NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval

by Susan Feldman

ONLINE, May 1999

NLP research pursues the elusive quesion of how we understand the meaning of a sentence or a document.

Searching is a language game. Find just the right combination of words and you have the key to the black box of answers that we call a database. Guess wrong, and the box remains mum, or worse, it spews back nonsense. So, we craft our queries with care. Logical and proximity operators are chosen judiciously, words truncated only so far and no further. Search fields are selected for precision without omission. But suppose that we could build some of that knowledge into the system. Suppose that we could give it some of our understanding and also loosen it up so that it was more forgiving. Suppose that we could get it to give us clues about what to do next. That's the hope and the promise of Natural Language Processing (NLP) in information retrieval.

WHAT IS NATURAL LANGUAGE PROCESSING?

Children learn language by discovering patterns and templates. We learn how to express plural or singular and how to match those forms in verbs and nouns. We learn how to put together a sentence, a question, or a command. Natural Language Processing assumes that if we can define those patterns and describe them to a computer then we can teach a machine something of how we speak and understand each other. Much of this work is based on research in linguistics and cognitive science.

NLP research pursues the elusive question of how we understand the meaning of a sentence or a document. What are the clues we use to understand who did what to whom, or when something happened, or what is fact and what is supposition or prediction? While words--nouns, verbs, adjectives and adverbs--are the building blocks of meaning, it is their relationship to each other within the structure of a sentence, within a document, and within the context of what we already know about the world, that conveys the true meaning of a text.

People extract meaning from text or spoken language on at least seven levels. In order to understand Natural Language Processing, it is important to be able to distinguish among these, since not all "NLP" systems use every level.

Phonetic or Phonological Level

Phonetics refers to the way the words are pronounced. This level is not important for written text in information retrieval systems. It is crucial to understanding in spoken language and in voice recognition systems. I will never forget my freshman roommate from the Bronx rattling on about the great play they were reading in English by a guy named "Shore." "I never heard of him," I said. "You never heard of George Bernard Shore [Shaw]?" she answered.

Morphological Level

The morpheme is a linguistics term for the smallest piece of a word to carry meaning. Examples are word stems like *child* (the stem for *childlike, childish, children*) or prefixes and suffixes like *un-*, or *-ation*, or *-s*. Many new search engines are able to determine word stems on a rudimentary level. This usually means that they can automatically offer you both the singular and plural form of a word, which is a nice feature. Automatic use of morphology without accompanying understanding, however, can also return some pretty funny documents. One example that I stumbled over in PLS in its early days was that it would stem a word like "*communication*" and return *community, commune, communication,* or *communism.*

Syntactic Level

When we parsed a sentence in grade school, we were identifying the role that each word played in it, and that word's relationships to the other words in the sentence. Position can determine whether a word is the subject or the object of an action. *Fred hit John* and *John hit Fred* both use the same words, but the meaning is quite different, particularly to the person who is the object of the verb *hit.*

NLP systems, in their fullest implementation, make elegant use of this kind of structural information. They may store a representation of either of these sentences, which retains the fact that Fred hit John or vice versa. The may also store not only the fact that a word is a verb, but the kind of verb that it is. They characterize dozens of different kinds of relationships, such as AGENT, POSSESSOR OF, or IS A. When this additional information is stored, it makes it possible to ask "Who left the golden apple for Atlanta?" without retrieving reams of information about apples in the city of Atlanta.

The structure of a sentence conveys meanings and relationships between words even if we don't know what they mean. Take, for example, this line from Lewis Carroll's poem *The Jabberwocky*:

> Twas brillig in the slithy toves.

In this sentence, we know that the scene that is being described happened in the past (*Twas*). Brillig is probably an adjective, perhaps describing the weather in the *slithy toves*. We know that *slithy* is an adjective describing toves, which is a plural noun. All this meaning is conveyed by the syntax of the sentence. It fits the same template as "It was windy in the silent city."

Semantic Level

The semantic level examines words for their dictionary meaning, but also for the meaning they derive from the context of the sentence. Semantics recognizes that most words have more than one meaning but that we can identify the appropriate one by looking at the rest of the sentence. The charm of the *Amelia Bedelia* books by Peggy Parish comes mostly from Amelia Bedelia confusing the senses of a word (draw the drapes, dress the chicken, dust the furniture, put out the lights).

Let's look at our Lewis Carroll example again. *The Jabberwocky* makes sense at the syntactic level. It has some major problems at the semantic level. At the semantic level, we can finally ask what do *brillig, slithy* and *toves* mean. We found quite a bit of meaning in that sentence on the syntactic level, but it fails completely at the semantic level.

Some meaning is entirely dependent on context. It is context that determines which sense to assign to the verb *to draw* in "*He drew his snickersnee*" (*The Mikado*, Gilbert and Sullivan). Even if we don't know what a *snickersnee* is, the fact that the rest of the song is about an execution determines that he is talking about drawing a sword of some sort, not an artistic activity.

In English, precise meaning is often carried by noun phrases--two nouns together that mean something quite different from their constituent words. In fact, using an appropriate noun phrase is an excellent technique for searching any retrieval system. By using both syntactic and semantic levels, NLP can identify automatically phrases such as box office, carbon copy, dress circle, walking stick, blind date, or reference book.

Discourse Level

This level examines the structure of different kinds of text and uses document structure to extract additional meaning. For instance, a newspaper article typically reports the most important facts--who, what, when, where, how--at the beginning, usually in the first paragraph. It makes predictions about the impact of the events towards the end. In contrast, a mystery novel never tells you who did it and how it was done until the end. A technical document starts with an abstract, describing the entire contents of the document in a single paragraph and then enlarging on all these points in the body of the work. NLP uses this predictable structure "to understand what the specific role of a piece of information is in a document, for example-- is this a conclusion, is this an opinion, is this a prediction or is this a fact [1]?"

Pragmatic Level

The role of the word within the structure of a sentence narrows the choice of reasonable meanings.

The practical or pragmatic level depends on a body of knowledge about the world that comes from outside the contents of the document. For example, we know that France, Germany, Belgium, and Spain are in Europe, but we have to tell most information retrieval systems to search for a list of all the countries in Europe if we are interested in the predicted effects of the European Currency Unit (ECU) on their separate economies. Some information retrieval researchers feel that the only way to add this level of outside knowledge is to gather everything we know about the world and use it as a reference guide or knowledge base for information systems. One extremely thorough example of this approach is the Cyc Project, started by Doug Lenat in 1984 [2]. It is "constructing a foundation of basic 'common sense' knowledge--a semantic substratum of terms, rules, and relations--that will enable a variety of knowledge-intensive products and services." You can understand the extent of this project if you realize that, like a baby, but far more slowly, these researchers had to define everything that they knew about the world. The problem with building such a vast knowledge base is that it takes too long, and it looks backward to what we knew while it is not very good at adding new information quickly.

All these levels of understanding are intertwined. The role of the word within the structure of a sentence narrows the choice of reasonable meanings. A combination of context and syntax determine what *plant* means in the sentence, "I watered the plant as soon as I got back to the house." We know that it isn't reasonable to assume that plant is a verb in this sentence, since it is the object of the verb "to water." That is context-independent. However, within the context of this sentence we can narrow the meaning still further, since it is unlikely that the speaker carried home an industrial plant, or that she would water it when she got it there.

Because each of these levels of language understanding follows definable patterns or templates, it is possible to inject some language understanding into a computer system by using those definitions. The higher the level, the more difficult this becomes, however. One of the greatest challenges for NLP systems is to distill a sentence or document down to an absolutely precise, unambiguous representation of its contents. These formal representations must be unambiguous and contain not only the words and meaning in a sentence, but the structure of that sentence as well. Computers, after all, are not human, and they will do only what they are told to do. They do not make flying leaps of understanding based on skimpy evidence, so they do not deal well with ambiguity.

STAGES OF INFORMATION RETRIEVAL PROCESSING

If you want to understand the role of NLP in information retrieval, you must understand the retrieval processes first. NLP can be applied at any or

all of these stages. I believe that systems that use more NLP, and at more levels of language understanding, have the most potential for building the data mining and advanced information retrieval systems of the future.

First, let's define some terms. We throw around words like Boolean, statistical, probabilistic, or Natural Language Processing fairly loosely. Boolean systems, i.e., traditional systems--Dialog and LEXIS-NEXIS are examples--are based on Boolean logic. They use operators such as AND, OR, NOT in mathematical statements. Boolean systems are exact match systems. You get what you ask for. If you ask for *rebellion* and *Angola* within ten words of each other, and a document contains *rebellion* and *Angola* within 15 words of each other, you won't retrieve it.

Probabilistic or statistical systems use probability and statistics to predict what might be not only exact, but close matches to a query. In these systems, you get what you ask for, but you might also get what you *should* have asked for. Unfortunately, you also retrieve other documents that contain your query terms, but not the information you wanted. These systems will return *rebellion* and *Angola* as long as either appears in the document, but they will first return those documents with more occurrences of either or both terms, preferably as close together as possible.

NLP systems will look at how the words are used and what they mean within the query. They may use either Boolean or statistical methods for matching a query, such as *rebellion (w10) Angola*, or they may not be able to handle the Boolean form of the query. Where they excel is in queries that give them real text to chew on. For instance, *"When did the rebellion in Angola start and who are the leaders of each faction?"* gives an NLP system the chance to use its strengths. In a query such as this one, the NLP system would interpret correctly that you wanted to know a date about the people involved, and that the rebellion happened in Angola. It has already captured the phrases within any document.

In reality, it is possible to use pieces of all of these approaches. A Boolean system could easily return relevance ranked documents. Many statistical systems use some NLP features such as automatic stemming and identification of proper nouns. They may even look like they understand meaning when they offer related words in a "concept search." What they are really doing is giving you a list of other words that occur frequently within the same documents that contain your query terms.

HOW INFORMATION RETRIEVAL WORKS

In very rough terms, a basic text retrieval system consists of three separate files: the file of full records or full text documents, including all bibliographic and indexing information; the Dictionary, which is an alphabetical list of all the unique words in the database; and an Inversion List, which stores all of the locations of occurrences for each word in the dictionary. This structure is called an inverted file. Searching very large databases is efficient with this file structure, since each word in the Dictionary is an entry point for beginning a search.

Step 1: Document Processing
Documents are input to the system. While several forms of tagging and information extraction can also take place at this first stage, most information retrieval systems build an inverted file, or list of words in alphabetical order. Stopwords are left out of this list. New documents are interfiled into the existing list, so that the system has all occurrences of every word in one place, with their position within each document. Increasingly, text retrieval systems also add or create knowledge bases with internal lexicons, semantic networks, or lists of phrases, synonyms, and personal pronouns.

Many systems extract additional information at this stage, or perform various operations on the words when they store them. These may include stemming, identification of part of speech, whether the word is a proper noun (Gates versus gates), and perhaps the relationship of that word to the others within the sentence, the paragraph, or the document. Some may also automatically assign indexing terms, or broad subject categories. A few NLP systems create and store a formal representation of each sentence including the role each word plays and its relationship to other words in the sentence. Statistical systems compute weights at this stage.

Step 2: Query Processing
When a query comes in, it must be interpreted for the system. In Boolean systems, this is not as complex an operation as it is in full NLP systems, since the searcher has already phrased the question in computer-interpretable terms. NLP and statistically-based systems must do some of the work that searchers do in preparing a query. A statistical system identifies the terms to search for and it may look for stems and singular and plural forms. It may also assign weights to each term. A full NLP system tags all the parts of speech, identifies objects, subjects, agents, verbs, and also may expand geographic terms as well as add synonyms and alternate forms for proper nouns. Then it creates an unambiguous representation of the query for the system to match against its knowledge base. A partial NLP system would perhaps identify stems, as well as basic syntax and then create a query representation. Term weights are computed.

Step 3: Query Matching
The interpreted query is matched against the inverted file and the knowledge base, if there is one. Traditional online services match each query word exactly as it is entered, in the combination specified by the searcher. Thus, if we ask for *slithy()toves*, we will get *slithy toves*, but not *toves were slithy*. Statistical systems will get us *slithy () toves, slithy AND toves, slithy OR toves*. A full NLP system might be able to match the query "Is slithiness a common condition in toves?" Or "I am interested in the location of all slithy toves in New England." NLP would expand "New England" and add synonyms from its knowledge base, perhaps for "location."

Step 4: Ranking & Sorting
Once all the candidate documents are selected that match the query, they are sorted by date, by field, or by how relevant the document is predicted to be to the query.

Traditional systems commonly return results sorted in reverse chronological order, but they could sort in ascending chronological order, by author's name, by report number, etc. Statistical and NLP-based systems use the same kind of relevance ranking techniques. They rank the retrieved documents so that the top ones returned match the query the closest. Where these systems vary is in how they define a "close match." Relevance ranking can be the topic of a whole other article, but briefly, relevance is determined by first assigning a weight to each term. This weight is determined by how frequently the term appears in the document, and also how often it occurs in the database as a whole. Relatively rare terms in the database that occur frequently in a document are assigned a higher weight than words that are more common. Documents that have more occurrences of the query terms, and in which the terms appear closely together in the text are ranked highly. Usually (but not always) a document that contains all the query terms will be ranked higher than one that contains fewer terms but more appearances of those terms. Variations in how factors such as proximity, location of terms in the document, and emphasis on occurrence of all terms are weighted account for some of the differences in how search engines perform. There is also no reason why any system can't return documents sorted by any criterion. A Boolean system could add relevance ranking, and a statistical system could return documents in order by date.

NLP IN INFORMATION RETRIEVAL

Natural Language Processing can be added at any or all of these stages, using any or all of the seven levels of understanding. Full NLP interprets and stores meaning at all stages and at all levels for both the query and the document. As in any computer system, the choice of how much to add and where is based on practical considerations, such as how computationally expensive the additional processing will be. Will it slow down the processing of queries unacceptably? How much overhead does it add to document processing? Are the retrieved results so much better that it is worth the trade-off?

Most systems that boast that they are NLP are actually using NLP on the lower levels of understanding, and often this is for query interpretation only. For instance, most Web search engines can automatically stem query words for singular and plural forms. Some, like Infoseek and Ask Jeeves have added the ability to interpret some syntax by parsing the query sentences or phrases. They do not apply this technique to document storage.

Once we progress to the semantic level, the systems that qualify as NLP are relatively rare, particularly when we look at both document processing and query processing. ConQuest, now part of Excalibur, incorporates an extensive "lexicon" or dictionary that is implemented as a semantic network. These stored meanings are used as a "knowledge base" so that synonyms can be retrieved even if they aren't specifically requested. InQuery parses sentences, stems words, and recognizes proper nouns and concepts based on term co-occurrence. DR-LINK from MNIS performs full document and query processing on all levels of language understanding including the discourse and pragmatic levels, although not to the extent of adding a full Cyc knowledge base.

As computers increase their processing speeds and new approaches optimize the process for adding documents and matching queries, these questions will become moot. Therefore, we should consider if there are actual advantages to adding full NLP to information retrieval systems.

Document Processing

This area has the most promise for new improvements to information retrieval, particularly for Knowledge Management applications. It is at this stage that information is extracted and stored about each document. Retrieval systems can only retrieve what they contain. The more extensively analyzed the content is, the more potential the knowledge base has for use with future, more intelligent systems. If we can extract and store a rich collection of data, even if we don't use that data immediately, we have created a strong foundation for future applications as they arrive. The knowledge base can be used today by search engines to retrieve documents. It will be ready to add data mining applications as these become available. Question answering systems also depend on this deep analysis of text if they are to return smart answers. Very few NLP systems extract information on all the NLP levels. The only one I know of is the DR-LINK system from Manning and Napier Information Services. Others may extract entities and perform some stemming. The LinguistX parser from Xerox PARC extracts syntactic information, and it is used in Infoseek.

Query Interpretation and Matching

There are two obvious advantages to using NLP at this stage. The first is that it is much easier to convey our information needs, including intention and meaning, if we can use the full power of real language. Speaking in code is difficult, and it leaves out important aspects of thought. The second is that use of full NLP can eliminate problems that plague us, such as false drops and other right word/wrong meaning retrievals. NLP can focus a query without eliminating potentially useful documents. It should improve both recall and precision. NLP can also expand a query to add synonyms and alternate forms as well as related geographic terms.

Document Ranking

NLP can improve the ranking of documents because it has done a better job of matching the meaning and the intention of the query. It has more evidence of what is really relevant on which to base a relevance judgment.

SOME COMMON PROBLEMS IN INFORMATION RETRIEVAL

Any automatic system has stumbling blocks, and information retrieval systems are no exception, particularly since they must deal with the vagaries of human language. Human languages pose all kinds of complexities that are hard to resolve without resorting to the actual meaning of a word. Ambiguity is rife unless the system can bring not only the knowledge gained from the context of the text, but also the knowledge of the real world that people carry with them. NLP can solve some common problems that information retrieval systems have:

- **Too Many Synonyms**
  We delight in saying the same thing in as many different ways as possible. The better the writer, the more the variety. This creates headaches for searchers who must try to guess how an author phrased an idea. We need retrieval systems that match ideas, not words. In addition, different regions of the world or different subject specialties may have terms they use to mean the same thing, such as *lorry* and *truck*, or *elevator* and *lift*, or *pump* and *impeller*, or *hypertension* and *high blood pressure*. In an exact match system, we miss important works if we don't ask for these other synonyms. An NLP system should be able to expand the query automatically with appropriate synonyms and place names.

- **Too Many Meanings**
  Most words have more than one meaning. In fact, Dr. Elizabeth Liddy, who created the DR-LINK retrieval system now used by Manning and Napier Information Services, says that most words have, on average, seven meanings. Consider how many ways you can use the words "table," "fly," "bank," "charge," or "seat." This phenomenon is called "polysemy." A system that can determine the proper meaning of a word "disambiguates" it, in information retrieval parlance.

  We use different words that mean the same thing, but we also resort to metaphor and simile, just to add to the confusion ("he was a lion, stout of heart" might conceivably be talking about a fat cat). Ambiguity results when we examine words outside their context. For instance, a searching pitfall Barbara Quint often cites is "Ask for terrorism and you get sports." If I were interested in unrest in Africa, in a Boolean system I might ask for (*rebellion or battle or uprising or skirmish*) and (*South()Africa or Kenya or Rwanda or Zambia or Zaire orÉ*). This query would almost certainly retrieve, *"South African rugby team wins game in all-out battleÉ"* Quint adds, "It works for crime too: *Sammy Sosa stole home.*" An NLP system can determine from the context of the query that you are looking for political insurrections, not sports, and it will eliminate all the sports documents. I know this works because I've tested it.

- **Inability to Specify Important but Vague Concepts**
  If we want predictions of future stock performance, likelihood of instability in a country, or whether the interest rates will be raised in the next six months, there are no easy commands to describe these ideas in a traditional Boolean or even a statistically-based system. Since searchers are clever, they've created some workarounds for this problem. For instance, when I had this question with the Predicasts files I used to call Joe Hecht at their help desk. He obligingly gave me long strings of adjectives to plug in. Here's one he gave me for forecasts and predictions:

  > The following search string ANDed to a targeted set was often helpful in retrieving records containing projections or forecasts:
  > *predict? or projecting or projected or future or forecast? or trend or outlook or year()(200? or 201?)*

- **False Drops**
  Most of today's commercial and Web search technologies retrieve information without knowing what it means. They do this by matching strings of letters (words) in the query to the documents in the database in order to find exact or best matches. This is like trying to carry on a conversation with a parrot. The parrot can mimic speech, but it ties words to, at most, a treat or a curse, not to their inherent meaning. Even the much-vaunted ELIZA, an expert system for counseling, matches patterns and keywords, not meaning. The result in both these cases can be inappropriate utterances, in other words, false drops. Boolean systems in particular are plagued by this problem. If the searcher casts a fairly wide net with a simple AND query, a search for Japan's position on the NAFTA accords could retrieve a news round-up article from the *Financial Times* that discusses NAFTA in paragraph one, and Japan's problems with the falling yen in paragraph two. All query terms are present in the document, but the document is not relevant. Both traditional and statistical systems are prey to this problem, but it is worse in Boolean systems because they don't automatically look for proximity and frequency of terms.

- **Indexing Inconsistency**
  In the best of all possible worlds, all documents on the same subject would be assigned the same indexing terms. In fact, some studies have shown that indexer consistency is at best 50%. Given this fact of life, we need systems that can interpret variations on an idea and retrieve all of them.

- **Spelling Variations and Errors**
  What is the right way to spell gray/ grey, or theatre/theater, or aluminum/ aluminium? What about errors in spelling or typing that get into print and databases? This problem is compounded as we add automatically scanned text using optical character recognition to the databases of the world. Without careful proofreading, these scanned texts can easily add 30 spelling errors to each printed page.

The purpose of an information retrieval system is to find the most relevant materials for the user while it eliminates the least relevant. We measure the ability of the system to accomplish this feat as its precision and its recall. It is usually viewed as an either/or proposition: more precision/ less recall or more recall/less precision. Boolean systems fit on one end of this spectrum: they find precisely what you ask for. If you have asked for what you want (not a very common situation on the first query), then you will get what you wanted. If, as is more common, you have only a vague idea of how to ask your question, you may retrieve nothing that is helpful.

Statistical systems are positioned at the other end of this spectrum. They emphasize recall and yield large sets. However, they are also designed to give you precision by ranking the large sets they retrieve to present the most relevant documents first. The advantage of this kind of information retrieval system is that if you haven't asked for what you want, you may be given clues about how to modify your query from the partially relevant materials that are also retrieved. Alternate terms that you find co-occurring with your own query terms can help you focus or expand your query.

Full NLP systems can refine the statistical retrieval to improve its precision, but they can also expand the Boolean retrieval in a focused manner, to improve its recall. They do this by adding additional sources of evidence from natural language understanding when they interpret both the document and the query.

As searchers, we have developed a set of devices that help us hone and focus a query. These are based on our knowledge of language and text and how they are structured. Therefore, we ask for proximity of search terms, for phrases rather than actual words, for lists of synonyms, and for words that appear in the lead paragraph. Each of these commands sharpens the query, but we also lose some information that might be valuable. In other words, we are looking for a right answer, or a good answer, but not for all the possible right or good answers. Most researchers in information retrieval agree that information retrieval systems rarely retrieve more than 20% of the possibly relevant materials in the database. That should be of some concern to our profession. We can not, in all honesty, claim to have found everything on a subject if all we have found is 20%. Therefore, we should be extremely interested in any new technologies that can boost both precision and recall at the same time. Where will Natural Language Processing take us?

CURRENT RESEARCH DIRECTIONS

Data Mining and Entity Extraction

This is a promising area of research with some products, such as NetOwl from SRA and KNOW-IT from TextWise, already on the market. Good entity extraction is entirely an NLP-dependent process. It extracts the names of people, places, and things from text and stores them, sometimes with other related information. While this seems relatively simple if all you want is a list of proper names, recall that every sentence as well as every proper name starts with a capital letter. How do you decide if the first word at the beginning of the sentence is also a proper name? NLP systems can go further and store the names together with indicators of what they are. Some of those indicators are other ways to refer to the person--*the President,* or *the senior Senator from New York* would retrieve President Clinton or Senator Moynihan. Some systems also store chronological information about an entity, enabling them to extract Clinton as President of the U.S. in 1998, but Franklin Roosevelt in 1941. The chronological information makes it possible to construct a timeline of an entity's history automatically. We could follow Zubin Mehta as he moved from conducting one orchestra to another. If we can't remember the current Speaker of the House, we could find that knowledge by asking "Who is the Speaker of the House in 1999?"

Systems like KNOW-IT store those entities as well as their relationships to each other. When we have this information in our knowledge base, we can ask "Who bombed Iraq in 1998," and not retrieve a list of whom Iraq bombed.

The import of good data mining technology is that we can build and use a stored knowledge base of information that will help us determine patterns and trends--of behavior, of occurrence, of research interests. For instance, data mining techniques can help medical researchers find out which antibiotics are becoming drug resistant over time. In intelligence work, being able to detect patterns is extremely important. Analysts

sometimes miss small changes that can become important clues to future behavior by companies or other countries. Presented as we are by an overwhelming welter of facts, tools that help us sort facts into patterns are invaluable. Data mining techniques, with good full NLP behind them, will form the basis for decision support systems. These systems will offer suggestions for action, based on data stored in a knowledge base.

Improved Filtering or Alerting Techniques
One of the main problems with automatic alerting services is that they don't change as the subject field changes. Significant information can be lost as terminology moves to the new jargon of the month. For instance, an alert set up two years ago for information retrieval would have missed *data mining, datamining, knowledge bases, knowledge management,* and *decision support systems.*

Cross Language Retrieval
An increasingly polyglot world needs to have access to information on a subject no matter what language it is written in. If an NLP system can reduce a sentence to an abstract representation of its meaning, what would stop it from retrieving information from many languages simultaneously? The representations should be the same no matter what language they came from originally. Current translating systems often rely on stored dictionaries and rote translating one word at a time from one language to the other, resulting in translations, such as "How does Ca go?" from the original "Comment a va?" Retrieving ideas instead of words seems a more promising approach, and some of these systems are almost ready for the market.

Speed
How do you extract and store large quantities of information about documents and then use it in matching queries? Is it faster to store phrases as single units or as separate words? What are techniques for representing text formally so that it is unambiguous to the computer?

Automatic Summarization
One of the major problems professionals have these days is sorting through the volumes of information that are deposited daily on their desks and desktops. Good trustworthy summaries of most of that information would enable the user to determine what should be read in more depth, and what just needs a cursory glance. Imagine coupling this with a data mining system so that the summary could be presented as a quick one-page status report.

Machine-Aided Indexing
Traditional indexing and abstracting techniques can't keep up with the flow of information in a timely manner. For time-dependent content, some sort of automatic categorization is needed. Studies on indexer consistency (or the lack thereof) indicate that any system that manages to be accurate and consistent 50% of the time is performing as well as human indexers do.

Domain-Dependent Versus Domain-independent Systems
How much knowledge does a retrieval system need to have in its knowledge base in order to work effectively and intelligently? Some companies build specialized knowledge bases for each subject domain, but this limits how adaptable they are to a different subject. Others rely on what they know about specialized text structure in order to operate independently of the domain. However, most NLP systems do maintain a lexicon of some sort. For instance, knowledge of geography and place names enables them to return Maine, Massachusetts, New Hampshire, Vermont, Rhode Island, and Connecticut when you ask for hotels in New England. This is called query expansion, and well-designed systems also add synonyms to your original query if the information is in the system's knowledge base.

CONCLUSION

Without NLP, we have gone about as far as we can go. Text databases are getting bigger. Search engines are returning larger and larger sets of documents. While Boolean search techniques allow us to narrow down our retrieval to a manageable size, they eliminate too many potentially

valuable documents. Statistical search techniques overwhelm us with documents, even with relevance ranking. NLP presents new tools for honing a search query so that it states our information need fully and then matches that query with an elaborate knowledge base built with NLP techniques.

My prediction is that the best systems in the future will be those that combine useful features from several information retrieval technologies. They may incorporate the precision of Boolean logic in its extended form with statistical ranking and recall. They will extract and store meaning at all levels of natural language understanding, and they will use intelligent agents to supply updated profiles of what the user's interests are. Then they will combine all these technologies with as-yet-to-be invented elegant techniques that tailor retrieval to the needs of the individual. We can dream, can't we?

ASK JEEVES
Ask Jeeves, Inc.
918 Parker Street, Suite 12
Berkeley, CA 94710
510/649-8685
Fax 510/649-8633
jeeves@ask.com

CENTER FOR INTELLIGENT INFORMATION RETRIEVAL
Department of Computer Science
Lederle Graduate Research Center, Box 34610
University of Massachusetts, Amherst
Amherst, MA 01003
413/545-0463
Fax 413/545-1789 croft@cs.umass.edu

CLARITECH CORPORATION
5301 Fifth Avenue
Pittsburgh, PA 15232-2124
412/621-0570
Fax 412/621-0569 info@Claritech.com

CYCORP, INC. (CYC PROJECT)
3721 Executive Center Drive, Suite 100
Austin, TX 78731
512/342-4000; Fax: 512/342-4040
info@cyc.com

ERLI

USA
Citicorp Center One Sansome Street
Suite 1050 (10th floor)
San Francisco, CA 94104
415/392-6500
Fax 415/392-6555
info-usa@erli.com

EUROPE
Immeuble "Le Méliès"
261, rue de Paris
F-93556 Montreuil Cedex
France

INQUIZIT
InQuizit Technologies
725 Arizona Avenue, Suite 204
Santa Monica, CA 90401
310/576-4910, 888/576-4910
Fax 310/576-7961
info@inquizit.com

MNIS
Manning & Napier Information Services
1100 Chase Square
Rochester, NY 14604
800/278-5356, 716/454-0050
Fax 716/454-2516 service@mnis.net

MUSCAT
The Westbrook Centre
Milton Road
Cambridge, CB4 1YG
England
+44 (0) 1223 715000
Fax +44 (0) 1223 715001 http://www.muscat.com/

ORACLE CORPORATION (CONTEXT)
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065

SOVEREIGN HILL SOFTWARE(INQUERY
) 100 Venture Way
Hadley, MA 01035
413/587-2222
Fax 413/587-2246 info@sovereign-hill.com

SRA (NETOWL)
SRA International, Inc.
4300 Fair Lakes Court
South Building
Fairfax, VA 22033-4232
800/511-6398
Fax 703/802-4145 netowl@netowl.com

TEXTWISE

+33 (0)1 49 93 39 00
+33 (0)1 49 93 39 39 info-eur@erli.fr

EXCALIBUR
1921 Gallow Road, Suite 200
Vienna, VA 22182
703/761-3700
Fax 703/761-1990
info@excalib.com
http://www.excalib.com/

INFOSEEK
Infoseek Corporation
1399 Moffett Park Drive
Sunnyvale, CA 94089
800/781-INFO, 408/543-6000
Fax 408 734 9350

TextWise LLC
2-212 Center for Science and Technology
Syracuse, NY 13244
315/443-1989
Fax 315/443-4053 webmaster@textwise.com

XEROX PARC
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
650/812-4000
http://www.parc.xerox.com/

ACKNOWLEDGEMENTS

Thanks to Joe Hecht who was my favorite Predicasts Help desk contact in the old days. Thanks also to Barbara Quint, searcher extraordinary, and to Liz Liddy, my mentor in all things NLP.

FOR FURTHER REFERENCE

Allen, James. *Natural Language Understanding.* 2nd Ed. Addison Wesley, 1995.

Brenner, Everett. "Beyond Boolean--New Approaches to Information Retrieval." National Federation of Abstracting and Information Services, Philadelphia, 1996.

Croft, W. Bruce. "Approaches to Intelligent Information Retrieval." *Information Processing & Management* 23, No. 4 (1987): pp. 249-254.

Feldman, Susan E. "Searching Natural Language Search Systems" *Searcher* (October, 1994): pp. 34-39.

Feldman, Susan E. "Testing Natural Language: Comparing DIALOG, TARGET, and DR-LINK." *ONLINE* 20, No. 6 (Nov. 1996) pp. 71-79.

Harman, Donna K., ed. *The First Text Retrieval Conference (TREC-1).* Bethesda, MD: National Institute of Standards and Technology, March 1993. (Available from NTIS: PB93-191641).

Hayes, Philip J. and Gail Koerner. "Intelligent Text Technologies and Their Successful Use by the Information Industry". *Proceedings of the Fourteenth National Online Meeting,* 1993 : pp. 189-196.

Jacobs, Paul S., ed. *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* Hillsdale, NJ:

Lawrence Erlbaum Associates, 1992.

Jacobson, Thomas L. "Sense-making in a Database Environment." *Information Processing and Management* 27, No. 6 (1991): pp. 647-657.

Jacoby, J. and V. Slamecka. *Indexer Consistency Under Minimal Conditions.* Documentation, Inc. Bethesda, MD. 1962 RADC-RDR-62-426. Contract AF30(602)-2616. ASTIA AD288087

Kowalski, Gerald. *Information Retrieval Systems: Theory and Implementation.* Kluwer. Boston, 1997.

Lenat, Douglas B. and Ramanathan V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project..* Addison-Wesley, 1990.

Liddy, Elizabeth. "Enhanced Text Retrieval Using Natural Language Processing." ASIS Bulletin (April/May 1998). On the Web at http://www.asis.org/Bulletin/Apr-98/liddy.html.

Liddy, Elizabeth D. Woojin Paik, Edmund S. Yu, and Mary McKenna. *Document Retrieval Using Linguistic Knowledge.* Syracuse University, 1994, RIAO Proceedings, 1994.

Meadow, Charles T. *Text Information Retrieval Systems.* Academic Press, 1992.

Pritchard-Schoch, Teresa. "Natural Language Comes of Age." *ONLINE* 17, No. 3 (May 1994): pp. 33-43.

Spink, Amanda and Howard Greisdorf. "Partial Relevance Judgments and Changes in Users' Information Problems During Online Searching." National Online Meeting, 18th Proceedings (1997): pp. 323-334.

REFERENCES

[1] Liddy, Elizabeth. Enhanced Text Retrieval Using Natural Language Processing. *ASIS Bulletin*, April/May 1998. On the Web at http://www.asis.org/Bulletin/Apr-98/liddy.html.

[2] Lenat, Douglas B.; Guha, Ramanathan V. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley. 1990. Or on the Web at http://www.cyc.com/.

Communications to the author may be addressed to Susan Feldman, Principal, Datasearch, 170 Lexington Drive, Ithaca, NY 14850; 607/257-0937; sef2@cornell.edu.