RC 20338 (01/16/96)
Digital Libraries

IBM Research Report

Extracting Names from Natural-Language Text

Yael Ravin
T. J. Watson Research Center
yael@watson.ibm.com

Nina Wacholder
CRIA, Columbia University
nina@cs.columbia.edu

IBM Research Division
Almaden . T.J. Watson . Tokyo . Zurich

Extracting Names from Natural-Language Text

Yael Ravin
Nina Wacholder

IBM Research
T. J. Watson Research Center
Yorktown Heights, NY 10598
yael@watson.ibm.com
nina@watson.ibm.com

Abstract

We describe Nominator, a module we developed to extract proper names from natural language text, which is currently being integrated into IBM products and services. Using fast and robust heuristics, Nominator locates names in text, determines what type of entity they refer to -- such as person, place or organization -- and groups together all the variant names that refer to the same entity. For example, "President Clinton", "Mr. Clinton" and "Bill Clinton" are grouped as referring to the same person. Each group is assigned a "canonical name", (e.g., "Bill Clinton") to distinguish it from other groups referring to other entities ("Clinton, New Jersey"). Nominator produces a dictionary, or database, of names associated with a collection of documents.

Table of Contents

TABLE OF CONTENTS

Introduction

Text processing applications, such as machine translation systems, information retrieval systems or even spellcheckers, share a common need -- they require some analysis of their input text into atomic units, or tokens, which the applications can then manipulate. A spellchecker, for example, needs to identify individual terms in the text in order to compare their spelling to its dictionary of known terms. For many applications, intensive natural-language preprocessing or parsing is not feasible, and so most commonly, these systems perform instead minimal tokenization of their input into individual, blank-separated, independent tokens.

However, it is well known that words in running text are not independent of each other. In fact, words often form multi-word expressions that should not be further decomposed. In particular, proper names of people, organizations, places, laws, and other entities, are important multi-word textual entities to identify and preserve. Names require special processing by most text applications: they cannot be translated, checked for spelling or interchanged with synonyms in the same way common words are. In addition, names are very good indicators of the content (or the meaning) of a piece of text. The knowledge they encapsulate can be used by applications dealing with textual information.

Identifying names in all their variations is of great value to many text processing applications. It allows the application to treat what appear to be different strings as references to the same entity. In an information retrieval application, for example, a user may ask for documents about "IBM". If the system can locate all the occurrences of "IBM" and all of its variants ("International Business Machines Corp." but not "International Business") as hits, its performance will be both more precise and more satisfactory to the user.

An automatic, fast and robust module, that identifies and extracts proper names from running text is a desirable enhancement for any text-processing application. At the Digital Library Department of IBM's T.J. Watson Research Center, we have developed Nominator to perform this task. (We thank J. Prager for the module's name.) The module is currently being integrated into IBM products and services. Outside of IBM, it was used by the Computer Science department at the University of Pennsylvania in their participation in the Machine Understanding Conference (MUC), sponsored by ARPA. In the Oct 95 MUC, UPenn came in third out of seven participants, due to a technicality. Once the technicality was removed, they ranked first in both recall and precision.

Nominator is a fully automatic set of heuristics to locate names in text, determine what type of entity they refer to -- such as person, place or organization -- and group together all the variant names that refer to the same entity. For each group of equivalent variant names, it produces a "canonical name" -- the most permanent and least ambiguous name -- which serves as the representative of the group. The canonical name may be one of the variants in the group, or it may be made up of portions of different ones.

For each occurrence of a name in a document, Nominator outputs its location in the text and its canonical form. In addition, it also outputs a dictionary, or database, of names associated with a collection of text documents. Each entry in the dictionary is a canonical name, listed with its entity type and all its variants from the entire collection.

The input to Nominator is running text, such as news stories, Web pages, or computer-related literature. The input is minimally tokenized - that is, separated into individual words and sentences. (This tokenization is performed by Sentsep, another tool developed at Watson, by R. Byrd.) The module's heuristics rely on patterns of typography (capitalization and punctuation) and on contextual information to identify and classify proper names. They also make use of three "authority" lists: 1600 tags and other name indicators, such as "Mr." or "Inc."; 700 most common place names, such as "USA" or "Paris"; and about 20,000 first names. (A full description of these lists is found in Appendix B.)

Nominator is written in C and consists of about 6000 lines of code. It currently runs on AIX. On a RISC/6000 model 970 machine (SPECint92 -- 57.50; SPECfp92 -- 99.20), it can process 40 MB of text an hour. We continue to work on improving its performance.

The interest in identifying names in text has increased recently and has produced several commercial name identification products, most notably NameFinder from Carnegie Group and NameTag from SRA International, announced only a few months ago. Some of the Nominator functions do not seem to be available (yet) from NameFinder, such as finding place names or identifying all the occurrences of all variants of a given name. An important feature of Nominator is that it identifies names without recourse to an already established database of names, because applications cannot be assumed to maintain such a database. Carnegie Group is currently working on such a feature for NameFinder. SRA's NameTag, by contrast, seems to have all the features that our Nominator does.

Apart from clear differences in functionality between name processing modules, it is difficult to evaluate the quality of the output they produce due to the lack of benchmark tests available. In a manual, in-house evaluation, Nominator performed remarkably well. The names identified by a linguist in 88 Wall Street Journal articles were compared to those identified by the module. A more detailed evaluation is found in Appendix A. It concluded that Nominator had a very high recall of 97.8%. That is, it identified 97.8% of all the names occurring in the text. Its precision was also high - 91.8% -- that is, 91.8% of the names it identified were indeed valid names. The latest numbers disclosed by Carnegie Group, based on their own evaluations, ranged between 60% and 70%. See (Hayes 1994). We do not have information about the quality of SRA's NameTag output.

As for speed, we have no information about NameFinder, but SRA claims that NameTag processes 35,000 characters per second on a UNIX machine (unspecified model). This is about 3 times faster than the performance we report above. According to their literature, SRA uses finite-state technology to implement their name finding heuristics. We are currently studying the use of such technology ourselves, to improve performance. (See Section 7.)

Non-commercial name identification has also been the subject of ongoing research in various academic settings. See the survey of the literature in Appendix C.

In the following sections we discuss some assumptions and principles that guide our approach to the analysis of names. These have led us to define four steps in the process of extracting names from text:

- Form a candidate name list
- Split candidates into smaller names
- Group names into equivalence classes
- Aggregate classes across documents

We describe in detail the heuristics involved in each of those steps in Sections 2-6. More technical detail is available in pseudo-code or flow-charts from the authors.

Section 1: Some principles and assumptions

We chose to design Nominator as a set of heuristics which operate on characters and strings and do not employ more intensive natural-language processing tools such as parsers, expert systems or inference engines. This design choice differentiates our approach from the one taken by several academic projects involving name identification (See Mani & al. 1993 and McDonald 1993). It makes the module fast and robust; but it also limits the extent to which the program can "understand" the text being analyzed and resolve potential ambiguity. Many word-sequences that are easily recognized by human readers as names are ambiguous for Nominator, given the restricted set of tools available to it. In ambiguous cases, the module is designed to make conservative decisions, following the principle that "noise" (the inclusion of non-names or of non-name parts in otherwise valid name sequences) is preferred to data omitted, because the more precise information remains recoverable and is not lost.

For similar reasons of robustness and efficiency, Nominator does not perform an intensive analysis of the context in which names occur. It does not build a discourse representation or a model of the domain described in the text, as do the approaches referred to above. Rather, its notion of context is the optimal unit of text on which to operate. We find a single document - a news story, a technical report, or a legal case description to be such unit. We rely on a stylistic convention respected in most well-edited text which defines the document as a meaningful unit of context for names: entities are typically introduced once with their most explicit name and then referred to by shorter, more informal variants. The first three steps we have defined for the process of extracting names take place within the context of a single document. First a candidate name list for the entire document is collected. Then the candidate names on the list are compared to each other and the list is further refined: some names are discarded; others are modified. In the third step the list is divided into groups of equivalent names. Each group is assigned a canonical name and a type - person, organization, etc.

Nominator does not have access to any knowledge of the world and does not reason about it to infer, for example, that "the President" and "William Clinton" are two names of the same entity at a certain time. However, it does make limited use of the overall context (i.e., the document collection). After all documents have been analyzed and their canonical names extracted, in the fourth step, identical canonical names from different documents are aggregated and all of their variants combined. Thus, if in one document "President Clinton" was a variant of "William Clinton", while in another document "Governor Clinton" was a variant of "William Clinton", both will now be variants of one aggregated "William Clinton" group. In this sense, the module uses the larger context to "learn" more variants for a given name. In addition, identical canonical names may acquire different types, due to differences in the information available in the documents from which they were extracted. Thus, "Jordan Hills" may be of type PLACE?, in one document where Nominator did not have enough evidence to make a firmer decision; while in another document, in the context of "Mr. Hills", it categorized "Jordan Hills" as PERSON. When the identical canonical names are aggregated, the "stronger" type overrides the weaker one.

Finally, naming conventions vary in different domains. Nominator was developed primarily to deal with names common in the news. To identify product names, chemicals, or drug names, additional heuristics will be needed. Similarly, different natural languages have their own conventions for naming and will require to adapt our mostly English-based heuristics to their own data. Even within the same domain and the same language, naming conventions are sometimes disregarded by people. People enjoy creating novel and unconventional names, giving themselves perhaps even more license to do so than in other aspects of language, such as the introduction of alternate spellings or technical terms. A store named "Mr. Tall" and a woman named "April Wednesday" (McDonald 1993) come to mind. Nominator may fail to correctly analyze these unconventional names.

Section 2: The name extraction procedure

In this section we present an overview of the name-extracting procedure. We illustrate the process with the following sample document, taken from the TIPSTER CD-ROM collection (NIST 1993).

Nominator forms a candidate name list by scanning the tokenized document and collecting all sequences of capitalized tokens (or words) and selected prepositions. Some exceptions are discussed in Section 3.

The candidate list extracted for the sample document contains:

<DOC>
<DOCNO> WSJ881117-0178 </DOCNO>
<HL> Law: Non-Lawyers May Get 'Partner' Titles Under New Washington,
D.C., Bar Code </HL>
<AUTHOR> Jill Abramson (WSJ Staff) </AUTHOR>
<SO> </SO>
<IN> FIN </IN>
<DATELINE> WASHINGTON </DATELINE>
<TEXT>

In a move that would represent a major break with tradition in the legal profession, law firms in this city may become the first in the nation to reward non-lawyers with the cherished title of partner.

The District of Columbia Bar has recommended adoption of a new code of ethics that would permit such a move, reflecting broad changes in the kind of legal services offered by the city's firms. The new ethics code must be approved by the D.C. Court of Appeals before it goes into effect and will apply only to lawyers and law firms that practice here.

The professional conduct of lawyers in other jurisdictions is guided by American Bar Association rules or by state bar ethics codes, none of which permit non-lawyers to be partners in law firms.

The ABA has steadfastly reserved the title of partner and partnership perks (which include getting a stake of the firm's profit) for those with law degrees.

But Robert Jordan, a partner at Steptoe & Johnson who took the lead in drafting the new district bar code, said the ABA's rules were viewed as "too restrictive" by lawyers here. "The practice of law in Washington is very different from what it is in Dubuque," he said. ...

District of Columbia Bar
D.C. Court of Appeals
ABA
Robert Jordan
Steptoe & Johnson
ABA
Washington
Dubuque
...
Mr. Jordan of Steptoe & Johnson
...

Each candidate name is then examined to detect if it contains several independent names, and if so, split into as many names as are found in it. This is a challenging task, since the internal structure of such name combinations can be very complex. In our sample,

Mr. Jordan of Steptoe & Johnson

is split into

Mr. Jordan

and

Steptoe & Johnson.

Without recourse to semantics or world knowledge, we do not always have sufficient evidence. In such cases we prefer to err on the conservative side and not split. This explains the presence of "names" such as "American Television & Communications and Houston Industries Inc." or "Dallas's MCorp and First RepublicBank and Houston's First City Bancorp. of Texas" in our results. We discuss the details of the heuristics for splitting names in Section 4.

The module then refines the expanded list of candidates by comparing the sequences to each other and removing (parts of) sequences that are found invalid. For example, "August" or "May" is collected as a candidate name since it is ambiguous between a date and a person name. But if no other name is found to contain it as a substring (e.g., "May Smith"), it is safe to assume that the singleton is a date and delete it from the list.

Finally, Nominator groups all variants referring to the same entity in a class, and either chooses one of the variants as the canonical name or constructs a canonical name from components of different variants. This grouping is done together with assigning a type that categorizes the entity referred to by the class, as these two tasks go hand in hand. The result of this step for the sample document is shown below. Each canonical name is followed by its type and then by any additional variants.

District of Columbia Bar >>ORG
D.C. Court of Appeals >>ORG
American Bar Association >>ORG : ABA
Steptoe & Johnson >>ORG
Washington >>PLACE
Dubuque >>PLACE
Robert Jordan >>PERSON : Mr. Jordan

This is discussed in Section 5.

Nominator can provide information about names extracted from documents in a variety of ways. For an information retrieval application, it provides two types of output: a list of terms to be indexed and a dictionary of all names identified in a document collection. Typically, information retrieval systems perform an indexing operation on the text as a backend batch process. They index the occurrences of words that appear in the text into a structure suitable for subsequent searches - usually, an inverted list consisting of a word and the locations of all its occurrences. For indexing names, our module provides the information retrieval indexer with sets containing a variant name, its position (paragraph, sentence and first word) and the canonical name associated with it. The variant is indexed as an occurrence of its canonical name at the variant's position in the text.

In addition, Nominator produces a dictionary of names. It outputs into a file entries containing a canonical form, the entity type it refers to, and all the variants found for it in the document. The dictionary is used at query time to identify mentions of names in the query and map those onto canonical forms. The set of canonical forms matching the query is then searched for by the query engine which returns all occurrences of those canonical names in the document collection. (See Ravin et al. in preparation, for more details.)

The Nominator writes entries into the dictionary after processing each document. At the end of the run, when a whole document collection is processed, the last step in name processing takes place - equivalent classes listed in the dictionary are aggregated across documents, and their variants are merged. This is discussed in Section 6.

In the following sections, the stages in the processing of names are discussed in more detail. Section 3 describes forming the candidate name list; Section 4 describes how names are split into smaller names; Section 5 discusses the forming of equivalence classes.

Section 3: Forming a candidate name list

In well edited texts in English, initial capitalization is a good indicator for names. Since the style conventions of English require that each word of a name start with an uppercase letter, a good candidate list for (multi-word) names can be obtained by collecting all sequences of initially capitalized words appearing in the text. There are a few exceptions to this generalization. On one hand, some initially capitalized words in English are clearly not names -- adjectives denoting provenance, such as "Irish" or "Parisian"; certain acronyms, such as "TV" or "CD"; or dates, such as "January" or "Labor Day." On the other hand, some non-capitalized words are part of names. Personal names may include prefixes ("Antoine de Saint-Exupery") or punctuation ("Thomas (Bud) Bolsky"); organization names often include prepositions ("American Society for the Prevention of Cruelty to Animals") or numerals ("Century 21"); place names may include commas ("Mt. Kisco, New York").

Nominator handles these exceptions by allowing name sequences to include any of a small set of explicitly listed lowercase words as well as numerals under certain conditions. While picking up all numerals in a text can significantly increase the length of the candidate name list, certain configurations of numerals and other elements can be ruled out as forming invalid names. Numerals are concatenated to name sequences only if they do not follow a preposition, a comma, a date, or another number.

Two stylistic conventions in English interfere with identifying names on the basis of initial capitalization: the first is the convention that all the words in titles are initially capitalized, whether they are part of proper names or not. For best performance, Nominator skips the title of the document being analyzed, provided that the title is marked by a recognizable tag. Unmarked titles or titles that appear in the body of the text, such as titles of movies, books and other creative works, are identified as names but remain uncategorized as to type.

The second convention is the initial capital at the beginning of each sentence in English. This requires special treatment of sentence-initial words. In most cases, sentence-initial capitalized words turn out to be non-names: they are typically single words, which occur only at sentence-initial positions. Such non-names are eventually deleted from the final list of names. (See Section 5.) However, when a sentence-initial capitalized word is the first in a sequence of several capitalized words, it is more ambiguous.

The majority of nonvalid sentence-initial sequences of capitalized words contain as their first word an adverb or some other frequent word that does not usually form part of a name, while the rest of the sequence is a valid name, as in "Traditionally, Britain and West Germany" or "Yesterday the New York Stock Exchange". The following heuristic seems to work: If the first word is more than 4 characters, ends in "ly", but is not a person's first name (e.g. "Frequently", but not "Beverly"), it is considered to be an adverb and is removed from the name sequence. A list of about 400 other frequent sentence-initial words is also consulted. Finally, when the name is truly ambiguous, (compare a sentence-initial "New Sears" and "New Coke"), a conservative decision to keep it intact is made.

Text tokens that cannot be part of names, such as most lowercase words and punctuation, typically signal the end (or boundary) of a candidate name sequence. Once the sequence is bounded, further processing of its contents becomes possible. Certain leading or trailing tokens can be removed. Some of this "cleaning" is trivial; for example, trailing blanks, hyphens or opening parentheses are removed. Some is more intricate: a final "'s" or numeral is removed if the word preceding it is a place (e.g., "France 1987") or an all-capital word ("IBM's"); but not otherwise ("Century 21" or "Macy's"), as it might constitute part of the name.

The number of words in the sequence is also indicative. Many words can occur as part of names, but do not constitute names on their own. Thus "Advertising" is discarded, while "Advertising International" is not. (These words appear in the authority list mentioned in the Introduction.)

Similarly, some sequences of (modified) adjectives are discarded as they also can occur as part of names, but are not names on their own, such as "African" or "North American", but not "American Airlines". (Note that "American" on its own is a problem. As it occurs so frequently as an adjective, we discard it, even though it may sometimes refer to a valid named entity, such as "American Airlines".)

Section 4: Splitting sequences into smaller names

The most complex part of the name extraction process is the splitting of one candidate sequence into two or more names. When a sequence contains more than one name, it is in fact a linguistic structure that needs to be parsed. It poses, unfortunately, a subset of the most thorny problems known in natural language parsing.

The first problem is the syntactic ambiguity of conjunctions. Many of the complex name structures are linguistic constructions containing conjunctions ("and" and commas), which are highly ambiguous and which can be properly disambiguated only with access to the semantics of their conjuncts. Consider "Victoria and Albert Museum" versus "IBM and Bell Laboratories". The components of both names are syntactically identical

Proper_Name conj Proper_Name Noun

but their internal structure is not

[Proper_Name conj Proper_Name] Noun

versus

[Proper_Name] conj [Proper_Name Noun]

Another way of expressing the difference is that in the first case the conjunction is within the scope of the noun "Museum", that is, "Museum" has stronger scope than "and". In the second case, the inverse is true: the noun "Laboratories" is within the scope of the conjunction, yielding two different proper names that should be split.

Nominator does not perform a semantic analysis of the names it processes. Even if it did, note that in this case, knowledge of the meaning of words such as "museum" and "laboratories" would not be enough. Human understanding of these names depends on knowledge of the world and the entities that populate it - one museum; two industrial companies. Imparting such knowledge to a computer program is notoriously difficult and labor intensive and is very much beyond the scope of a module like Nominator. Instead, Nominator operates with a set of heuristics to determine the most likely scope of conjunctions within name sequences.

The problem is further complicated by the fact that conjunctive structures are recursive. For example, we could form (and we often find in text) structures like: "The MOMA, the Museum of Natural History and the Victoria and Albert Museum". To unpack such name structures, the splitting in Nominator is a recursive process, examining conjunctions ("and" and commas) from right to left, splitting if necessary, and submitting the split pieces to the process again. When all the conditions for splitting conjunctions from the right are exhausted, we examine conjunctions from the left. In addition, the various heuristics are ordered, as some are more reliable than others and should apply first.

Another well-known natural language parsing problem present in the processing of names is the syntactic ambiguity of prepositional phrases. Many of the complex name structures contain prepositions ("of" or "for") and prepositional phrases ("for the Advancement of Education" or "against Cruelty to Animals").

Based on syntax alone, it is not possible to determine what name or noun is being modified by the prepositional phrase, as these phrases may modify any of the nouns preceding them in a sentence. Consider "Ronald Reagan on Election Day" versus "Council on Foreign Relations". The components of both sequences look very similar but their internal structure is not. In the first, the prepositional phrase "on Election Day" does not modify Reagan, but rather some event described by words occurring earlier in the sentence. The sequence should be split into two names. In the second case, the prepositional phrase describes the council, and is therefore part of the name.

Unlike names with conjunctions, we find that the boundaries of names with prepositional phrases are more difficult to establish, even for humans. For example, is the location of an organization part of its name proper or is it a modifier, which should be separated from its name? "City University of New York", for example, is the official name of that institution, clearly indicating the "New York" is part of its name. However, "Discount Corp. of New York" should perhaps be split, as "New York" simply adds information about the corporation. In our experience, human judgements vary on this issue. Consider "Federal Reserve Bank of New York", "Western Co. of North America", "Commodity Exchange in New York", and so on.

Like conjunctions, prepositional phrases form recursive structures ("Axis Mundi in New York to Ziba Design in Beaverton, Ore.") and need to be unpacked recursively. They also interact with conjunctions. Like conjunctions and prepositional phrases, the "'s", which designates the possessive in English, also forms complex and ambiguous structures (e.g. "Donoghue's Money Fund Report" versus "Britain's University of Sussex"). In fact, the behavior of all three structures is similar enough that we use the same set of splitting heuristics for all.

The heuristics for splitting names in Nominator are very rich in coverage and quite precise in the results they yield. (See Appendix A for the full evaluation.) Currently, Nominator does not address the further need for reconstruction of names. Reconstruction is required in two cases: One is ellipsis, as in "Nancy and Ronald Reagan", where a full treatment of the names requires not only splitting but also restoring the missing parts to form "Nancy Reagan" and "Ronald Reagan". The other is inversion, as in "General Electric Co.'s Information Services", which is a variant of "Information Services of General Electric Co." and should perhaps be recognized as such.

Section 5: Grouping names in equivalent classes

Section 3 discussed how the text of the document, in the form of tokens, is scanned and an initial list of candidate name sequences is collected. Section 4 discussed how each sequence on the list is split into smaller individual names if appropriate. This section describes how names on the list are compared to each other and grouped into equivalent classes.

In a typical document, there will be many mentions of the same entity. Trivially, these will be repetitions of an identical string -- duplications -- and they can be collapsed. (For applications that care about duplicate occurrences, such as indexing in information retrieval, the information is preserved.) In addition to identical duplicates, Nominator identifies two other kinds of duplication: One is the repetition of a sentence-initial capitalized word somewhere else in the document. This is taken as indication that the sentence-initial word is likely to be a name and not capitalized just because it happens to be the first word of a sentence. Thus, if "White" occurs at sentence-initial position and also as a substring of another name (e.g., "Mr. White") or capitalized in the middle of a sentence, it is kept as an occurrence of a name. Otherwise, it is discarded. We will get rid of, for example, the single occurrence of "White" in "White paint is ...". This simple heuristic turns out to perform very well.

The second use of duplication is for deciding when to strip a possessive "'s" off of names. If a name on the list is identical to another, or a proper substring of it, except for the presence of a final "'s", this serves as evidence that the possessive "s" is not a true part of the name and should be removed. Thus, if the document contains both "Ford's" (originating from "Ford's latest model") and "Ford", the "'s" on the former is stripped away. The assumption, which proves right in the overwhelming majority of cases, is that if the "'s" were truly part of the name, (e.g., "Macy's"), there would be no mention of the name without it in the document.

Documents contain repeated mentions of the same entity by different names, or variants. Nominator cycles through the list of names, identifying variant names for the same entity, grouping them together in an equivalence class and removing them from the list. Each variant may contain different pieces of information about the entity named. Typically, one of the variants will contain strong evidence for the type of entity it refers to. A name containing "Mr." or "Professor" indicates that the entity is PERSON; a name containing "Corp." provides strong indication that the entity is ORG. Nominator traverses the list of names, analyzing the internal structure of each one to identify a variant with strong evidence for a type. When such a valid variant is encountered, it becomes the first member of the equivalence class for the entity it refers to, and it determines the entity type of the class. For example, if "Mr. Everest" occurred on the name list, it will be picked up as a valid human name and become the first member of the person "Everest" class.

Nominator then traverses the list of names again to find all the other variants that could belong to the same class. It analyzes each name (if it hasn't already done so) and this time, accepts names with somewhat lower validity score. Nonvalid names, for example, do not establish equivalence classes (at this stage of the process) but they are accepted as variants of ones already established. All variants that contain "Everest", for example, are considered candidate members for the person "Everest" class. (Nominator also checks for possible acronyms, e.g., "JFK" for "John F. Kennedy".) "Everest" by itself is analyzed as a nonvalid human name -- it is neither valid (we cannot be sure it is a person name) nor invalid (but it could refer to one). "Mt. Everest", by contrast, is analyzed as an invalid person name (although it is a valid place name), and is not accepted as a member of the person "Everest" class.

Other valid human names, for example, "Mrs. Everest", are also rejected. As part of the name analysis Nominator performs a consistency check on all the members of a certain class, and rejects an equivalence between "Mr." and "Mrs.", which violates the gender agreement on the entity. (Currently we do not check the consistency of male and female first names, so that "Helen Everest" would be accepted as an equivalent of "Mr. Everest".) After "Robert Everest" was accepted, if "Bob Everest" or "R. Everest" occur, they will be accepted too, but "Alan Everest" will be rejected because of a conflict in the first names.

Finally, the name analysis performed to form equivalence classes also gathers all the name parts (first name, middle name, suffix, etc.) for that class, which are now combined to form the canonical name. For a person, the canonical name consists of all the human name parts, except for the job title (such as "President" or "Prime Minister") as this may change over time. If a first name exists, the personal title ("Mr.") is also removed because it is redundant. But if there is no first name the personal title is retained, as "Mr. Ford", for example, is more informative than "Ford". If two variants contributed compatible but not identical first names (e.g., "Robert" and "Bob"), the longest one is chosen.

Similar processing is done with organization names, place names and miscellaneous other name types and their variants. The canonical name for a company, for example, is the longest string preceding the company-end tag ("International Business Machines", for example).

After all the valid variants and their equivalents have been removed from the name list, names with lower validity scores are allowed to form equivalence classes. Thus, if "Robert Everest" -- a nonvalid human name -- occurs in the document in the absence of any other valid variant (such as "Mr. Everest"), it will be assigned a weaker type - PERSON?. This type is assigned to names like "Beverly Hills", or "Gray Hall", which, in the absence of stronger evidence, are truly ambiguous. These weaker types (PERSON? and PLACE?) prove to be very useful in the fourth and last step - that of aggregating names across documents, as described in Section 6.

Finally, many names remain untyped. A recent run of Nominator on 20 MB of Wall Street Journal documents produced a total of about 42,000 canonical names. 64% of the names were untyped. This is a very large portion, and we are examining ways to reduce the number of untyped names.

Names remain untyped for a variety of reasons:

1. Many names are neither people, places, organizations nor any of the other legal or financial entities we categorize into. These names remain uncategorized and constitute their own canonical form. ("Star Bellied Sneetches" or "George Melloan's Business World").

2. There is a residue that should have been categorized but wasn't, due to omissions in the module logic or, more often, omissions in the list of tags or clue words. In the past, this was a major source of untyped names, but over the past few months, this residue has shrunk considerably.

3. Various non-names are capitalized in the text. We have developed some heuristics to eliminate many from the list -- such as dates, part numbers, etc. but the variety of capitalized non-names is quite large. For example, newspaper articles often refer to other articles by citing their titles. Thus we get names like "Another Market Crash Happened". (In the future, we are considering broadening the scope of our name recognition heuristics to include titles of articles, books and other works of art.) Finally, the first words of leading paragraphs are often capitalized for stylistic reasons. We are looking into the best way to normalize case in these instances.

4. The largest group of untyped names are variants that do not contain enough evidence about their type. If "IBM", for example, occurs in a document without "International Business Machines", Nominator does not type it. We discuss ways in which some of these untyped names are dealt with at the aggregation stage, across documents in Section 6.

5. Finally, many invalid sequences are picked up due to poor document parsing. For example, in tables, many "names" are formed due to concatenation across table columns. A better document analysis tool would be helpful in this context.

The processing of names on the document level concludes with the appending of the equivalent classes formed to the name dictionary. The canonical name appears first, followed by the entity type and then all the other variants.

Section 6: Aggregation of classes across documents

As described in the previous sections, the document is the optimal unit of context for identifying equivalent variants and defining a canonical name to represent them. In the larger context of a collection of documents, variants may be ambiguous. For example, short variants, like "General Mills" can refer to either a person ("General George Mills") or a company ("General Mills Inc."); acronym variants, like "IRA", to either an organization ("Irish Republican Army") or a financial entity ("Individual Retirement Account"). It is clear, however, that the same entities are referred to in many documents, sometimes by different variants, and that there is need for aggregation of all the different mentions within the same domain.

Nominator uses the smaller context of the document to disambiguate the occurrence of variants, by associating them with their proper canonical name. Once this association has taken place, equivalence classes can be safely merged (or aggregated) across documents if their canonical names are identical. The exact form of the canonical name is very important: it should be explicit enough to distinguish between different named entities, yet normalized enough to aggregate all mentions of the same entity across many documents.

Canonical forms of human names are comprised of the following name parts, if found: first name, middle name, last name, and suffix. Professional or personal titles and nick names are not included to allow for maximum aggregation. In our collection of documents from the Wall Street Journal, the variant "Federal Reserve Chairman Alan Greenspan" was associated with the canonical name: "Alan Greenspan" in one document. In other documents of the 20 MB of Wall Street Journal text, the following variants were also associated with it: "Mr. Greenspan", "Greenspan", "Federal Reserve Board Chairman Alan Greenspan", "Fed Chairman Alan Greenspan" and "Chairman Alan Greenspan". Because they were all associated with the identical canonical form, they were aggregated into one entry in the name dictionary.

The canonical names of companies do not include the tags usually ending company names, as these may vary among different mentions. Both "Allegheny International Inc." and "Allegheny International Corp." are variants of "Allegheny International" and so they aggregate. Finally, we aggregate canonical names that are near identical -- names that differ in the presence or absence of hyphens ("Rolls-Royce" vs. "Rolls Royce"), blanks and other minor ways.

The aggregator (implemented by R. Byrd at Watson) can take advantage of the larger context of the document collection and its accrued evidence to make certain decisions not possible in the earlier stages. In particular, it can compare the entity type of two identical canonical names and if one is higher on the type hierarchy than the other, the process assigns the higher type to the aggregated entry. The type hierarchy is as follows:

ORG, PERSON, PLACE
PERSON?
PLACE?
OTHER
UNCAT

Thus the PERSON? "Mikhail Gorbachev" of one document aggregates with the PERSON "Mikhail Gorbachev" of another, whose variant "Mr. Gorbachev" was responsible for the higher type. This aggregation of weaker types is particularly helpful in reducing the number of untyped entries. The untyped "Houghton-Mifflin" of one document is aggregated with the ORG "Houghton-Mifflin" of another, whose higher type was contributed by its variant "Houghton-Mifflin Co."

To disallow erroneous aggregation of references to different entities, we do not aggregate over identical canonical names with equally strong entity types - such as "Boston" the company ("Boston Inc.") and "Boston" the place.

We do not aggregate over different canonical names. Thus, we keep the canonical place "New York" (the city or the state) distinct from the canonical "New York City" on one hand and "New York State" on the other. Similarly, with human names: "Jerry O. Williams" in one document is separate from the canonical "Jerry Williams" in another; or, more significantly, "Jerry Lewis" from one document is kept distinct from "Jerry Lee Lewis" from another. Although it is obvious to humans that "George Shultz" and "George P. Shultz" are probably references to the same person, Nominator has no access to the necessary knowledge of the world to make this decision. We are conservative with company names too, preferring to keep the canonical name "Allegheny International" and its variants separate from the canonical name "Allegheny Ludlum" and its variant, "Allegheny Ludlum Corp."

Even with such conservative criteria, aggregation over documents is quite drastic. The pre-aggregation name dictionary for 20MB of WSJ text contains 120,257 names. After aggregation, the name dictionary contains 42,033 names, or about a third of its non-aggregated size.

Section 7: Future work

Nominator relies on heuristics which by their ad-hoc nature do not achieve full recall and precision. However, improving their accuracy is an ongoing process - various mis-analyses are reported by users and other applications, and we continue to refine the heuristics and improve their coverage. In addition to this on-going effort, we are working on a few enhancements to Nominator.

Performance: We have reached near-optimal performance, given the algorithms Nominator currently uses. The most intensive part of the execution is the various pattern matching routines that are used to analyze the content of strings and substrings. To significantly speed up the performance of Nominator, we are studying the use of transducers, finite-state machines that promise to perform pattern matching much more efficiently. We have started a dialogue with the Computational Linguistics project at ECAM, the Paris Scientific Center of IBM. (For a discussion of the use of transducers in computational linguistics see Manaster-Ramer 1993).

Quality: As mentioned in Section 5, the proportion of untyped names is too high. We are studying ways in which a post process would consult the name dictionary produced for a collection to re-analyze untyped names and identify valid name substrings that they contain. In particular, the current processing of names does not make use of statistical information. We are considering keeping track of the frequency of occurrence of various untyped names, in order to discard those that occur very rarely. In addition, untyped names that are ambiguous variants of more common canonical names (e.g., "Ford") could be aggregated with one rather than another canonical name, depending on the statistical properties of the canonical names.

Acknowledgements

Misook A. Choi turned an initial slow and fragile prototype of Nominator into a production-level module, by essentially rewriting most of the code. The satisfaction of our customers speaks for the quality of her work.

Roy J. Byrd designed and implemented several modules used by Nominator - specifically, the sentence tokenizer, the abbreviation matching routine and the aggregator.

Herbert A. Chong assisted in speeding up the performance of Nominator.

Aaron Kershenbaum contributed the dictionary access mechanism.

Mary S. Neff, John M. Prager, and Alan D. Marwick provided feedback, insights and support.

References

Byrd, R.J., Y. Ravin and J.M. Prager. "Lexical Assistance at the Information Retrieval User Interface," *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas Nevada, April 1995.

Borgman, C.L. and S.L. Siegfried. "Getty's Synoname and its cousins: a survey of applications of personal name-matching algorithms," *Journal of the American Society for Information Science*, Vol. 43, No. 7, 459-476, 1992.

Coates-Stephens, S. "The analysis and acquisition of proper names for the understanding of free text," *Computers and the Humanities*, Vol.26, 441-456, 1993.

De Silva G.L. and Hull J.J. "Proper Noun Detection in Document Images," *Pattern Recognition*, Vol. 27, 2, 311-320, 1994.

Hayes, P. "NameFinder: Software that finds Names in Text," in *Proceedings of RIAO 94* 762-774, New York, October 1994.

Manaster-Ramer A. "Towards Transductive Linguistics," in K. Jensen, G.E. Heidorn and S.D. Richardson, eds., *Natural Language Processing: the PLNLP Approach*, 13-28, Kluwer Academic Publishers, Boston, 1993.

Mani, I., T.R. Macmillan, S. Luperfoy, E.P. Lusher, and S.J. Laskowski. "Identifying unknown proper names in newswire text," in B. Boguraev and J. Pustejovsky, eds., *Acquisition of Lexical Knowledge from Text: Proceedings of a Workshop Sponsored by the Special Interest Group on the Lexicon of the Association for Computational Linguistics*, 44-54, Columbus, Ohio, 1993.

McDonald, D.D. "Internal and external evidence in the identification and semantic categorization of proper names," in B. Boguraev and J. Pustejovsky, eds., *Acquisition of Lexical Knowledge from Text: Proceedings of a Workshop Sponsored by the Special Interest Group on the Lexicon of the Association for Computational Linguistics*, 32-43, Columbus, Ohio, 1993.

Myaeng, S.H. and E.D. Liddy. "Information Retrieval with Semantics Representation of Texts," *Proceedings of the 2nd Annual Symposium on Document Analysis and IR*, 201-215, Las Vegas, 1993.

NIST, *TIPSTER Information-Retrieval Text Research Collection*, on CD-ROM, published by *The National Institute of Standards and Technology*, Gaithersburg, Maryland, 1993.

Paik, W., E.D. Liddy, E. Yu, and M. McKenna. "Categorizing and standardizing proper nouns for efficient information retrieval," in B. Boguraev and J. Pustejovsky, eds., *Acquisition of Lexical Knowledge from Text: Proceedings of a Workshop Sponsored by the Special Interest Group on the Lexicon of the Association for Computational Linguistics*, 154-160, Columbus, Ohio, 1993.

Ravin Y., A. Kershenbaum, and R.J. Byrd. "Query Expansion with Enriched Vocabulary" (in preparation).

Wacholder N., Y. Ravin and R.J. Byrd. "Retrieving Information from Full Text Using Linguistic Knowledge," *Proceedings of the Fifteenth National Online Meeting*, New York, May 1994.

(No author). "SRA International unveils "NameTag"; Indexing engine is tuned for corporate on-line search & retrieval product designed to 'deliver proper names ... properly'" *Business Wire*, October 24, 1995.

Appendix A: Evaluation of Nominator

The following is based on a report written by Nina Wacholder, a linguist who has been working with our group. The report was written in October of 1994, and many of the errors listed here have since been removed, as we have continued to work on improving the quality of the module's output.

The process of evaluation

The evaluation was conducted on the first 88 documents in the Wall Street Journal on CD-ROM (1988), a portion of text that was not used during initial development. 2426 proper names were identified manually in the evaluation corpus; automatic sorting and removal of duplicates (by document) reduced this manually built list to 1354 proper names. Nominator processed the same corpus, generating 1409 proper names after sorting and removal of duplicates. The list of names generated by Nominator was then compared manually with the hand-built list. Nominator was judged to have found a valid name only if it found a form identical to the one identified manually or differing only with respect to initial "the" (e.g., "Rockefeller University" and "the Rockefeller University" were treated as identical).

Recall

Recall is a measure of coverage -- what portion of the total names was identified by the module? More precisely, recall is the ratio of the number of names found by both linguist and the module over the total number of names identified by the linguist.

RECALL: 1230/1354 or 90.8%

In practice, however, the recall rate is higher than 90.8%. Many proper names are retrievable even though the name that the module found is not identical to the name found manually. For example, addition of an extra 's to a proper name or failure to split a complex proper name does not preclude later applications from recovering the right proper name. By this more practical standard, an additional 95 names are retrievable, producing an actual recall rate of 97.8%.

ACTUAL RECALL: (1230 + 95) / 1354 = 97.8%

Precision

Precision is a measure of accuracy -- what portion of the names extracted by the module were valid names? Precision is the ratio of the number of names identified by both the module and the linguist over the total number of names extracted by the module.

PRECISION: 1230/1409 or 87.2%

Nominator relies heavily on conventions of capitalization. When the text does not follow these conventions, the error rate increases. A significant number of errors made by Nominator were due to poor formatting of the text, including incorrect marking of sentence breaks and tables not so identified. 69 errors made by Nominator can be attributed to formatting problems. When these erroneously identified names are subtracted from the total number of names identified automatically, the precision rate increases to 91.8%.

ACTUAL PRECISION: 1230/(1409 - 69) = 91.8%

Loss

This is our own measure for the number of names that are not retrievable by later processes because of errors produce by Nominator. Loss is a ratio of the number of "lost" valid names out of the total number of valid names that were found by the linguist in the corpus.

LOSS: 38/1354 = 2.8%

Note: LOSS plus ACTUAL RECALL add up to over 100% because errors were counted by type, and so, occasionally, two types of errors were engendered by a single proper name or conjoined sequence of proper names. For example, the string "Bob's Ski Shop in Portland, Oregon" occurred at the beginning of the sentence. But "Bob's Ski Shop" actually engendered two errors: the name "Bob's" was lost and "Ski Shop" was attached to "in Portland" (as "Ski Shop in Portland"), thereby counting as a bad combination. Although this method of evaluation inflates the error rate slightly, it was chosen because it produces an accurate picture of the types of errors made by Nominator.

Main sources of errors

There were 69 errors due to text formatting, such as faulty sentence breaks, or text that was all capitalized at the beginning of paragraphs. There were 24 errors due to wrong splitting of proper names with conjunctions; and 42 errors due to wrong splitting of other complex names. Finally, 16 errors were due to the ambiguity of sentence-initial capitalization.

False hits

False hits refers to the percentage of non-names wrongly identified by the module as proper names, out of the total names extracted by the module.

FALSE HITS: 38/1409 = 2.6%

Classification

Of the 1230 names correctly identified by the module, 263 or 21.4% were untyped (marked as UNCAT). See the discussion in Section 5 for the various reasons of why the module fails to type names.

In addition, there were 15 wrong type assignments, yielding a 1.2% error in classification.

CLASSIFICATION ERRORS: 15/1230 or 1.2%

Examples of wrong classification included: "Brigham Young" (the university), "Mr. Soda" (a division of a beverage company) and "Freddie Mac" (the loan) all classified as people.

The inverse of the classification errors is the classification precision, which is very high.

CLASSIFICATION PRECISION: (1230-15)/ 1230 = 98.8%

Finally, a few words about comparing these results to the results of another similar study, reported in Paik et al. (1993). Paik et al. report that their proper name processor, which includes a probabilistic part of speech tagger and a proper noun phrase bracketter, has a "success ratio" of about 95%. They do not discuss in detail what criteria they use to determine success so exact comparisons can't be made, but Nominator's actual recall rate of 97.8% and actual precision rate of 91.8% appear to be roughly comparable.

The precision rate for their proper noun categorizer, which breaks proper nouns down into 19 categories (city, province, country, company, date, religion, nationality,... miscellaneous) was 74% for 588 proper names. Nominator's precision rate is considerably higher but this difference is due in part to the fact that Nominator only has four categories. The comparison with Paik et al. is also obscured because Nominator discards certain proper names such as dates and nationalities (e.g. American, Japanese) which Paik et al. classify.

Appendix B: The features used by Nominator

The name extraction module consults authority files - lists of words with particular features. The presence of one of these words in a name is used as evidence for its type - human name, place name, or name of an organization. These special words also help determine the boundaries of names, or the internal parts of names.

Special words related to human names

Some limited number of words may end human names. These are ordinals (eg "1st"), Roman numerals, and the words "Junior" and "Senior". They are "weak" indicators of human names because they can also appear as part of non-human names ("Martin Luther King Junior High" or "Beverly Hill Cops II"). Others, such as "Esq." or "Jr.", are strong indicators of the end of human names.

Similarly, a limited number of words may begin human names, such as "Baron", "Lord" or "Sergeant". These are weak indicators because they are ambiguous. "Lord" or "Baron" for example could be last names too. Strong indicators of the beginning of human names are personal titles, such as "Dame" or "Dr." as well as professional titles, such as "Capt." or "Prof.".

Many indicators of human names are common (but capitalized) nouns that can take a prepositional phrase, usually an "of" phrase. "Professor" can immediately precede the name ("Professor Katz"), or it can be followed by a prepositional phrase, forming a professional title and then optionally followed by a human name, e.g., "Professor of Latin American Studies John Katz" .

Some of these human name indicators convey gender information. For example, "Sr." is male while "Queen" is female. Clashes of gender (if known) is used to disallow two otherwise compatible variants from sharing the same equivalence group.

Human names may have one or more prefixes that precede the last name, such as "de" or "la" (eg "Antoine de St. Exupery"). These form a small group of lower case and upper case prefixes.

Finally, royal human names have a particular structure - they don't have last names, for example, but include indicators of royal names such as "Prince" or "King".

In the absence of other indicators, the presence of a first name as the first word of a name is an indicator of a possible human name.

Special words related to place names

There are a few hundred names of well-known places: continents, countries, states of the United States, important cities, and other locations, such as the Alps, or the Gaza Strip.

As with human names, there are a few words that indicate the beginning of a place name, such as "Cape", "Fort" or "Lake". There are words that indicate the end of a place name, such as "Bridge" or "River". There are words that could indicate either, e.g.. "Mount Olympus" or "Green Mount".

As with human names, there are also words that can take a prepositional phrase, which is part of the name as in "City of New York".

A few words that indicate a place name are, unfortunately, also very common as first or last names. Whether they end up signaling a place name or not depends on the other names that appear in the context. In addition to being marked as place indicators, words like "Bay" or "Fort" are also possible first names, and words like "Park" or "Hall" are also last names.

Special words related to organization names

There are words that are very strong indicators of the end of an organization name, usually a company name, such as "Inc." or "Co.". If anything follows these indicators, it is usually more endtags or the company's location.

A second group of words indicate a company name, and may end it, but may also occur before the very end of the name, such as "Associates" or "Systems".

A third group indicate a company name but can occur anywhere in it, such as "Broadcasting" or "Club".

As with names of people and places, organization names can also contain common nouns that take prepositional phrases, and these phrases may be contained in the name as well, e.g., "The Museum of Broadcasting" or "The Association of Musicians of Greater New York".

Special words related to other entities

A small group of words are indicators of financial entities ("Index" in "Dow Jones Index"), literary entities ("Anthology" in "The Oxford Anthology"), and others.

Many words designate business or academic areas, or areas of responsibility of governmental agencies or departments. It is not possible to list them all, but the most common are listed here. They are used to identify the end of a professional title and the beginning of the name proper, as in "Professor of Urban Studies Lewis Johnson".

Words designating funding organizations are important because the names of funds may include one or more human names and should not be split into sub-names, as in "Mary and Paul Smith Foundation".

Currently the name-extraction module does not identify product names; however product indicators are used as negative evidence for human or place names.

Other special words

Words have properties which determine whether they can be part of names or not. Some words are capitalized but are not names, such as "TV" or "VCR"; others are capitalized adjectives ("British") or nouns ("Americans") that are also not names. The plural form of these adjectives, though, can appear as part of names. ("The Association for Elderly Americans" but not "The Association for Elderly American").

Function words (e.g., "of" or "for") can concatenate to names, but cannot begin or end names.

Some words are ambiguous between dates (or some other non-name) and valid names, such as "April" or "March". These are specially marked and only discarded later, based on the context in which they are found.

Other words are valid only as part of names, but can't constitute names on their own (e.g., "Associates").

Some words are very common at the beginning of sentences, but are very unlikely to be part of a name, even if they are followed by capitalized words, as in "Among IBM...". They cannot constitute names on their own.

Roman numerals are listed as they can be part of some names.

Finally, some modifiers are sometimes part of valid names, as is "North" in "North American Federation Trade Agreement", but not when they modify adjectives. ("North Americans" is not a name.)

Appendix C: Brief survey of the literature

Commercial Products:

NameFinder (Hayes 1994)

The Carnegie Group has developed a name finding program, which it makes available to text processing and information retrieval applications as an API.

Like Nominator, NameFinder finds names and variants of geographical areas, companies and people. Unlike Nominator, it requires a pre-existing database of names, where variants are explicitly listed with their canonical names, to be supplied by the customer. Carnegie Group is currently working on extending NameFinder to identify company names that do not pre-exist in the database. The success rate reported in the paper for this mode of operation was about 80%.

Because NameFinder uses a pre-existing database of names, it allows users to set some global controls. For example, users can specify that names be identified with or without capitalization; company names be identified with or without name-ending tags, etc.

The author cites over 99% of both precision and recall achieved by some applications built on top of NameFinder (that is, with full use of a pre-existing name database).

NameTag (Business Wire, October 24, 1995)

According to this news report, SRA's NameTag is very similar to IBM's Nominator, described in this technical report. Like Nominator, it finds, indexes and interprets proper names of people, places, and organizations. Like Nominator, it uses "computational linguistics and pattern-matching methods in a heuristic approach to determine what is a name, what sort of name it is, and how to categorize it." It finds variants of names and links them.

SRA has entered into partnerships with various other vendors: publishers and on-line information providers, such as Encyclopedia Britannica and Infonautics and information retrieval systems, such as Excalibur and Verity.

Academic systems:

Unlike the commercial products, which are APIs that can be integrated into different text processing applications, most academic name processing modules are an integral part of more comprehensive and intensive text understanding systems, making the comparison with the commercial products quite difficult.

Analysis and Acquisition of Proper Names (Coates-Stephens 1993)

Coates-Stephens' doctoral dissertation and subsequent publications represent seminal work and an important point of reference in the area of name processing. They constitute one of the most comprehensive treatment of all the aspects of name analysis in the context of full natural language parsing.

In his work, as in the later work by McDonald (discussed below) name identification is intimately integrated with a parser. For example, Coates-Stephens performs morphological analysis on the components of names to analyze "Chinese" as derived from "China" and therefore labels it as an adjective indicating origin. In addition, he analyzes the internal syntactic structure of names (e.g., [[Patriotic]adj [Salvation]n [Movement]n]np) and is interested in their semantic analysis. He differentiates between names whose components preserve their individual meaning, as in "Child Poverty Action Group", and names whose components' meanings no longer apply as is the case with "Metropolitan" in "Metropolitan Opera".

This comprehensive analysis of names as part of a general grammar of natural language allows for very high accuracy. For example, it allows the correct analysis of "the Missouri battleship" as a ship, ignoring the fact that "Missouri" is usually a place name; or of "an Olivetti computer", in which the company name is used only as an adjective.

Such careful analysis is not possible without the full analysis of the context, which Nominator does not do.

Coates-Stephens checked the recall and precision of his system on two samples of 100 stories of British newspaper of the same kind that was used for the development of the system. Out of a total of 6250 words, 685 were names; 595 were unique names; 450 were unknown to the system's dictionary. Of those, in sentences that parsed, the "success rate" was 79%. ("Success rate" probably means fully accurate syntactic and semantic analysis.) It was somewhat lower - 70% - in a sample of Wall Street Journal text.

<u>Internal and external evidence in the identification and semantic categorization of proper names</u> (McDonald 1993)

The paper stresses the importance of local context for accurate name analysis. In particular, information from the syntax and semantics of the sentence in which a name occurs may be necessary. If, for example, a pronoun like "who" or a person's age follows a name (e.g., "Michael Johnson, who ..."; "Bill Miller, 57, was nominated ..."), it is good indication that the name refers to a person.

The author describes his name identification system, which is part of a more general parsing grammar with context-sensitive rules: the SPARSER natural-language understanding system, developed at Brandeis University. It goes beyond the work of Coates-Stephens in that it employs context-sensitive rewrite rules for name analysis. In addition, the system builds and maintains a world-model for the text being analyzed. The model contains all the entities mentioned in the text and either maps new names onto them or instantiates new entities if needed.

Recall and precision are both very high - close to 100% on the "Who's News" articles in the Wall Street Journal material.

Execution time is not discussed in the paper. However, it is known that performance in such rule-intensive systems tends to be slow.

<u>Identifying unknown proper names in text</u> (Mani et al. 1993)

The paper stresses the need to identify unknown names, that is, names that do not pre-exist in a database, which has been a guiding principle for the design of Nominator as well.

As for the name analysis described by the authors -- as it is very dependent on the rest of their system and so interrelated with other discourse analysis issues, that it is hard to evaluate it on its own.

<u>Proper nouns for information retrieval</u> (Paik et al, 1993) and (Myaeng & Liddy 1993)

The name identification and analysis employed by the authors is very similar to ours. The only differences are that Nominator does not make use of such extensive gazetteers; on the other hand, it does not do such a fine categorization of names, into categories such as "US Government" or "developing country". These categories are needed for the authors' information retrieval application, in which proper name matching is used as a filter to discard documents irrelevant to a given query.

The authors report recall and precision in the low 90%, measured over 589 names, in 25 Wall Street Journal documents.

A survey of applications of personal name-matching algorithms (Borgman at al 1992)

This paper is a survey of the different needs that different applications have for name identification. The Getty Museum, for example, where the authors work, is primarily interested in identifying different spellings of the same artist's name. It needs to address problems of variations in name spelling. The authors discuss different algorithms for spell-checking of names.

This aspect of name analysis exceeds the scope of Nominator.

Appendix D: Sample output

The following list contains three unedited excerpts of the aggregated name dictionary created by processing 20 MB of Wall Street Journal text (NIST 1993). The canonical form is the first item on the line, followed by the entity type (marked with ">>") and the variants. Variants are separated by "@". An "*" marks a canonical name that has been "demoted" by the aggregator to be a variant of another "stronger" canonical name.

Alex Grass >>PERSON : Mr. Grass @ Grass
Alex Henderson >>PERSON : Mr. Henderson
Alex Katz >>PERSON? :
Alex Kotlowitz >>PERSON? :
Alex M. Cena >>PERSON? :
Alex Magno >>PERSON? :
Alex Morrow >>PERSON? :
Alex Musical Instruments >>ORG : Alex Musical Instruments Inc. in New York
Alex Spanos >>PERSON : Mr. Spanos
Alex von Bidder >>PERSON : Mr. von Bidder
Alex W. Hart >>PERSON : Mr. Hart @ Alex W. " Pete " Hart
Alexander >>UNCAT :
Alexanders >>UNCAT :
Alexandria >>PLACE :
Alexander >>ORG : Alexander 's Inc. , New York @ Alexander 's Inc.
@ Alexander 's
Alexander & Alexander Services >>ORG : Alexander & Alexander Services Inc.
@ Alexander & Alexander
Alexander & Baldwin >>ORG : Alexander & Baldwin Inc.
Alexandra Biryukova >>PERSON : Mrs. Biryukova
Alexander d'Arbeloff >>PERSON : Mr. d'Arbeloff
Alexander Dubcek >>PERSON? : Dubcek
Alexander Dunaev >>PERSON? :
Alexander the Great >>UNCAT : Alexander the Great 's *
Alexander H. Williams >>PERSON? :
Alexander Nanev >>PERSON? : Nanev
Alexandria in the Nile >>UNCAT :
Alexander Paris >>PERSON : Mr. Paris
Alexander Proudfoot >>ORG : Proudfoot @ Chicagoan Alexander Proudfoot
@ Alexander Proudfoot PLC
Alexander Vik >>PERSON : Mr. Vik
Alexandria , Virginia >>PLACE : Alexandria , Va. @ Alexandria
Alexander Vlasov >>PERSON : Mr. Vlasov @ Interior Minister Alexander Vlasov
Alexander Yakovlev >>PERSON : Mr. Yakovlev
Alexis Arguello >>PERSON? :

American Broadcasting >>ORG : American Broadcasting Cos. @ ABC
Americans for Bush >>UNCAT :
American Business >>UNCAT :
American Can >>ORG : American Can Corp.
American Cancer Society >>ORG :
American Capital >>ORG : Group Seeks Control of American Capital

@ American Capital Corp.
American Capital Corporate Bond Fund >>ORG :
American Capital Group >>ORG : American Capital Group Inc. , Clearwater , Fla.
@ American Capital
American Capital Pace >>UNCAT :
American Carriers >>ORG : American Carriers Inc. , Overland Park , Kan.
American Catholicism >>UNCAT :
American Cellophane League >>ORG :
American Century >>ORG : American Century Corp. , the San Antonio , Texas
American Chemical Society >>ORG :
American City Business Journals >>ORG : American City Business Journals Inc.
@ American City
American Civil Liberties Union >>ORG : ACLU
American Civil Liberties Union 's National Prison Project >>ORG :
American Civil Liberties Union
American College of Gastroenterology >>ORG :
American College of Obstetricians and Gynecologists >>ORG : American
College of Obstetrics and Gynecology *
American College of Surgeons >>ORG :
American Commonwealth >>UNCAT :
American Community Services >>ORG : American Community Services Inc.
@ American Community
American Conference 's Eastern Division >>ORG :
American Continental >>ORG : American Continental Corp.
American Corporate Counsel Association >>ORG :
American Council >>ORG :
American Council for Capital Formation >>ORG :
American Council on Education >>ORG :
American Credit Indemnity >>ORG : American Credit Indemnity Co. ,
Baltimore @ American Credit
American Cyanamid >>ORG : American Cyanamid Co. 's Lederle Laboratories
@ American Cyanamid Co.
American Deep Space Network >>ORG :
American and Delta >>UNCAT :
American Dental Association >>ORG : ADA
American Depository Receipts >>UNCAT : American Depositary Receipts *
American Diversified Real Estate >>ORG : American Diversified Real
Estate Inc. in Wichita , Kan.
American Dream >>UNCAT :
American Eagle >>UNCAT :
American East >>ORG : American East Inc.
American Ecology >>ORG : American Ecology Corp. of Agora Hills , Calif.
@ American Ecology Corp.
American Education Council >>ORG :

B&H Maritime Carriers >>ORG : B&H Maritime Carriers Ltd.
B'nai B'rith >>UNCAT :
B'sof U'machol >>UNCAT :
B. Bader >>UNCAT :
B. Dalton/Barnes & Noble >>UNCAT :
B. J. Cooper >>PERSON : Mr. Cooper @ B.J. Cooper
B. Jack Barnes >>PERSON : Mr. Barnes
B. Lance Sauerteig >>PERSON : Mr. Sauerteig
B. Lawrence Riggs >>PERSON : Dr. Riggs @ Dr. B. Lawrence Riggs
B. N. Pothitos >>PERSON : Mr. Pothitos @ B.N. Pothitos
B.A. >>UNCAT :

B.A.T. >>UNCAT :
B.A.T. Industries >>ORG : B.A.T. Industries PLC @ B.A.T. @ B.A.T Industries PLC @ B.A.T Industries * @ B.A.T
B.B. Real Estate Investment >>ORG : B.B. Real Estate Investment Corp. , Sacramento , Calif. @ B.B. Real Estate Investment Corp.
B.C. >>UNCAT :
B.C. , Cicero >>UNCAT :
B.C. fourth Century A.D. >>UNCAT : B.C.-fourth Century A.D.
B.C. Gas >>ORG : B.C. Gas Inc.
B.C. Pacific Capital >>ORG : B.C. Pacific Capital Corp.
B.F. Goodrich >>ORG : Goodrich @ B.F. Goodrich Co. , Akron , Ohio @ B.F. Goodrich Co.
B.F.Goodrich >>ORG : B.F.Goodrich Co.
B.H. Macomber >>ORG : B.H. Macomber Co.
B.J.F. >>UNCAT :
B.L. Ochman >>UNCAT :
B.R. Inman >>UNCAT :

DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 234: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 398: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 485: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 661: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 776: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 903: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1112: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1223: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1268: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1292: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1380: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1389: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1409: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1431: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1451: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1474: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1485: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1493: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1541: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1551: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1594: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1620: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1643: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1665: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1704: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1902: .H2
DSMBEG323I STARTING PASS 2 OF 2.
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 234: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 398: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 485: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 661: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 776: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 903: .H2

DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1112: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1223: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1268: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1292: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1380: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1389: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1409: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1431: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1451: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1474: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1485: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1493: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1541: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1551: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1594: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1620: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1643: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1665: .H3
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1704: .H2
DSMKH0505E CONTROL WORD PARAMETER MISSING.
DSMMOM395I 'RAVIN' LINE 1902: .H2