Disambiguation of Proper Names in Text

Nina Wacholder`
CRIA
Columbia University
New York, NY 10027
nina@cs.columbia.edu

Yael Ravin
TJ Watson Research Center
IBM
Yorktown Heights, NY 10598
yael@watson.ibm.com

Misook Choi
TJ Watson Research Center
IBM
Yorktown Heights, NY 10598
machoi@watson.ibm.com

Abstract

Identifying the occurrences of proper names in text and the entities they refer to can be a difficult task because of the many-to-many mapping between names and their referents. We analyze the types of ambiguity — structural and semantic — that make the discovery of proper names difficult in text, and describe the heuristics used to disambiguate names in Nominator, a fully-implemented module for proper name recognition developed at the IBM T.J. Watson Research Center.

1 Proper Name Identification in Natural Language Processing

Text processing applications, such as machine translation systems, information retrieval systems or natural-language understanding systems, need to identify multi-word expressions that refer to proper names of people, organizations, places, laws and other entities. When encountering *Mrs. Candy Hill* in input text, for example, a machine translation system should not attempt to look up the translation of *candy* and *hill*, but should translate *Mrs.* to the appropriate personal title in the target language and preserve the rest of the name intact. Similarly, an information retrieval system should not attempt to expand *Candy* to all of its morphological variants or suggest synonyms (Wacholder et al. 1994).

The need to identify proper names has two aspects: the recognition of known names and the discovery of new names. Since obtaining and maintaining a name database requires significant effort, many applications need to operate in the absence of such a resource. Without a database, names need to be discovered in the text and linked to entities they refer to. Even where name databases exist, text needs to be scanned for new names that are formed when entities, such as countries or commercial companies, are created, or for unknown names which become important when the entities they refer to become

topical. This situation is the norm for dynamic applications such as news providing services or Internet information indexing.

The next Section describes the different types of proper name ambiguities we have observed. Section 3 discusses the role of context and world knowledge in their disambiguation; Section 4 describes the process of name discovery as implemented in Nominator, a module for proper name recognition developed at the IBM T.J. Watson Research Center. Sections 5-7 elaborate on Nominator's disambiguation heuristics.

2 The Ambiguity of Proper Names

Name identification requires resolution of a subset of the types of structural and semantic ambiguities encountered in the analysis of nouns and noun phrases (NPs) in natural language processing. Like common nouns, ((Jensen and Binot 1987), (Hindle and Rooth 1993) and (Brill and Resnick 1994)), proper names exhibit structural ambiguity in prepositional phrase (PP) attachment and in conjunction scope.

A PP may be attached to the preceding NP and form part of a single large name, as in NP[Midwest Center PP[for NP[Computer Research]]]. Alternatively it may be independent of the preceding NP, as in NP[Carnegie Hall] PP[for NP[Irwin Berlin]], where *for* separates two distinct names, *Carnegie Hall* and *Irwin Berlin*.

As with PP-attachment of common noun phrases, the ambiguity is not always resolved, even in human sentence parsing (cf. the famous example *I saw the girl in the park with the telescope*). The location of an organization, for instance, could be part of its name (*City University of New York*) or an attached modifier (*The Museum of Modern Art in New York City*). Without knowledge of the official name, it is sometimes difficult to determine the exact boundaries of a proper name. Consider examples such as *Western Co. of North America, Commodity Exchange in New York* and *Hebrew University in Jerusalem, Israel.*

Proper names contain ambiguous conjoined

phrases. The components of *Victoria and Albert Museum* and *IBM and Bell Laboratories* look identical; however, *and* is part of the name of the museum in the first example, but a conjunction joining two computer company names in the second. Although this problem is well known, a search of the computational literature shows that few solutions have been proposed, perhaps because the conjunct ambiguity problem is harder than PP attachment (though see (Agarwal and Boggess 1992) for a method of conjunct identification that relies on syntactic category and semantic label).

Similar structural ambiguity exists with respect to the possessive pronoun, which may indicate a relationship between two names (e.g., *Israel's Shimon Peres*) or may constitute a component of a single name (e.g., *Donoghue's Money Fund Report*).

The resolution of structural ambiguity such as PP attachment and conjunction scope is required in order to automatically establish the exact boundaries of proper names. Once these boundaries have been established, there is another type of well-known structural ambiguity, involving the internal structure of the proper name. For example, *Professor of Far Eastern Art John Blake* is parsed as [[Professor [of Far Eastern Art]] John Blake] whereas *Professor Art Klein* is [[Professor] Art Klein].

Proper names also display semantic ambiguity. Identification of the type of proper nouns resembles the problem of sense disambiguation for common nouns where, for instance, *state* taken out of context may refer either to a government body or the condition of a person or entity. A name variant taken out of context may be one of many types, e.g., *Ford* by itself could be a person (Gerald Ford), an organization (Ford Motors), a make of car (Ford), or a place (Ford, Michigan). Entity-type ambiguity is quite common, as places are named after famous people and companies are named after their owners or locations. In addition, naming conventions are sometimes disregarded by people who enjoy creating novel and unconventional names. A store named *Mr. Tall* and a woman named *April Wednesday* (McDonald 1993) come to mind.

Like common nouns, proper nouns exhibit systematic metonymy: *United States* refers either to a geographical area or to the political body which governs this area; *Wall Street Journal* refers to the printed object, its content, and the commercial entity that produces it.

In addition, proper names resemble definite noun phrases in that their intended referent may be ambiguous. *The man* may refer to more than one male individual previously mentioned in the discourse or present in the non-linguistic context; *J. Smith* may similarly refer to more than one individual named Joseph Smith, John Smith, Jane Smith, etc. Semantic ambiguity of names is very common because of the standard practice of using shorter names to

stand for longer ones. Shared knowledge and context are crucial disambiguation factors. *Paris*, usually refers to the capital of France, rather than a city in Texas or the Trojan prince, but in a particular context, such as a discussion of Greek mythology, the presumed referent changes.

Beyond the ambiguities that proper names share with common nouns, some ambiguities are particular to names: noun phrases may be ambiguous between a name reading and a common noun phrase, as in *Candy*, the person's name, versus *candy* the food, or *The House* as an organization versus a *house* referring to a building. In English, capitalization usually disambiguates the two, though not at sentence beginnings: at the beginning of a sentence, the components and capitalization patterns of *New Coke* and *New Sears* are identical; only world knowledge informs us that *New Coke* is a product and *Sears* is a company.

Furthermore, capitalization does not always disambiguate names from non-names because what constitutes a name as opposed to a non-name is not always clear. According to (Quirk et al. 1972) names, which consist of proper nouns (classified into personal names like *Shakespeare*, temporal names like *Monday*, or geographical names like *Australia*) have 'unique' reference. Proper nouns differ in their linguistic behavior from common nouns in that they mostly do not take determiners or have a plural form. However, some names do take determiners, as in *The New York Times*; in this case, they "are perfectly regular in taking the definite article since they are basically premodified count nouns... The difference between an ordinary common noun and an ordinary common noun turned name is that the unique reference of the name has been institutionalized, as is made overt in writing by initial capital letter." Quirk et al.'s description of names seems to indicate that capitalized words like *Egyptian* (an adjective) or *Frenchmen* (a noun referring to a set of individuals) are not names. It leaves capitalized sequences like *Minimum Alternative Tax*, *Annual Report*, and *Chairman* undetermined as to whether or not they are names.

All of these ambiguities must be dealt with if proper names are to be identified correctly. In the rest of the paper we describe the resources and heuristics we have designed and implemented in Nominator and the extent to which they resolve these ambiguities.

3 Disambiguation Resources

In general, two types of resources are available for disambiguation: context and world knowledge. Each of these can be exploited along a continuum, from 'cheaper' to computationally and manually more expensive usage. 'Cheaper' models, which include no context or world knowledge, do very little dis-

ambiguation. More 'expensive' models, which use full syntactic parsing, discourse models, inference and reasoning, require computational and human resources that may not always be available, as when massive amounts of text have to be rapidly processed on a regular basis. In addition, given the current state of the art, full parsing and extensive world knowledge would still not yield complete automatic ambiguity resolution.

In designing Nominator, we have tried to achieve a balance between high accuracy and speed by adopting a model which uses minimal context and world knowledge. Nominator uses no syntactic contextual information. It applies a set of heuristics to a list of (multi-word) strings, based on patterns of capitalization, punctuation and location within the sentence and the document. This design choice differentiates our approach from that of several similar projects. Most proper name recognizers that have been reported on in print either take as input text tagged by part-of-speech (e.g., the systems of (Paik et al. 1993) and (Mani et al. 1993)) or perform syntactic and/or morphological analysis on all words, including capitalized ones, that are part of candidate proper names (e.g., (Coates-Stephens 1993) and (McDonald 1993)). Several (e.g., (McDonald 1993), (Mani et al. 1993), (Paik et al. 1993) and (Cowie et al. 1992)) look in the local context of the candidate proper name for external information such as appositives (e.g., in a sequence such as *Robin Clark, president of Clark Co.*) or for human-subject verbs (e.g., *say, plan*) in order to determine the category of the candidate proper name. Nominator does not use this type of external context.

Instead, Nominator makes use of a different kind of contextual information — proper names co-occuring in the document. It is a fairly standard convention in an edited document for one of the first references to an entity (excluding a reference in the title) to include a relatively full form of its name. In a kind of discourse anaphora, other references to the entity take the form of shorter, more ambiguous variants. Nominator identifies the referent of the full form (see below) and then takes advantage of the discourse context provided by the list of names to associate shorter more ambiguous name occurrences with their intended referents.

In terms of world knowledge, the most obvious resource is a database of known names. In fact, this is what many commercially available name identification applications use (e.g., Hayes 1994). A reliable database provides both accuracy and efficiency, if fast look-up methods are incorporated. A database also has the potential to resolve structural ambiguity; for example, if *IBM* and *Apple Computers* are listed individually in the database but *IBM and Apple Computers* is not, it may indicate a conjunction of two distinct names. A database may also contain default world knowledge information: e.g., with no other over-riding information, it may be safe to assume that the string *McDonald's* refers to an organization. But even if an existing database is reliable, names that are not yet in it must be discovered and information in the database must be over-ridden when appropriate. For example, if a new name such as *IBM Credit Corp.* occurs in the text but not in the database, while *IBM* exists in the database, automatic identification of *IBM* should be blocked in favor of the new name *IBM Credit Corp.*

If a name database exists, Nominator can take advantage of it. However, our goal has been to design Nominator to function optimally in the absence of such a resource. In this case, Nominator consults a small authority file which contains information on about 3000 special 'name words' and their relevant lexical features. Listed are personal titles (e.g., *Mr.*, *King*), organizational identifiers (including strong identifiers such as *Inc.* and weaker domain identifiers such as *Arts*) and names of large places (e.g., *Los Angeles, California*, but not *Scarsdale, N.Y.*). Also listed are exception words, such as upper-case lexical items that are unlikely to be single-word proper names (e.g., *Very, I* or *TV*) and lower-case lexical items (e.g., *and* and *van*) that can be parts of proper names. In addition, the authority file contains about 20,000 first names.

Our choice of disambiguation resources makes Nominator fast and robust. The precision and recall of Nominator, operating without a database of pre-existing proper names, is in the 90's while the processing rate is over 40Mg of text per hour on a RISC/6000 machine. (See (Ravin and Wacholder 1996) for details.) This efficient processing has been achieved at the cost of limiting 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 cases where Nominator cannot resolve an ambiguity with relatively high confidence, we follow the principle that 'noisy information' is to be preferred to data omitted, so that no information is lost. In ambiguous cases, the module is designed to make conservative decisions, such as including non-names or non-name parts in otherwise valid name sequences. It assigns weak types such as ?HUMAN or fails to assign a type if the available information is not sufficient.

4   The Name Discovery Process

In this section, we give an overview of the process by which Nominator identifies and classifies proper names. Nominator's first step is to build a list of candidate names for a document. Next, 'splitting' heuristics are applied to all candidate names for the purpose of breaking up complex names into smaller ones. Finally, Nominator groups together name vari-

ants that refer to the same entity. After information about names and their referents has been extracted from individual documents, an aggregation process combines the names collected from all the documents into a dictionary, or database of names, representative of the document collection. (For more details on the process, see (Ravin and Wacholder 1996)).

We illustrate the process of name discovery with an excerpt taken from a Wall Street Journal article in the TIPSTER CD-ROM collection (NIST 1993). Paragraph breaks are omitted to conserve space.

... 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. ... Some of these non-lawyer employees are paid at partners' levels. Yet, not having the partner title "makes non-lawyers working in law firms second-class citizens," said Mr. Jordan of Steptoe & Johnson. ...

Before the text is processed by Nominator, it is analyzed into tokens — sentences, words, tags, and punctuation elements. Nominator forms a candidate name list by scanning the tokenized document and collecting sequences of capitalized tokens (or words) as well as some special lower-case tokens, such as conjunctions and prepositions.

The list of candidate names extracted from the sample document contains:

American Bar Association
Robert Jordan
Steptoe & Johnson
ABA
Washington
Dubuque
Mr. Jordan of Steptoe & Johnson

Each candidate name is examined for the presence of conjunctions, prepositions or possessive *'s*. A set of heuristics is applied to determine whether each candidate name should be split into smaller independent names. For example, *Mr. Jordan of Steptoe & Johnson* is split into *Mr. Jordan* and *Steptoe & Johnson*.

Finally, Nominator links together variants that refer to the same entity. Because of standard English-language naming conventions, *Mr. Jordan* is grouped with *Robert Jordan*. *ABA* is grouped with *American Bar Association* as a possible abbreviation of the longer name. Each linked group is

categorized by an entity type and assigned a 'canonical name' as its identifier. The canonical name is the fullest, least ambiguous label that can be used to refer to the entity. It may be one of the variants found in the document or it may be constructed from components of different ones As the links are formed, each group is assigned a type. In the sample output shown below, each canonical name is followed by its entity type and by the variants linked to it.

American Bar Association (ORG) : ABA
Steptoe & Johnson (ORG)
Washington (PLACE)
Dubuque (PLACE)
Robert Jordan (PERSON) : Mr. Jordan

After the whole document collection has been processed, linked groups are merged across documents and 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 are treated as variants of an aggregated *William Clinton* group. In this minimal sense, Nominator uses the larger context of the document collection to 'learn' more variants for a given name.

In the following sections we describe how ambiguity is resolved as part of the name discovery process.

5 Resolution of Structural Ambiguity

We identify three indicators of potential structural ambiguity, prepositions, conjunctions and possessive pronouns, which we refer to as 'ambiguous operators'. In order to determine whether 'splitting' should occur, a name sequence containing an ambiguous operator is divided into three segments — the operator, the substring to its left and the substring to its right. The splitting process applies a set of heuristics based on patterns of capitalization, lexical features and the relative 'scope' of operators (see below) to name sequences containing these operators to determine whether or not they should be split into smaller names.

We can describe the splitting heuristics as determining the scope of ambiguous operators, by analogy to the standard linguistic treatment of quantifiers. From Nominator's point of view, all three operator types behave in similar ways and often interact when they co-occur in the same name sequence, as in *New York's MOMA and the Victoria and Albert Museum in London*.

The scope of ambiguous operators also interacts with the 'scope' of NP-heads, if we define the scope of NP-heads as the constituents they dominate. For example, in *Victoria and Albert Museum*, the conjunction is within the scope of the lexical head *Museum* because *Museum* is a noun that can take PP modification (*Museum of Natural History*) and

205

hence pre-modification (*Natural History Museum*). Since pre-modifiers can contain conjunctions (*Japanese Painting and Printing Museum*), the conjunction is within the scope of the noun, and so the name is not split. Although the same relationship holds between the lexical head *Laboratories* and the conjunction *and* in *IBM and Bell Laboratories*, another heuristic takes precedence, one whose condition requires splitting a string if it contains an acronym immediately to the left or to the right of the ambiguous operator.

It is not possible to determine relative scope strength for all the combinations of different operators. Contradictory examples abound: *Gates of Microsoft and Gerstner of IBM* suggests stronger scope of *and* over *of*; *The Department of German Languages and Literature* suggests the opposite. Since it is usually the case that a right-hand operator has stronger scope over a left-hand one, we evaluate strings containing operators from right to left. To illustrate, *New York's MOMA and the Victoria and Albert Museum in London* is first evaluated for splitting on *in*. Since the left and right substrings do not satisfy any conditions, we proceed to the next operator on the left — *and*. Because of the strong scope of *Museum*, as mentioned above, no splitting occurs. Next, the second *and* from the right is evaluated. It causes a split because it is immediately preceded by an all-capitalized word. We have found this simple typographical heuristic to be powerful and surprisingly accurate.

Ambiguous operators form recursive structures and so the splitting heuristics apply recursively to name sequences until no more splitting conditions hold. *New York's MOMA* is further split at *'s* because of a heuristic that checks for place names on the left of a possessive pronoun or a comma. *Victoria and Albert Museum in London* remains intact.

Nominator's other heuristics resemble those discussed above in that they check for typographical patterns or for the presence of particular name types to the left or right of certain operators. Some heuristics weigh the relative scope strength in the substrings on either side of the operator. If the scope strength is similar, the string is split. We have observed that this type of heuristic works quite well. Thus, the string *The Natural History Museum and The Board of Education* is split at *and* because each of its substrings contains a strong-scope NP-head (as we define it) with modifiers within its scope. These two substrings are better balanced than the substrings of *The Food and Drug Administration* where the left substring does not contain a strong-scope NP-head while the right one does (*Administration*).

Because of the principle that noisy data is preferable to loss of information, Nominator does not split names if relative strength cannot be determined. As a result, there occur in Nominator's output certain '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.*

6 Resolution of Ambiguity at Sentence Beginnings

Special treatment is required for words in sentence-initial position, which may be capitalized because they are part of a proper name or simply because they are sentence initial.

While the heuristics for splitting names are linguistically motivated and rule-governed, the heuristics for handling sentence-initial names are based on patterns of word occurrence in the document. When all the names have been collected and split, names containing sentence-initial words are compared to other names on the list. If the sentence-initial candidate name also occurs as a non-sentence-initial name or as a substring of it, the candidate name is assumed to be valid and is retained. Otherwise, it is removed from the list. For example, if *White* occurs at sentence-initial position and also as a substring of another name (e.g., *Mr. White*) it is kept. If it is found only in sentence-initial position (e.g., *White paint is ...*), *White* is discarded.

A more difficult situation arises when a sentence-initial candidate name contains a valid name that begins at the second word of the string. If the preceding word is an adverb, a pronoun, a verb or a preposition, it can safely be discarded. Thus a sentence beginning with *Yesterday Columbia* yields *Columbia* as a name. But cases involving other parts of speech remain unresolved. If they are sentence-initial, Nominator accepts as names both *New Sears* and *New Coke*; it also accepts sentence-initial *Five Reagan* as a variant of *President Reagan*, if the two co-occur in a document.

7 Resolution of Semantic Ambiguity

In a typical document, a single entity may be referred to by many name variants which differ in their degree of potential ambiguity. As noted above, *Paris* and *Washington* are highly ambiguous out of context but in well edited text they are often disambiguated by the occurrence of a single unambiguous variant in the same document. Thus, *Washington* is likely to co-occur with either *President Washington* or *Washington, D.C.*, but not with both. Indeed, we have observed that if several unambiguous variants do co-occur, as in documents that mention both the owner of a company and the company named after the owner, the editors refrain from using a variant that is ambiguous with respect to both.

To disambiguate highly ambiguous variants then, we link them to unambiguous ones occurring within the same document. Nominator cycles through the list of names, identifying 'anchors', or variant names

that unambiguously refer to certain entity types. When an anchor is identified, the list of name candidates is scanned for ambiguous variants that could refer to the same entity. They are linked to the anchor.

Our measure of ambiguity is very pragmatic. It is based on the confidence scores yielded by heuristics that analyze a name and determine the entity types it can refer to. If the heuristic for a certain entity type (a person, for example) results in a high condifence score (highly confident that this is a person name), we determine that the name unambiguously refers to this type. Otherwise, we choose the highest score obtained by the various heuristics.

A few simple indicators can unambiguously determine the entity type of a name, such as *Mr.* for a person or *Inc.* for an organization. More commonly, however, several pieces of positive and negative evidence are accumulated in order to make this judgement.

We have defined a set of obligatory and optional components for each entity type. For a human name, these components include a professional title (e.g., *Attorney General*), a personal title (e.g., *Dr.*), a first name, middle name, nickname, last name, and suffix (e.g., *Jr.*). The combination of the various components is inspected. Some combinations may result in a high negative score — highly confident that this cannot be a person name. For example, if the name lacks a personal title and a first name, and its last name is listed as an organization word (e.g., *Department*) in the authority list, it receives a high negative score. This is the case with *Justice Department* or *Frank Sinatra Building*. The same combination but with a last name that is not a listed organization word results in a low positive score, as for *Justice Johnson* or *Frank Sinatra*. The presence or absence of a personal title is also important for determining confidence: If present, the result is a high confidence score (e.g., *Mrs. Ruth Lake*); No personal title with a known first name results in a low positive confidence score (e.g., *Ruth Lake, Beverly Hills*); and no personal title with an unknown first name results in a zero score (e.g., *Panorama Lake*).

By the end of the analysis process, *Justice Department* has a high negative score for person and a low positive score for organization, resulting in its classification as an organization. *Beverly Hills*, by contrast, has low positive scores both for place and for person. Names with low or zero scores are first tested as possible variants of names with high positive scores. However, if they are incompatible with any, they are assigned a weak entity type. Thus in the absence of any other evidence in the document, *Beverly Hills* is classified as a ?PERSON. (?PERSON is preferred over ?PLACE as it tends to be the correct choice most of the time.) This analysis of course can be over-ridden by a name database listing *Beverly Hills* as a place.

Further disambiguation may be possible during aggregation across documents. As mentioned before, during aggregation, linked groups from different documents are merged if their canonical forms are identical. As a rule, their entity types should be identical as well, to prevent a merge of *Boston* (PLACE) and *Boston* (ORG). Weak entity types, however, are allowed to merge with stronger entity types. Thus, *Jordan Hills* (?PERSON) from one document is aggregated with *Jordan Hills* (PERSON) from another, where there was sufficient evidence, such as *Mr. Hills*, to make a firmer decision.

8 Evaluation

An evaluation of an earlier version of Nominator, was performed on 88 Wall Street Journal documents (NIST 1993) that had been set aside for testing. We chose the Wall Street Journal corpus because it follows standard stylistic conventions, especially capitalization, which is essential for Nominator to work. Nominator's performance deteriorates if other conventions are not consistently followed.

A linguist manually identified 2426 occurrences of proper names, which reduced to 1354 unique tokens. Of these, Nominator correctly identified the boundaries of 91% (1230/1354). The precision rate was 92% for the 1409 names Nominator identified (1230/1409). In terms of semantic disambiguation, Nominator failed to assign an entity type to 21% of the names it identified. This high percentage is due to a decision not to assign a type if the confidence measure is too low. The payoff of this choice is a very high precision rate — 99 % — for the assignment of semantic type to those names that were disambiguated. (See (Ravin and Wacholder 1996) for details.

The main reason that names remain untyped is insufficent evidence in the document. If *IBM*, for example, occurs in a document without *International Business Machines*, Nominator does not type it; rather, it lets later processes inspect the local context for further clues. These processess form part of the Talent tool set under development at the T.J. Watson Research Center. They take as their input text processed by Nominator and further disambiguate untyped names appearing in certain contexts, such as an appositive, e.g., *president of CitiBank Corp.*

Other untyped names, such as *Star Bellied Sneetches* or *George Melloan's Business World*, are neither people, places, organizations nor any of the other legal or financial entities we categorize into. Many of these uncategorized names are titles of articles, books and other works of art that we currently do not handle.

207

9 Conclusion

Ambiguity remains one of the main challenges in the processing of natural language text. Efforts to resolve it have traditionally focussed on the development of full-coverage parsers, extensive lexicons, and vast repositories of world knowledge. For some natural-language applications, the tremendous effort involved in developing these tools is still required, but in other applications, such as information extraction, there has been a recent trend towards favoring minimal parsing and shallow knowledge (Cowie and Lehnert 1996). In its minimal use of resources, Nominator follows this trend: it relies on no syntactic information and on a small semantic lexicon – an authority list which could easily be modified to include information about new domains. Other advantages of using limited resources are robustness and execution speed, which are important in processing large amounts of text.

In another sense, however, development of a module like Nominator still requires considerable human effort to discover reliable heuristics, particularly when only minimal information is used. These heuristics are somewhat domain dependent: different generalizations hold for names of drugs and chemicals than those identified for names of people or organizations. In addition, as the heuristics depend on linguistic conventions, they are language dependent, and need updating when stylistic conventions change. Note, for example, the recent popularity of software names which include exclamation points as part of the name. Because of these difficulties, we believe that for the forseeable future, practical applications to discover new names in text will continue to require the sort of human effort invested in Nominator.

References

Agarwal R. and L. Boggess, 1992. A simple but useful approach to conjunct identification In *Proceedings of the 30th Annual Meeting of the ACL*, pp.15-21, Newark, Delaware, June.

Brill E. and P. Resnick, 1994. A rule-based approach to prepositional phrase disambiguation, URL: http://xxx.lanl.gov/list/cmp.lg/9410026.

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

Cowie J. and W. Lehnert., 1996. Information Extraction In *Communications of the ACM* , Vol.39(1), pp.83-92.

Cowie J., L. Guthrie, Y. Wilks, J. Pustejovsky and S. Waterman, 1992. Description of the Solomon System as used for MUC-4 In *Proceedings of the Fourth Message Understanding Conference*, pp.223-232.

Jensen K. and Binot J-L, 1987. Disambiguating prepositional phrase attachments by using on-line definitions, In *Computational Linguistics*, Vol. 13, 3-4, pp.251-260.

Hayes P., 1994. NameFinder: Software that finds names in text, In *Proceedings of RIAO 94*, pp.762-774, New York, October.

Hindle D. and M. Rooth., 1993. Structural ambiguity and lexical relations, In *Computational Linguistics*, Vol.19, 1, pp.103-119.

Mani I., T.R. Macmillan, S. Luperfoy, E.P. Lusher, and S.J. Laskowski, 1993. Identifying unknown proper names in newswire text. In B. Boguraev and J. Pustejovsky, eds., *Corpus Processing for Lexical Acquisition*, pp.41-54, MIT Press, Cambridge, Mass.

McDonald D.D., 1993. Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev and J. Pustejovsky, eds, *Corpus Processing for Lexical Acquisition*, pp.61-76, MIT Press, Cambridge, Mass.

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

Paik W., E.D. Liddy, E. Yu, and M. McKenna, 1993. Categorizing and standardizing proper nouns for efficient information retrieval, In B. Boguraev and J. Pustejovsky, eds, *Corpus Processing for Lexical Acquisition*, pp.44-54, MIT Press, Cambridge, Mass.

Quirk R., S. Greenbaum, G. Leech and J. Svartik, 1972. *A Grammar of Contemporary English*, Longman House, Harlow, U.K.

Ravin Y. and N. Wacholder, 1996. Extracting Names from Natural-Language Text, IBM Research Report 20338.

Wacholder N., Y. Ravin and R.J. Byrd, 1994. Retrieving information from full text using linguistic knowledge, In *Proceedings of the Fifteenth National Online Meeting*, pp.441-447, New York, May.