricted texts the total set of words that names can be comprised of can not be known in advance. The set is unbounded, growing at an apparently constant rate with the size of the corpus, while the growth of other classes of content words tapers off asymptotically (Liberman 1989). This means that we cannot have a lexicalized grammar for proper names since the bulk of the names we will encounter will be based on words that are unknown at the time the grammar is written.

Complicating this picture is the fact that virtually any normal word can do double duty as part of a name: " … *Her name was equally preposterous. April Wednesday, she called herself, and her press card bore this out.* " MacLean 1976 pg.68. This means that one either introduces a massive and arbitrary ambiguity into the normal vocabulary, allowing any word to be part of a name, or one looks for another means of parsing proper names, which is the course that was taken with PNF. For PNF we have separated the three action—delimiting, classifying, and recording—into distinct and largely independent operations.

2.1 Delimit

The delimit operation is based on a simple state machine rather than on the application of rewrite rules. This reflects that fact that the internal constituent structure of a proper name is typically a sequence of an indefinite number of elements with local groupings into embedded names that are sisters in the name as a whole. Because of the indefinite length of the subsequences, a phrase structure account would impose a tree structure on the components of the name that was just an artifact of the rule application machinery rather than a reflection of the actual constituency.

PNF's delimitation algorithm simply groups any contiguous sequence of capitalized words (including 'sequences' of length one).[5] This is virtually always the

---

[4] Sparser uses a moderately complex control structure to ensure that the grammar is applied deterministically (each span of text only ever receives one analysis) and that the semantic interpretation is monotonic and indelible. The grammar—the rules of constituent combination—is specified by a set of phrase structure rules, but its runtime operations are more like those of a categorial grammar where the identity of each constituent label determines its possibilities for combination; the original rules are not a material part of the process as they would be in a conventional phrase-structure based parsing algorithm.

[5] It is reasonable to depend upon the existence of mixed-case text, since the number of online sources that use only uppercase is rapidly diminishing and will probably disappear once all of the Model-33

correct thing to do as the example below illustrates, though the exceptions have to be treated carefully as discussed later.

> "*The Del Fuegos, O Positive, and We Saw the Wolf will perform acoustic sets in Amnesty International USA Group 133's Seventh Annual Benefit Concert at 8 p.m. on Friday, March 19, at the First Parish Unitarian Universalist Church in Arlington Center*." (Arlington Advocate, 3/18/93)

A sequence is terminated at the first non-capitalized word or comma; other punctuation is handled case by case, e.g. "&" is taken to extend sequences, and periods are taken as terminators unless they are part of an abbreviation.

2.2 Classification

Classifying a proper name is a two-step process. First, the regular parsing routines are applied within the delimited word sequence. This embedded parsing process introduces any information the grammar has about known words or phrases. Such information is the basis for the most of the structure within a proper name, and provides the name-internal evidence on which the classification will be based at this stage. For PNF this includes:

- embedded references to cities or countries, e.g. "*Cambridge Savings Bank* ".
- open class 'keywords' like "*Church*" or "*Bank*" (following Coates-Stephens terminology), and the incorporation-terms used by companies of various countries when giving their full legal names ("*Inc*." in the U.S.A., "*P.T*." in Indonesia, "*G.m.b.H*." in Germany, etc.).
- the relatively closed class of stylized modifiers used with people like "*Jr*.", "*Sir*", "*Mr*.", "*Dr*.".
- items used for heuristic classification judgments (the items above are definitive) such as including initials (a strong indicator that the name refers to a person or a company based on a person's name), punctuation like "&" (a company marker), or ambiguous modifiers like "II" (which invariably means 'the second', but may be used with Limited Partnerships as well as people).

The parsing stage will reveal when the capitalized word based delimitation process has included too much. One such case is of course when a proper name appears just after the capitalized word at the start of a sentence: "*An Abitibi spokesman said …*". This is handled by recognizing closed-class grammatical functional words as such during the embedded parse, and resegmenting the word sequence to exclude them.

Teletypes and other 6-bit data entry terminals in the world are finally junked. In any event, to handle all-uppercase texts within PNF it is only the delimitation algorithm that must be changed. A good approximation of the needed segmentation is independently available from the distribution of function words and punctuation in any text. In this example these are the commas, the apostrophe-s, "*in*", "*at*", and "*on*". A mistake would be made in "*We Saw the Wolf*" [sic] which in any event will be problematic without external context.

Another, more interesting case is where we have a sequence of modifiers prefixed to a proper name that are themselves proper names, e.g. "*Giant Group said [it is] seeking to block a group led by Giant Chairman Burt Sugarman from acquiring …*". In this situation there is no hope for correctly separating the names unless the grammar includes rules for such references to companies and titles, in which case they will appear to the classification process as successive edges (parse nodes) with the appropriate labels ('company', 'title') so that they can be appreciated for what they are and left out of the person's name.[6] It is important to appreciate that all of these considerations only make sense when one is analyzing proper names in the context of a larger system that already has grammars and semantic models for titles and employment status and such; they are hard to justify in an application that is simply name spotting.

In practice, the operations of delimiting and classifying are often interleaved, since the classification of an initially delimited segment can aid in the determination of whether the segment needs to be extended, as when distinguishing between a list of names and a compound name incorporating commas, e.g. "*… a string of companies – including Bosch, Continental and Varta – have announced co-operative agreements …*" (The Financial Times, 5/16/90) versus "*HEALTH-CARE FIRM FOLDS: Wood, Lucksinger & Epstein is dissolving its practice*." (Wall Street Journal 2/26/91). We will describe this process in the extended example at the end of the paper.

Once the words of the sequence have been parsed and edges introduced into the chart reflecting the grammar's analysis, the second part of the classification process is initiated and a state machine is passed over that region of the chart to arrive at the most certain classification possible given just this name-internal evidence. If no specific conclusion can be reached, the sequence will be covered with an edge that is simply given the category 'name', and it will be up to external evidence to improve on that judgment as will be described later. If a conclusion is made as to the kind of entity being named, then the edge will be labeled with the appropriate semantic category such as 'person', 'company', 'newspaper', etc.

2.3 Recording

The recording process now takes over to determine what the new edge should have as its denotation in the discourse model. Before this denotation is established, PNF's representation of the name is just a label and the sequence of words and edges (parse nodes) internal to the name (e.g., edges over an embedded reference to a city or region). What we are providing now via the recording process is a structured representation of the name qua 'name'—a unique instance of one of the defined classes of names that reifies this specific pattern of words and embedded references.

---

[6] It is perhaps a matter of judgment to hold that a person's title is not a part of their name, but that policy appears to be the most consistent overall since it permits the capitalized premodifier version of a title of employment (e.g. "*Chairman*") and its predicative lowercase version (as in an appositive) to be understood as the same kind of relationship semantically—a different relationship than the one between a person and her conventional title such as "*Dr*.".

Including names as actual entities in the semantic model, rather than just treating them as ephemeral pointers to the individuals they name and only using them momentarily during the interpretation process, provides us with an elegant treatment of the ambiguity that is intrinsic to names as representational devices. Real names, unlike the hypothetical 'rigid designators' entertained by philosophers, may refer to any number of individuals according to the contingent facts of the actual world. We capture this by making the denotation of the lexico-syntactic name—the edge in Sparser's chart—be a semantic individual of type 'name' rather than (the representation of) a concrete individual. The name object in turn may be then associated in the discourse model with any number of particular individuals of various types: people, companies, places, etc. according to the facts in the world. The ambiguity of names is thus taken not to be a linguistic fact but a pragmatic fact involving different individuals having the same name.

The structure that the semantic model imposes on names is designed to facilitate understanding subsequent references to the individuals that the names name. The type of name structure used predicts the kinds of reduced forms of the name that one can expect to be used. This design criteria was adopted because, again, the overarching purpose of PNF is to contribute to the thorough understanding of extended unrestricted texts. This means that it is not enough just to notice that a given name has occurred somewhere in an article, something that is easy to do by looking for just those cases where the full company name is given with the 'incorporation term' that well edited newspapers will always provide when a company is introduced into a text, e.g. "*Sumitomo Electric Industries, Ltd*.".

The model PNF constructs for the name must be rich enough to be able to recognize that that same individual is being talked about later when it sees, e.g., "*Sumitomo Electric*" (or "*the company*"). In addition, PNF must be able to distinguish that individual from subsequent references to other companies that share part of its name: "*Sumitomo Wiring Systems*", or to correctly deduce a subsidiary relationship "*Sumitomo Electric International (Singapore)*". By the same token, people and companies or locations that share name elements should be appreciated as such: "*the Suzuki Motors Company … Osamu Suzuki, the president of the company*".

To facilitate such subsequent references, not only does each proper name receive a denotation as an entirety, but the words that comprise it are also given denotations which are related, semantically, to the roles the words each played in that name and in the names of other individuals. Thus the word "*Suzuki*", for example, is taken to always denote the same semantic object, prosaically printed as #<name-word "suzuki">. In turn this object is related to (at least) two individuals—to the car company by way of the relation 'first-word-in-name', and to its president by the relation 'family-name'.

4 The setting for the process

In order to supply the external evidence needed to accurately categorize proper names and understand them semantically, a language understanding system must include

grammars (and their attendant semantic models) for properties and event-types that are characteristically associated with the kinds of individuals that the names name, and these grammars should have as broad a coverage as possible.

SPARSER has been applied to understanding news articles about people changing their jobs (particularly the Wall Street Journal's "Who's News" column), and with a lesser competence to articles on corporate joint ventures and quarterly earnings. As a result, it has quite strong semantic grammars for some of the very most frequent properties of companies and people in business news texts: the parent–subsidiary relationship between companies, age, titles, and for a few of the more common event-types.

A complementary consideration is what approach will be taken to such relatively mundane things as punctuation, capitalization, or abbreviations. For SPARSER, since it is designed to work with well-edited news text written by professional journalists, punctuation is retained and there are grammar rules that appreciate the (sometimes heuristic) information that it can provide. The whitespace between words is also noted (newlines, tabs, different numbers of spaces) since it provides relatively reliable evidence for paragraphs, tables, header fields, etc., which in turn can provide useful external evidence.

Additionally, SPARSER is designed to handle a constant, unrestricted stream of text, day after day, and this has led to a way to treat unknown words that allows it to look at their properties without being required to give them a long-term representation which would eventually cause the program to run out of memory.

To illustrate how these work, and at the same time establish the setting in which proper name processing takes place, we will now describe the lower levels of SPARSER's operation, starting with its tokenizer and populating the terminal positions of the chart.

4.1 Tokenizing

The tokenizer transduces characters into objects representing words, punctuation, digit sequences, or numbers of spaces. It is conservatively designed, just grouping contiguous sequences of alphabetic characters or digits and punctuation, and passing them all through to be the terminals of the chart. Even the simplest compounds are assembled at the chart level by sets of rules that are easily changed and experimented with. For example, rather than conclude within the tokenizer that the character sequence "$47.2 million" is an instance of money, the tokenizer just passes through six tokens, including the space.

A word is 'known' if it is mentioned in any of the rules of the grammar.[7] A known word has a permanent representation, and the tokenizer finds and returns this object when it delimits the designated sequence of characters. The 'token-hood' of any

---

[7] Note that since Sparser uses a lexicalized semantic grammar, words have preterminal categories like 'title' or 'head-of-Co-phrase' (e.g. "*company*", "*firm*", "*enterprise*"), or are often treated just as literals as with prepositions or with words like "*ago*" in "44 years ago".

given instance of the word type is represented by the word filling a particular location in the chart.

The tokenizer separates the identity of a word from its capitalization. A word is defined by the identity of its characters. The pattern of upper and lowercase letters that happens to occur in a given instance is a separate matter, and is represented in the chart rather than with the word.[8] When the chart is populated with terminals, each position records the word that starts there, its capitalization, and the kind of whitespace that preceded it, all given as separate fields of the position object. The token scan is done incrementally in step with the rest of the SPARSER's actions.

4.2 Word-triggered operations

Sparser's processing is organized into layers. Tokenizing and populating the terminals of the chart is the first level, then comes a set of special operations that are triggered by words or their properties (e.g. the property of ending in "*ed*" or of consisting solely of digits). The the next layer is the application of phrase structure rules, and finally there is the application of heuristics in order to spanning the gaps caused by unknown words. Semantic interpretation is done continuously as part of what happens when a rule completes and an edge is formed. We will not describe the last two layers (see McDonald 1992 for a description of the phrase structure algorithm), but will briefly describe the word-level operations since they include triggering PNF.

Actions triggered just by the identity of a word include forming initials and known abbreviations, and particularly the recognition of multi-word fixed phrases which we call "polywords" following Becker (1975). Polywords are immutable sequences of words that are not subject to syntactically imposed morphological changes (plurals, tense) and that can only be defined as a group. Polywords are a natural way of predefining entities that have fixed, multi-word names such as the countries of the world, the states of the US, major cities, etc. Instances of this relatively closed class of individuals are a valuable kind of evidence in the classification of proper names.

When PNF finishes the recognition and classification of a new name, it adds to the grammar a polyword rule for that sequence of words, with the recorded name-object as the polyword's denotation. This permits the process to be short-circuited the next time the name is seen. Note that this does not stop PNF from running its delimiting operation the next time that sequence is seen; it only speeds up the classification and recording. If we allowed the polyword operation to take complete precedence, we would never see the longer word sequences that embed known names ("*New York Port Authority*").

There are also special rules that allow paired punctuation (parentheses, quotation marks, etc.) to be grouped even if the words separating them are not all known. This is particularly useful for picking up nicknames embedded within a person's name since that nickname will often be given as a word in parentheses embedded within the

---

[8] One can deliberately define a capitalization-sensitive version of a word, e.g. to syntactically distinguish titles in pre-head position from those in appositives or elsewhere. In such cases there is a distinct word object for the capitalized version, with a link to the case-neutral version of the word.

person's name ("*Justice Byron (Whizzer) White*"). Subsidiaries of companies are often marked for their geographical area in the same way, e.g. "*manufactured by UNIVERSAL FLUID HEADS (Aust.) PTY. LTD.*" (taken from the name plate on a camera tripod).

The first check at the word-level is for actions triggered by a word's properties, particularly here the properties of its characters. This is how compound numbers are formed ("*42,357.25*"), triggering off words that consist of sequences of digits ("*42*"), and it is how PNF is triggered. Every time a chart position is reached that indicates that the following word is capitalized, PNF is called. PNF then takes over the process of scanning the successive terminals of the chart until it scans a word that is not capitalized, calling other SPARSER mechanisms like polyword recognition or phrase structure rewrite rules as needed.

When PNF is finished, its results are given in the edge it constructs over the sequence of capitalized words and selected punctuation that comprise the name. Since Sparser uses a semantic grammar, the label on the edge is the constituent's classification—a semantic category like 'person'. There is also conventional label (always NP for a name) included with the edge for default or heuristic rules of phrasal formation; see McDonald (1993) for the details of this two label system.

5 Walking through an example

In this final section of this paper we will look at the processing of the following paragraph-initial noun phrase from the Wall Street Journal of 10/27/89, article #34:

> "*An industry analyst, Robert B. Morris III in Goldman, Sachs & Co.'s San Francisco office, said ...*"

The capitalization of the very first word "*An*" triggers PNF, but the delimitation process stops immediately with the next word since it is lowercase. The classification pass through the (one word) sequence shows it to be a grammatical function word, and classification applies the heuristic 'single word sequences consisting solely of a non-preposition function word are not to be treated as names' and takes no further action. PNF is then finished; the article reading of "*An*" will have been introduced into the chart during classification; and the scan moves on.[9]

As the parse moves forward, the title phrase "an industrial analyst" is recognized and the comma after it is marked as possibly indicating an appositive (or also a list of titles, though this is less likely).

---

[9] We have yet to see a company whose name was "*The*", though an ad running in The Boston Phoenix during May of 1993 included a graphic for an upcoming entertainer named "the The". Of course there are companies like Next Inc. and On Technology, which, like the names of race horses or boats, add spice to the grammarian's life by overloading the interpretations of closed-class words. The only consistent treatment we have arrived at for these ("*On*" referring to the company does occur in sentence-initial position) is to treat the words as ambiguous and to introduce two edges into the chart, one for each reading. We only do this if the full name of the company appeared earlier in the article, however, at which time the preposition will have received its denotation as an element of a name and the basis of the ambiguity can be established.

PNF is triggered again by the capitalization of "*Robert*", and the delimitation process takes it up to the word "*in*". Running the regular rules of the grammar within that sequence uncovers the initial and the generation-indicator "III" for 'the third'. We do not maintain any lists of the common first names of people or such, so consequently both "*Robert*" and "*Morris*" are seen as unknown words. The initial and generation-indicator are enough, however, to allow the sequence to be reliably classified as the name of a person.

Given that classification, an edge is constructed over the sequence and given the label 'person', and the recording process constructs a name object for the edge's denotation. The pattern given by the classifier is 'name – initial – name – generation-indicator', from which the name subtype 'person's name with generation' can be instantiated. This type of name object takes a sequence of first names or initials, a last name (the word before "III"), and then the "III" in a slot that also gets words like "*Junior*". Let us call this new name-individual Name-1.[10]

Part of the recording process is the creation of denotations for the words "*Robert*" and "*Morris*". Discourse model objects are created for them of type 'single word element of a name', and rules are added to the grammar so that the next time PNF sees them in a text, the embedded parse during classification will immediately recover those same objects. In addition, we attribute properties to the names (the semantic objects) 'Robert' and 'Morris'—'Robert' is the first name of Name-1 and 'Morris' is its last name.

This policy of letting words like "*Morris*" denote single-word name objects with semantic links to the full names they are part of (with those names in turn linked to the people or other types of individual whose names they are) provides a very direct way to understand subsequent references involving just part of the original name (e.g. "*Mr. Morris*"), as we can trace the abbreviated name directly to the person just by following those links. (Of course the links will also take us to anyone else the who has been recorded in the world model who has that same last name, hence the need for a good discourse model that appreciates the context set up by the article being processed.)

Moving on, the rest of this example text is "*in Goldman, Sachs & Co.'s San Francisco office*" and the PNF is triggered again at the word "*Goldman*". This is seen as a one word sequence because the conservative delimitation process takes the comma just following as a reason to stop, and, again, it is an unknown word. Not being a function word and not including any reliable internal evidence, "*Goldman*" is spanned with an edge labeled just 'name' and just a new single-word name is recorded as its denotation. Given the significance of commas for name patterns, PNF also makes a note (sets a flag) that this comma was preceded by a name.

PNF immediately resumes with "*Sachs & Co.*", stopping the delimitation process when it recognizes the "'" and "*s*" tokens as constituting an apostrophe-s, which is a definitive marker for the end of a noun phrase. During the delimitation process, the abbreviation "*Co.*" will also have been recognized and expanded to "*Company*", and

---

[10] There is no interesting limit on the number of 'first names' a person can have, so we have not yet found it profitable to have any more structure in that field than simply an ordered sequence; consider "*M.A.K. Halliday"* or "*(Prince) Charles Philip Arthur George*".

the "*&*" noted and appreciated as being a punctuation mark that can appear in names. Punctuation is always handled during the course of delimitation, since the identify of the punctuation is crucial to whether the name sequence should be continued beyond the punctuation or stopped.

The presence of the "*&*" and the word "*Company*" are definitive markers of companies and the classification process will start the assembly of a pattern to send off to be recorded. In this case however, as noted earlier, there is what amounts to an interaction between classification and delimitation. Part of what the classifier knows about companies is that there is a profusion of cases where the name of the company is a sequence of words separated by commas (law firms, advertising agencies, etc.; any sort of partnership tends to use this name pattern). Appreciating this, the process looks for the contextual note about the preceding comma. Finding it, it observes that the name in front of the comma is not itself classified as a company (which would have indicated a list of companies rather than a single name), and it proceeds to assimilate the edge over "*Goldman*" and the comma into the name it is already assembling. Had there been still more 'stranded' elements of the name followed by commas, this would have been noted as well and those elements added in.

Occasionally the name of a company like this is given with "*and*" instead of the special punctuation character. Had that happened here, the fact that the "*and*" preceded the word "*company*" would have been sufficient evidence to take the whole sequence as the name of a single company, however if there had been no such internal evidence within any of the elements of the conjunction, they would have been grouped together as unconnected names spanned by a single edge labeled 'name', leaving it to external evidence from the context to supply a stronger categorization (both as to what category of name was involved and whether they were one name or several).

We can see this use of external evidence in operation with the next capitalized word sequence that PNF delimits, "*San Francisco*". With access to a good gazetteer we could have already defined San Francisco as the name of a city using a polyword. However, just by using external evidence and without needing any word list, we can conclude that it is a location, and probably a city, just by looking at its context: the word "*office*".

As said earlier, the availability of mechanisms that use external evidence like this allows PNF to make a weak analysis that can be strengthened later. In this case it will see "*San Francisco*" as a sequence of two unknown words. Without any internal evidence to base its judgment on, it can only (1) accept the sequence as a phrase and span it with an edge, indicating thereby that the words have a stronger relationship to each other than either has to its neighbors, and (2) give this edge the neutral label 'name'.

After PNF is done with "*San Francisco*", the phrase structure component of Sparser takes over. Sparser's rewrite rule facility includes context-sensitive as well as context-free productions, including for this case the rule

```
name -> location /  ___ "office"
```

This says that an edge labeled 'name' can be respanned with a new edge with the label 'location' when the name edge appears just in front of the word "office". Context

sensitive rules are handled in Sparser with the same machinery as context free rules, the only difference is what happens when the rule is completed. The context sensitive rule is coded as though it had a righthand side like a context free rule, in this case the pair of labels 'name' + "*office*". The only difference is that when this pattern is matched, instead of covering the whole righthand side with a new edge as would be done for a context free rule, with a context sensitive rule we just respan the one indicated constituent. In this case the 'name' is respanned by an edge labeled 'location'.

Similarly, if the name of the person in this example had been just "*Robert Morris*", where there would have been no available internal evidence to indicate its classification (rather than "*Robert B. Morris III*"), we could later have applied either of two context free rules: one working forwards from the definitively recognized title, the other backwards from the prepositional phrase 'in-company'.

```
name -> person / title "," ____
name -> person / ____ in-company
```

A repertoire of such context-sensitive rules or their equivalent is needed if a proper name classification facility is expected to work well with the open-ended set of name words found in actual texts; Sparser used a set of roughly 30 rules to handle the names in the blind test on the Who's News column mentioned earlier.

6 Conclusions

The combination of the internal evidence provided by PNF's delimit, classify, and record processes with the external evidence provided by the context in which a name appears as operationalized in a set of context-sensitive rewrite rules has proved nearly one hundred percent effective in identifying and semantically characterizing the names of people and companies that occurred in the specific, relatively small sublanguage where we originally developed and tested it.

As we have begun to move beyond this sublanguage to look at the wider range of proper names and capitalized sequences that occur in, e.g., an entire issue of the Wall Street Journal—deliberately staying within the register of carefully edited, largely informative, presentations of current events—we have found two things. The first that the vocabulary of what Coates-Stephens calls 'keywords', words that provide internal evidence about how a name is to be classified such as "newspaper", "(cruise) liner", etc. is enormous and will require semi-autonomous 'corpus mining' techniques or similar mass methods if it is to be dealt with adequately. That this should be so is not particularly surprising, since what these keywords typically are is just the common noun naming a kind of thing, but it is sobering to be reminded when working for a long time in just one sublanguage how many kinds of things there are.

The second is that the range of kinds of things that are conventionally represented in English using proper nouns and capitalization far exceeds what is seen in short articles on restricted subjects. A vast array of things have names: movies, books, legislation, parks, astronomical phenomena ("*the Milky Way*"), etc. To a certain extent this wider set of classes of individuals comes accompanied by keywords that serve to identify the class (e.g. "*Act*" "*Park*"), but equally often it is not, especially for things that are named by the use of titles or that are forced by law to have all different names (boats, race horses) and so tend to use ordinary words in unordinary ways.

The goal of our continuing work is less to assimilate this much extended space of possibilities into our proper name grammar (though for keywords that is not being neglected) than to refine our techniques for ensuring that the semantic classifications that are made are made accurately—that no guess work is incorporated into PNF on the basis of a few instances of what appears to be good heuristic internal evidence. The largest factor in achieving this goal has turned out to be having a complete set of the context sensitive rules that are used to render observations about the context in which a given class of named individual occurs into a useful form. This is leading to the development, some of it foreshadowed in McDonald 1993, of a system of what amount to rule-definition macros that take on the work of automatically constructing the set of context sensitive rules implied by each new predicate or attribute that is added to the grammar and semantic model. Human grammar writers too often forget to include even obvious cases, and the more that the knowledge of the grammar writer can be encoded in a system of automatic rule-writing facilities, the more successful the grammars will be in the long run.

7 References

Alshawi, Hiyan (ed.) (1992) *The Core Language Engine*, MIT Press.

Becker, Joe (1975) "The Phrasal Lexicon", in Schank & Webber (eds.) proceedings of TINLAP-1, ACM, 60-63.

Coates-Stephens, Sam (1992) "The Analysis and Acquisition of Proper Names for the Understanding of Free Text", *Computers in the Humanities*.

Cowie, Jim, Louise Guthrie, Yorick Wilkes, James Pustejovsky, and Scott Waterman (1992) "Description of the Solomon System as used for MUC-4", in the *Proceedings of the Fourth Message Understanding Conference: MUC-4* , June 1992, Morgan Kaufmann.

Liberman, Mark (1989) Panel presentation at the 27th Annual Meeting of the ACL, Vancouver.

MacLean, Alistair (1976) *The Golden Gate*, Fawcett Publications, Greenwich Connecticut.

Masand, Brij M. & Roger D. Duffey (1985) "A Rapid Prototype of an Information Extractor and its Application to Database Table Generation", working paper, Brattle Research Corporation, Cambridge, MA.

McDonald, David D. (1992) "An Efficient Chart-based Algorithm for Partial-Parsing of Unrestricted Texts", proceedings of the 3d Conference on Applied Natural Language Processing (ACL), Trento, Italy, April 1992, pp. 193-200.

_____ (1993) "The Interplay of Syntactic and Semantic Node Labels in Partial Parsing", proceedings of the International Workshop on Parsing Technologies, August 1993, 171-186.

Rau, Lisa F. (1991) "Extracting Company Names from Text", proceedings of the Seventh Conference on Artificial Intelligence Applications, February 24-28, 1992, IEEE, 189-194.