In N. R. Adam, B. K. Bhargava & Y. Yesha (Eds.), Digital Libraries. Current Issues (pp. 165-180). Berlin: Springer-Verlag, 1995.

Corpus linguistics for establishing the natural language content of digital library documents

Robert P. Futrelle, Xiaolan Zhang and Yumiko Sekiya

Biological Knowledge Laboratory and
Scientific Database Project
College of Computer Science
161 Cullinane Hall
Northeastern University
Boston, MA 02115

{futrelle,xzhang,sekiya}@ccs.neu.edu

Abstract

Digital Libraries will hold huge amounts of text and other forms of information. For the collections to be maximally useful, they must be highly organized with useful indexes and intra- and inter-document linkages. This brings with it a demand for ever-better methods for automated analysis of text to build the indexes and links. It requires turning implicit information, "encrypted in natural language" into explicit information. We discuss approaches to the automation task built on the techniques of corpus linguistics. This paper focuses on word classification as an example of the utility of corpus methods. Results are presented for the syntactic and semantic classification of words from a biological corpus. The word classes identified can then be used for indexing, query expansion, syntactic analysis and for linking separate library collections by aligning word senses. The paper also discusses derivative objects, diagram analysis and authoring tools. Finally, we outline a new approach to word classification and other language structure analyses based on the *minimal complexity principle*, in turn based on the theory of Kolmogorov complexity.

Introduction

Any collection of information that is called a "library" must be an organized collection, and the Digital Libraries of the future (DLs) are no exception (Lunin and Fox 1993; Schnase, Leggett et

al. 1994). Access is not possible without organization. Some of this organization is explicitly introduced at authoring time as the descriptive elements: source, title, authors, affiliation, keywords, etc. A great deal more information is only implicit, "encrypted" in the document's text. Thus, objects under discussion, the processes or events, the times and places of events, value judgments, etc., are all useful in organizing the collection to guide prospective users of DLs.

The natural language methods discussed in this paper are broadly referred to as "Corpus Linguistics" methods. This is intended to denote a variety of methods (parsing, natural language understanding, semantic analysis, etc.) but as adapted to work on very large corpora. See the two issues of the journal *Computational Linguistics* specifically devoted to using large corpora, for a good overview of corpus linguistics (Church and Mercer 1993).

The structured information extracted from a corpus serves at least two purposes. The first is to improve the indexing of the collection, often allowing the user to access focused information, typically small parts of large documents that are of the most interest. The second is to construct *derivative* objects which codify or summarize specific types of information from one or more documents. These could be automatically constructed abstracts or tabular summaries of data.

The extraction of implicit information must be automated as much as possible, because the effort that would be required to do it manually is too large. It would be best if even the categories of information were discovered automatically, "bootstrapped" from the corpus itself. (For an excellent justification of the need for bootstrapping approaches, see (Finch and Chater 1993).) When bootstrapping is used, the collection becomes self-organizing and, if done well, should produce consistent descriptions free of biases created by ad hoc organizational designs prescribed in advance. Self-organization, by its nature, is useful in characterizing specific knowledge domains, and genres, and their sublanguages, e.g., popular versus technical writing. The challenge then becomes, how can we bootstrap from text to produce both structured categories and specific information?

This paper focuses on the biological research literature, the largest single collection of scientific literature in the world, comprising some 600,000 articles per year, totaling 3 billion words.

Our first experiments used a 250 thousand word collection of biological abstracts. Current work is focusing on a 4 million word corpus, the 1993 papers of the *Journal of Bacteriology*, taken from the American Society for Microbiology (ASM) CD-ROM.

We emphasize the importance of linguistic analysis for building and using digital libraries. Most of the discussion centers on the syntactic and semantic classification of words, as developed by many workers and extended and applied by us to biological corpora (Futrelle and Gauch 1993). The extension of these techniques to more complex linguistic problems is discussed briefly.

Information retrieval and browsing

In digital libraries, many modes of interaction will be supported. The two major ones will certainly be querying the collection to retrieve information, the classical information retrieval approach (Salton 1989), and browsing by following various types of links. It is necessary to build organization into the collection to directly support these user modes.

Start-up and the steady state

To do linguistic analysis of text, the system must possess a basic vocabulary and knowledge of the structure of English. A corpus of many millions of words contains enough information so that the system should be able to induce the structure and relation of all the important elements of English using extensions of the current methods of corpus linguistics while needing very little human intervention (Charniak 1993). The approach is most successful if a focused domain is used. This initial process of discovery constitutes the *start-up phase* (Futrelle and Zhang 1994). The task is nothing short of bootstrapping the structure of natural language itself, which is daunting but not impossible. The results obtainable by corpus linguistics methods grow more impressive every year, so the goal is worth pursuing vigorously.

After the training corpus is analyzed and the various patterns of English have been induced, they can be used to analyze the new text entering the system. This is the *steady-state regime*. New items and structures will continue to appear in the steady-state, but at a greatly reduced frequency.

During the start-up phase, the types of information about language that need to be captured include:

- word and phrase disambiguation
- domain-specific word use
- domain-specific thesauri for query expansion
- translation of specialized markup (e.g., subscripts and superscripts in chemical nomenclature and numerical forms)
- ontological relations for knowledge frame building

In the steady-state, further information is extracted:

- analysis of internal contents (information distribution among sections and paragraphs)
- clustering of documents based on domain-specific information
- building knowledge frame instances by extracting the specific contents of documents

The corpus linguistic techniques used for these analyses combine statistical analysis and machine learning methods. In the building of frames, knowledge-based methods are needed.

The Linguistic Database

The information build up about words and language structure is stored in a *linguistic database*. This database becomes a permanent part of the digital library so that it can be used to analyze new items that are added to the collection as well as for on-line analysis of user queries and the production of derived objects.

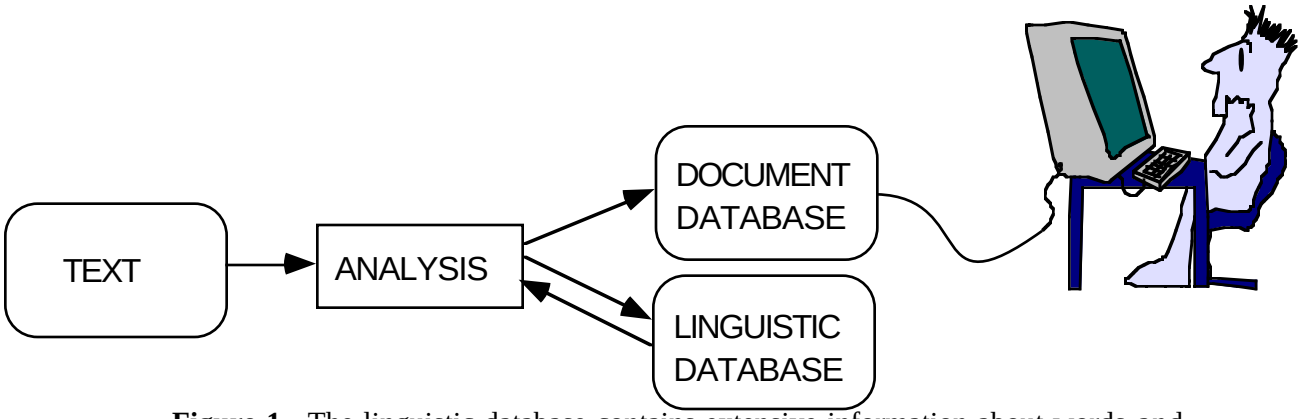

Figure 1. The linguistic database contains extensive information about words and language structure. In the *start-up phase*, the database is built from the initial text presented to the system. In the *steady state*, the database is augmented when new information is encountered and it is used to mediate user interaction. Some of the major uses of the database are to contribute to procedures which analyze the text to index it, derive secondary collections of information and build knowledge frames.

One of the most important parts of the database is the lexicon, which contains statistical, morphological, syntactic, semantic and domain-related information about words. We will discuss the problems involved in trying to automatically build a lexicon and the related thesaurus.

Domain and genre

One of the frustrating things about many current search and retrieval systems for documents is their lack of domain sensitivity; this can markedly reduce the precision of search. A simple example of this is when a word such as "date" is used in a query, information about fruits, times and social relationships is returned. "Date" is a polysemous word; it has multiple senses. Though more complex queries can help narrow the search, there is no direct way to tell such systems that a particular meaning of the word is the one intended. Even though such words are obviously ambiguous, most systems have no mechanism to store or use the different senses, and the texts that are indexed and searched do not distinguish senses either.

There are three aspects of the domain sensitivity problem. The first is to recognize multiple word senses and to tag the corpus text accordingly. The second is to bring together a collection of word senses, identifying them as the vocabulary of a knowledge domain. The third is to employ such knowledge in assisting a user in formulating queries and in executing those queries. For a

discussion of three approaches to word sense disambiguation, see (Gale, Church et al. 1992; Gale, Church et al. 1992).

Domains of knowledge are not crisply delimited, so it is necessary to use measures of appropriateness, not absolute inclusion or exclusion. Furthermore, the vocabulary used for one genre will differ from another, even in the same domain, e.g., an account of a breakthrough in genetics as reported in a newspaper versus a scientific journal. In addition to word senses, the entire style and organization of a document will vary in different genres (Paice and Jones 1993).

Words

Language begins with words. There are a number of characteristics of words that can be usefully exploited in the analysis of corpora. Text presents us with a collection of homographs, words that can initially be distinguished only by their spelling (their orthography). It is useful to distinguish a word as *type*, which is a single entity, e.g., "DNA", versus a word *token* or occurrence, e.g., the many occurrences of the string "DNA" in a text (Lyons 1977).

Words have rich structure and interrelations, including,

Morphology/inflection:	("dog" and "dogs")
Morphology/derivation :	("filter" and "filtration")
Multiple senses:	"stock" (cattle (n), fill larder (v))
Domain-specific senses:	"clone" (computers vs. genes)
Synonyms/antonyms:	"hot" and "cold"
Hyponyms/hypernyms:	"collie" < "dog" < "animal"
Phrasal nouns/verbs:	"New York City", "think up"
Capitalization, punctuation:	"Gene", "gene", "gene," , "gene:"
Abbreviations, acronyms:	"ACM", "VCR"
Complex structures:	"[^{35}S]dATPαS,"
Etymology:	history ("lasing" from "LASER")

One of the primary problems faced by all natural language analysis systems is the problem of resolving ambiguities among the various possible senses of a given word, the problem of *lexical ambiguity resolution* (Allen 1987; Hirst 1987).

Among the various types of information that can aid word classification, one is word morphology, especially for technical text. Stemming, or removing suffixes, is a simple morphological technique for relating words to a common form, approximately their root form. Stemming algorithms can never match the quality of a careful morphological analysis (Ritchie, Russell et al. 1992; Sproat 1992). Because of this, it is useful to expend a large one-time effort to properly compute the morphology of, say, the 500,000 most frequently met terms, build a database of these and never analyze any of them again. Since a good morphological analysis algorithm has a number of heuristics and special cases that it must consult in any event, e.g., relate "used" to "use" but not "need" to "nee", and "mutants" to "mutant" but not "whereas" to "wherea", this argues for the importance of a definitive database which is built, refined, and maintained over time.

Word classification

The most useful classification of words separates them along syntactic and semantic dimensions. In the syntactic domain the primary classes are part-of-speech (noun, verb, adjective, ...) and syntactic category (subject, object, indirect object, ...). Part-of-speech classifiers, "taggers", have been developed to a high degree, both for supervised tagging, with training sets (Church 1988), and in unsupervised mode in which little or no training data is needed (Brill and Marcus 1992; Cutting, Kupiec et al. 1992). In the semantic domain the simplest relations are those between synonyms and between antonyms; other important relations include hyponymy (subclass, superclass) and meronymy (part, whole) (Cruse 1986). Word classification is an important and useful analysis to do, because it assists in,

- Building focused browsing tools
- Developing thesaural expansion for querying
- Bootstrapping higher-order structures

- Query processing

The data presented in Figures 2 and 3 are based on the analysis of a 200,000 word corpus (227,408, to be precise) composed of 1700 abstracts from a specialized field of biology (Futrelle and Gauch 1993). The results shown are subtrees of the full classification tree for the 1,000 most frequent words (covering 80% of all word occurrences in that corpus). A binary clustering algorithm is used, joining two subclusters at each step, where the simplest subcluster is a single word. The method is *unsupervised*, meaning that no training set of correctly classified items is needed in advance. The method, based on (Finch and Chater 1992), is detailed in (Futrelle and Gauch 1993) and can be briefly described as follows:

1. The contexts of each word occurrence are used, the two words immediately preceding and following the word of interest, schematically, the pattern C1,C2,W,C3,C4. The 1,000 highest frequency Ws were studied.

2. The frequencies of the context words appearing in the four positions for the set of W occurrences are totaled separately for each of the four positions. The only context words for which the frequencies were accumulated were the ones that had the 150 highest frequencies in the entire corpus. This resulted in a 4x150 = 600 element vector V of context word frequencies, 1,000 vectors in all.

3. Each element of each vector was rewritten in terms of its mutual information value instead of its raw frequency.

4. The similarities S_{ij} of all pairs of vectors V_i and V_j were computed as the inner product of the mutual information vectors. In (Finch and Chater 1992; Finch and Chater 1993), the raw frequencies were used, but a rank correlation coefficient was used to compute similarities, another way of dealing with the widely varying word occurrence frequencies.

5. The resulting matrix of similarities is subjected to a hierarchical agglomerative clustering analysis. The two words with the greatest similarity are joined first, forming a subcluster. Then that subcluster and all other words are compared and the most similar pair of items is joined, and so on, until the root is reached, containing all words in a binary tree. Portions of the resulting 1,000 node tree are shown in Figures 2 and 3.

In Figures 2 and 3, the similarities quoted refer to the similarities of the last two items joined together, unless otherwise noted. Each subcluster shown is homogeneous for part-of-speech, e.g., nouns tend to cluster with nouns, adjectives with adjectives, etc. But the most striking thing about the data is the tight clustering of semantically related items. The method obviously does not cluster synonyms separately from antonyms, e.g., "higher" and "lower" are tightly clustered. A little reflection reveals why this must be so. The choice of "higher" or "lower" at a given point in the text is not determined by the syntactic context, but instead indicates some knowledge of the world placed there by the author to inform the reader. A reader cannot predict, on the basis of local syntactic context alone, which will occur and therefore there is no indication in the immediately surrounding text as to which will occur. Since there is no such distinguishing information and the local contexts are otherwise the same, the similarity analysis places the words together. For a similar analysis of the lack of pure synonym clusters (and the frequent occurrence of antonyms) see (Grefenstette 1992).

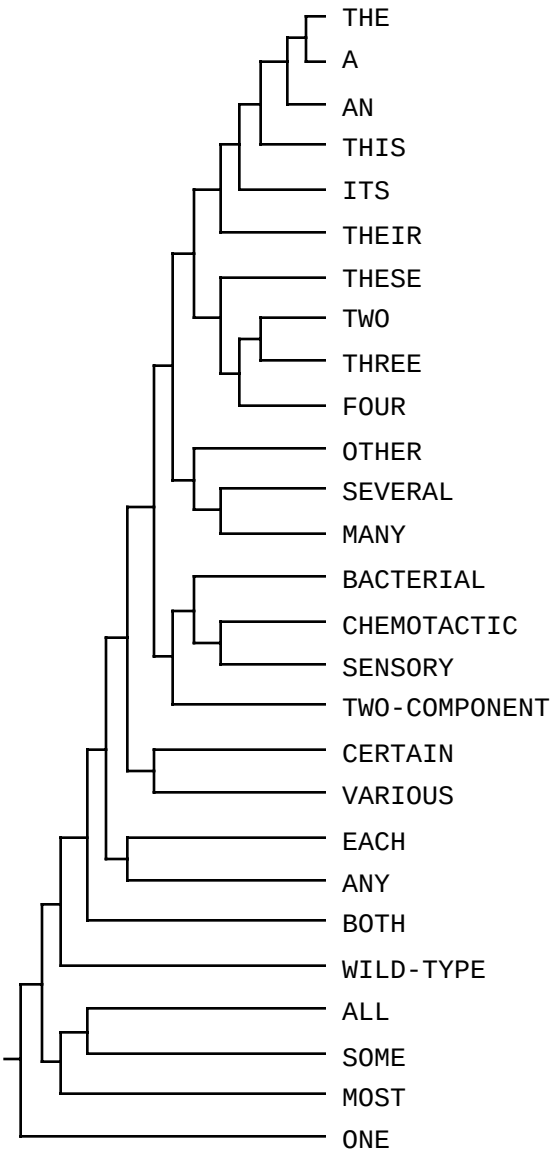

Figure 2A. Cluster containing articles ("A", "AN", "THE") and related modifiers. This is cluster #665 out of 1,000, with similarity 0.298. The "THE/A" cluster contained is #5, with similarity 0.622.

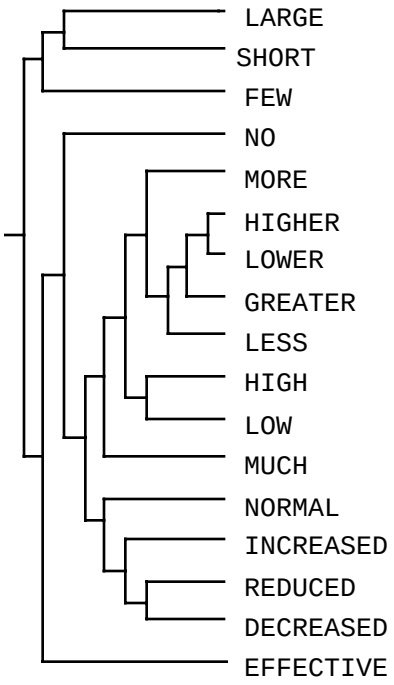

Figure 2B. Cluster containing adjectives describing magnitudes and relative magnitudes. This is cluster #757 with similarity 0.275. Note that some of the -ed items here ("INCREASED", etc.) could also be classified as verbal items (participles), but the algorithm used only allowed them to appear in a single cluster. See Figure 3 for a large collection of -ed forms.

Figure 2. Excerpts from word classification results. The 1,000 highest frequency words in a biological corpus of 200,000 words were clustered using a technique developed by (Finch and Chater 1992; Finch and Chater 1993). Word similarity was computed using the similarity of their contexts, the immediately adjacent words, two on the left and two on the right. The similarity did not take into account any morphological properties of the words. Note the occurrence of some very high-frequency domain-specific terms (bacterial, etc.) in 2A. Details are discussed in the text and in (Futrelle and Gauch 1993).

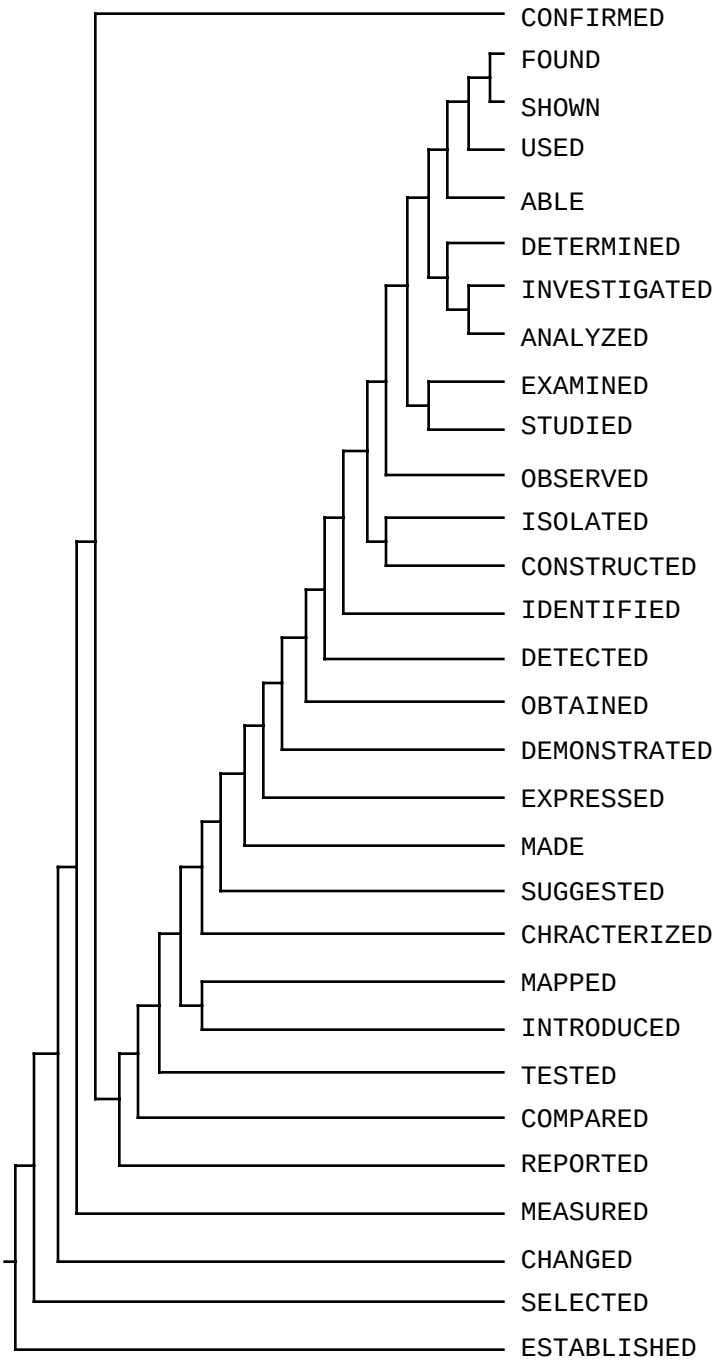

Figure 3A. Cluster #265 containing 30 -ed forms, similarity 0.398. "ABLE" is the most anomalous item in this cluster.

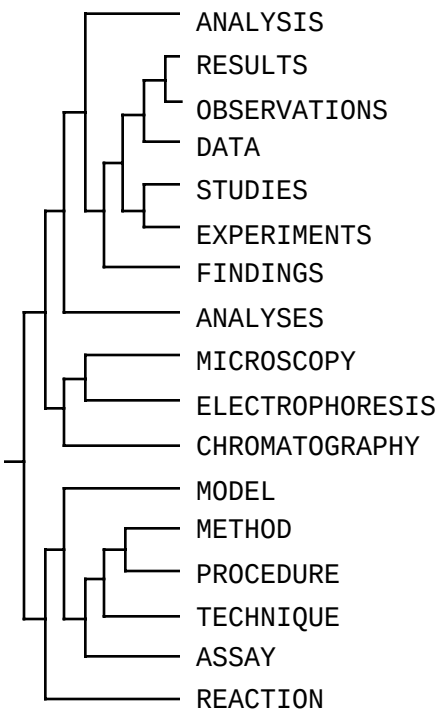

Figure 3B. Cluster #243 , similarity 0.405, containing methods nouns.

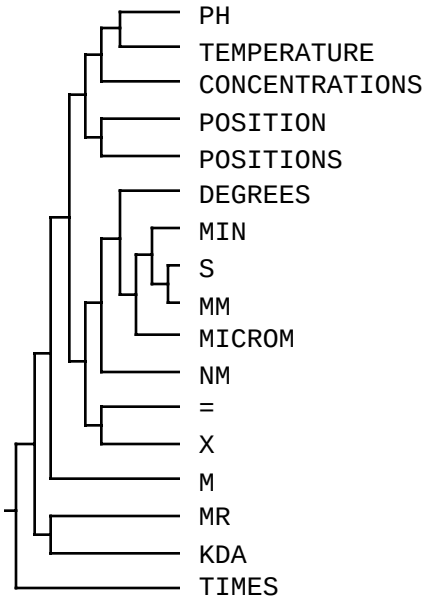

Figure 3C. Cluster #517 containing physical unit designators, similarity 0.332.

Figure 3. Additional clusters computed using the techniques described in Fig. 2.

In our other work, we looked at the classification of single word occurrences. When this is done, the statistical reliability of method drops, because data from multiple occurrences is not initially merged. To improve the reliability, context words were expanded by adding to them words that were found to be similar in the analysis described above. The set of words similar to a base word W, truncated at some minimal similarity, is called the *simset* of W. Using simsets in this way, we successfully classified occurrences of some -ed forms as well as some words that occurred with frequency 1 (Futrelle and Gauch 1993).

Other approaches to unsupervised word classification include (Brill and Marcus 1992) who used a statistical analysis of the distribution of word pairs, (Myaeng and Li 1992) who used coocurrence frequencies in a five-word window (but ignored the exact positions relative to the word of interest) and (Schutze 1992) who uses a vector space model with a very large context, a 1,000 character window.

The classification technique we used has some weaknesses, including,

- The method only works well with word types not tokens. Thus, every occurrence of a word such as "complex" must be classified identically, even though some occurrences (in biology) are nouns referring to a coordinated aggregate of molecules and other occurrences are adjectives meaning "complicated".
- A large similarity matrix must be constructed and then modified as the algorithm executes. The matrix has N(N-1)/2 distinct elements. For N=1,000, this is a half a million elements. Because of this, the method could not be extended to cover all the distinct words in typical corpora which can be hundreds of thousands of items.
- Once the classification is computed, the classes are not then used to characterize the context words, to further refine the classification and render it more self-consistent.

Because of the above difficulties, we have developed and are experimenting with a new approach to classification based on the *principle of minimal complexity* (Futrelle, Zhang and Sekiya,

unpublished). The fundamental assumption of the method is that of Occam's Razor: that subject to certain basic constraints, the simplest assignment of classes to word occurrences is the best assignment. The foundational theory underlying such methods is Kolmogorov complexity (Cover and Thomas 1991; Li and Vitanyi 1993). It is a general approach that can go beyond the standard information and entropy methods. Our method proceeds by searching for class assignments that lead to the minimally complex structural description of the text. One implementation of this approach is being used in our current work on word classification. It begins by assuming a fixed number of classes, $C_1,...,C_n$, and attempts to satisfy the following three constraints simultaneously:

1. Every class C_i should be used to an appreciable degree. (Using only one class would result in an extremely simple but vacuous description.)

2. Each word should only be assigned to a few classes. (If every word occurrence could be assigned to any class freely, then the entire text could be described as a simple structure such as, $C_1,...,C_n,C_1,...,C_n,C_1,...,C_n,...$)

3. The number of distinct patterns should be as small as possible. A "pattern" is typically a sequence of class assignments to k adjacent words, where k is a small integer.

These three constraints are in conflict, so obviously the method has to achieve some balance between them. The method overcomes the three basic limitations of the clustering method described earlier. Because the method is based on optimization of assigned patterns, a variety of approaches can be used to implement it. For one thing, it does not require that the text be processed sequentially, as the much-used Markov models do (Charniak 1993). Those models have to assume that class assignments are based on the left context of a word or the right context, but not both simultaneously. The minimal complexity method also can be used in a supervised learning mode in which the assignments are made initially based on an already classified text (the training set) and subsequent assignments are made with those assignments "frozen" except for new words. The training phase can also be unsupervised, just as the clustering methods are. To jump-start the computations, one could assign class labels consistently to a few frequent and well-known common classes, e.g., placing the articles, "a", "an" and "the" in a single class.

Word sense alignment between independent systems

In the future, there will undoubtedly be many independent digital libraries, all involved in analyzing corpora. When word classification analyses are done, a set of word senses for each word will be generated, but there will be no simple way to find the correspondence between word senses in the distinctly processed collections. For example, if we had two such distinct libraries, each would discover two distinct word senses for "train": train1(railroad) and train2 (educate) (Zernik 1991). On its face, such classification offers no way to find which sense in one library corresponds to which sense in the other. However, if we exchanged simsets of train1 and train2, then the four simsets could be aligned, pairwise, by looking at the overlap of the simsets (using homograph identity only).

Higher-order analysis of language

The multiple senses of individual words creates what is called the *lexical ambiguity* problem. At the sentence level, there are additional problems of *structural ambiguity*. The two primary causes of structural ambiguity in English are prepositional phrase attachment (PP-attachment) and conjunctions. A simple sentence such as, "She drove down the street in her car." is unambiguous for humans, because the prepositional phrase "in her car" is obviously attached to the verb "drove". But it can cause problems for automated systems because of another possible reading in which "street" is located "in her car". Similarly, conjunctions can create ambiguous structures, such as the preferred analysis, "[Pick up the phone and call] or [write a note]." versus "[Pick up the phone] and [call or write a note]."

Problems of structural ambiguity have been thought to be particularly difficult because they apparently depend on world knowledge, considerations that lie well beyond syntactic analysis and the methods of corpus linguistics. But given a large enough corpus, it may be possible to find examples of unambiguous structures to guide disambiguation algorithms. Thus, a number of constructions analogous to "She drove down the street." might be found, but none would be found

of the general form, "The street in her car was wide." Similar approaches can be used to guide the resolution of PP-attachment (Hindle and Rooth 1993).

Determining the structure of sentences is the next level of problems beyond working with words. This is normally attacked by developing grammars that describe language and then using parsing to discover the actual structure of any given sentence. Stochastic grammars are grammars induced from text that embody the preferences found in real text (Charniak 1993). The minimal complexity methods we described earlier can be extended to the problem of discovering grammars, to augment already existing techniques.

More complex needs — knowledge frames

Text is full of information. It would be best if the information could be so well analyzed that it could be reduced to a highly organized and schematic form that could be stored in a database. Then, instead of querying the text, the query could be directed toward the database, a well-structured object for which query methods are well-developed. In corpus linguistics, this process is described as knowledge frame-filling (Rau and Jacobs 1991; Hobbs, Stickel et al. 1993; Jacobs and Rau 1993). The great challenge of all this is that the potentially well-organized information is "encrypted" in natural language text, which must be "decrypted" to discover it. Extracting knowledge frames from text is a difficult problem that will be with us for a long time. It is at best, a method for expository text (not poetry!). We will not discuss this topic further, other than to give some examples of the types of knowledge frames that are of interest in the biological text we study (Figure 4).

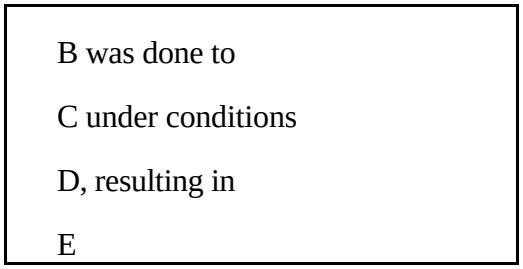

Figure 4A.

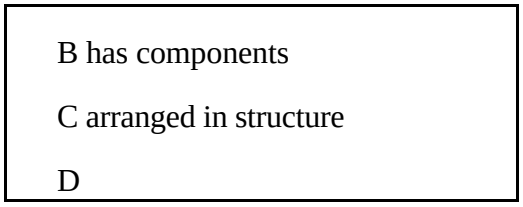

Figure 4B.

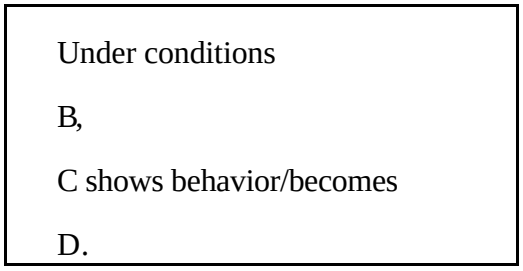

Figure 4C

Figure 4. Three types of knowledge frames of particular use in biology. Knowledge extracted from the text would be used to fill instances of such frames

In most work in this field, knowledge frames are developed manually, but it is possible to develop them at least semi-automatically, once semantic word classes are available and some syntactic analysis of sentences is done. The reason that this is possible is because of the close correspondence between the predicate-argument structure of sentences and the knowledge frames that are ultimately desired. Thus, if we compare Fig. 3A which contains semantically related terms such as "EXAMINED" and "STUDIED", we could analyze the structure of sentences that use them and develop frames such as the one in Fig. 4A, identifying "EXAMINED" with "B was done to", i.e., an examining act was performed.

Derivative objects

Now that full text sources are appearing online, it is desirable to build derivative objects from them. For single documents, there has been progress on automating the construction of abstracts (Rau, Jacobs et al. 1989; Paice 1990; Paice and Jones 1993). But "data mining" techniques can be used to derive another type of derivative document or dataset from a number of related documents. In many information retrieval systems it is possible to rank the documents found in response to a query. On the other hand, suppose that a user of a system wants a list of all the zoos in the United states together with their locations and a brief description of each. Such a list may exist in some specialist publication, but assuming that this was not online, the list could be derived by analyzing a collection of more conventional sources. No ranking of the sources or resulting items in the derived object would be needed or even desired. Building derivative objects can be greatly aided by the corpus analysis approaches that we have described.

Diagrams — Contents and analysis

Diagrams form an important part of the contents of many documents. Important as they are, there is little work done on analyzing diagrams to extract information from them to aid in the structuring and retrieval of particular diagrams or the documents containing them. We have described some of these issues (Futrelle, Kakadiaris et al. 1992) and have more recently succeeded in analyzing some quite complex diagrams [N. Nikolakis, thesis in preparation].

Multidatabases for natural language

In order to be fully integrated into a digital library, documents have to reside in databases of some type. There is a basic mismatch between documents, normally conceived of as flat text, and databases, which by nature are highly structured. There is some work on building text databases, but it is still in its infancy, surprisingly enough (Gonnet, Baeza-Yates et al. 1992; Loeffen 1994). In corpus linguistics a tradition has grown up that treats corpora as large character stream files, processing them with tools such as grep, lex and perl (Baeza-Yates 1992). When text is processed in this way, it is often altered by adding in-line annotations that change the file positions of

elements, making it impossible to build stable indexes pointing into the data. It makes sense to keep the text stable, making annotations to it in separate, updatable structures such as B-trees. Database techniques also allow incremental changes and annotations to be made to a corpus without having to process large files in their entirety. In our work (Futrelle and Zhang 1994) we have used multidatabases (Kim 1993) that employ indexed flat files as well as large persistent arrays in the spirit of relational databases as well as persistent object stores (Bertino and Martino 1993). The demands of large scale text processing are great, so often a new database structure has to be designed when a major new computation is to be done. However, every new structure is tied back to the original corpus through indexes, so the information is cumulative. Our work on databases for use in corpus linguistics analysis is shedding light on database needs for digital libraries, since there is obvious overlap between the two sets of requirements.

Authoring tools for capturing content

One of the problems with all information systems is that so much of what an author knows about a document is lost when the document is created. An author does know in fact, which sense of each word is intended, how ambiguous sentences are meant to be construed, what the components and relations are in a diagram, etc. The quality and organization of a collection in a digital library would clearly be enhanced if more of the author's knowledge could be captured at authoring time and included in the document. Currently, enormous efforts (such as we have been describing) have to be expended to recover the information lost at authoring time. Research is needed to develop ways to overcome this problem.

An author should not be required to write in a radically different "unambiguous" style or to explicitly specify which word senses are intended at each point in the document. Instead, the authoring application should attempt to capture and confirm the information while the author is present to verify the choices made. There are two challenges in building authoring tools that would aid in capturing more detailed information. The first is to make capturing the information as unobtrusive as possible, so as to not interfere with the author's goal of rapidly and flexibly creating a document in a non-distractive environment. The second challenge, and somewhat a corollary of

the first, is to endow the authoring tools with enough intelligence to analyze the author's input and give feedback as to the system's interpretation of it, so the author can quickly confirm or alter the interpretation. As part of this, machine learning techniques can be used to identify favored word interpretations and modes of expression, so that the system would not have to constantly ask the author for verification. Alternatively, verification of the interpretation could be deferred, so as to not distract the author unduly.

Intelligent authoring tools are also needed for graphics and diagrammatic material in papers. They present their own special demands.

Integrating intelligent authoring tools into digital libraries is non-trivial, because of representational issues. Thus, if a particular sense of a word is chosen, how is the chosen sense to be identified to one or more digital library systems which add the document to their collection? (One idea on how to do this was presented in the word sense alignment section earlier.) Representing the resolution of structural ambiguity in a portable format would be even more difficult, but still a worthy goal.

Conclusions

The methods of corpus linguistics can reveal a great deal of information about word use and language structure by careful processing of very large corpora. This information can be used for adding organizational structure to digital libraries both in terms of individual document content and inter-document relations. The structure discovered by corpus linguistics methods reflects the actual use of words and language style in particular domains and genres, rather than being constrained by pre-built categories. The data presented here has demonstrated the power of simple word classification methods for discovering semantically related clusters of word clusters. Work in progress based on the *principle of minimal complexity* overcomes a number of limitations of current classification methods and should discover more detailed and accurate information about word relations and text structure.

Acknowledgments

This work was supported in part by a grant from the National Science Foundation, IRI-9117030, and by a Fulbright Senior Scholar grant (to RPF) for research in the United Kingdom, 1993-94.

References

Allen, J. (1987). Natural Language Understanding. Reading, MA, Benjamin Cummings Publishing Co., Inc.

Baeza-Yates, R. A. (1992). String Searching Methods. Information Retrieval : Data Structures and Algorithms. W. B. Frakes and R. Baeza-Yates. Englewood Cliffs, New Jersey, Prentice Hall**:** 219-240.

Bertino, E. and L. Martino (1993). Object-Oriented Database Systems. Reading, MA, Addison-Wesley.

Brill, E. and M. Marcus (1992). Tagging an Unfamiliar Text with Minimal Human Supervision. AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language (Working Notes), Cambridge, MA.

Charniak, E. (1993). Statistical Language Learning. Cambridge, MA, MIT Press.

Church, K. W. (1988). A Stochastic Parts Program and Noun Parser for Unrestricted Text. Proc. 2nd Conf. on Applied Nat. Lang. Processing, Austin, TX, Assoc. for Computational Linguistics.

Church, K. W. and R. L. Mercer (1993). "Introduction to the Special Issue on Computational Linguistics Using large Corpora." Computational Linguistics **19**(1): 1-24.

Cover, T. M. and J. A. Thomas (1991). Elements of information theory. New York, Wiley.

Cruse, D. A. (1986). Lexical Semantics. Cambridge, UK, Cambridge U. Press.

Cutting, D., J. Kupiec, et al. (1992). A Practical Part-of-Speech Tagger. Proc. 3rd Conf. on Applied Natural Language Processing, Assoc. for Computational Linguistics.

Finch, S. and N. Chater (1992). Bootstrapping Syntactic Categories Using Statistical Methods. Proc. 1st SHOE Workshop, Tilburg U., The Netherlands.

Finch, S. and N. Chater (1993). Learning Syntactic Categories: A Statistical Approach. Neurodynamics and Psychology. M. Oaksford and G. Brown. San Diego, CA, Academic Press**:** 295-319.

Futrelle, R. P. and S. Gauch (1993). Experiments in syntactic and semantic classification and disambiguation using bootstrapping. Acquisition of Lexical Knowledge from Text, Columbus, OH, Assoc. Computational Linguistics.

Futrelle, R. P., I. A. Kakadiaris, et al. (1992). "Understanding Diagrams in Technical Documents." IEEE Computer **25**(7): 75-78.

Futrelle, R. P. and X. Zhang (1994). Large-Scale Persistent Object Systems for Corpus Linguistics and Information Retrieval. Digital Libraries '94, College Station, Texas, Dept. of Computer Science, Texas A&M University.

Gale, W. A., K. W. Church, et al. (1992). "A Method for Disambiguating Word Senses in a Large Corpus." Computers and the humanities **26**(5): 415-.

Gale, W. A., K. W. Church, et al. (1992). Work on Statistical Methods for Word Sense Disambiguation. AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language (Working Notes), Cambridge, MA.

Gonnet, G. H., R. A. Baeza-Yates, et al. (1992). New Indices for Text: PAT Trees and PAT Arrays. Information Retrieval. W. B. Frakes and R. Baeza-Yates, Prentice Hall**:** 66-82.

Grefenstette, G. (1992). Finding Semantic Similarity in Raw Text: the Deese Antonyms. AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language (Working Notes), Cambridge, MA.

Hindle, D. and M. Rooth (1993). "Structural Ambiguity and Lexical Relations." Computational Linguistics **19**(1): 103-120.

Hirst, G. (1987). Semantic interpretation and the resolution of ambiguity. Cambridge, Cambridge University Press.

Hobbs, J. R., M. E. Stickel, et al. (1993). "Interpretation as abduction." Artificial intelligence **63**(October 1993): 69-142.

Jacobs, P. and L. F. Rau (1993). "Innovations in text interpretation." Artificial Intelligence **63**(October 1993): 143-191.

Kim, W. (1993). Object-Oriented Database Systems: Promises, Reality, and Future. International Conference on Very Large Data Bases, Dublin, Ireland.

Li, M. and P. Vitanyi (1993). An Introduction to Kolmogorov Complexity and Its Applications. New York, Springer-Verlag.

Loeffen, A. (1994). "Text Databases: A Survey of Text Models and Systems." SIGMOD RECORD **23**(1): 97-106.

Lunin, L. F. and E. A. Fox (1993). "Perspectives on ... Digital Libraries: Introduction and Overview." Journal of American Society for Information Science **44**(8): 441-445.

Lyons, J. (1977). Semantics, Cambridge University Press.

Myaeng, S. H. and M. Li (1992). Building Term Clusters by Acquiring Lexical Semantics from a Corpus. CIKM-92, Baltimore, MD, ISMM.

Paice, C. D. (1990). "Constructing literature abstracts by computer: Techniques and prospects." Information Processing & Management **26**(1): 171-186.

Paice, C. D. and P. A. Jones (1993). The Identification of important concepts in highly structured technical reports. SIGIR '93, Pittsburgh, PA, ACM.

Rau, L. F. and P. S. Jacobs (1991). Creating Segmented Databases From Free Text for Text Retrieval. Proc. 14th Ann. Inter'l ACMSIGIR Conf. on R&D in Information Retrieval, Chicago, IL, ACM Press.

Rau, L. F., P. S. Jacobs, et al. (1989). "Information extraction and text summarization using linguistic knowledge acquisition." Information Processing & Management **25**(4): 419-428.

Ritchie, G. D., G. J. Russell, et al. (1992). Computational Morphology - Practical Mechanisms for the English Lexicon. Cambridge, MA, The MIT Press.

Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Reading, MA, Addison-Wesley Pub. Co.

Schnase, J. L., J. J. Leggett, et al., Eds. (1994). Proceedings of Digital Libraries '94. The First Annual Conference on the Theory and Practice and Digital Libraries. College Station, Texas.

Schutze, H. (1992). Context Space. AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language (Working Notes), Cambridge, MA.

Sproat, R. (1992). Morphology and Computation. Cambridge, MA, The MIT Press.

Zernik, U. (1991). Train1 vs. Train2: Tagging Word Senses in Corpus. Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon. U. Zernik. Hillsdale, New Jersey, Lawrence Erlbaum Associates, Publishers: 97-112.