A System for Discovering Relationships by Feature Extraction from Text Databases [1]

Jack G. Conrad
West Publishing Company
St. Paul, MN 55164

conrad@research.westlaw.com

Mary Hunter Utt
Digital Equipment Corporation
Littleton, MA 01460

utt@netcur.enet.dec.com

Abstract

A method for accessing text-based information using domain-specific features rather than documents alone is presented. The basis of this approach is the ability to automatically extract features from large text databases, and identify statistically significant relationships or associations between those features. The techniques supporting this approach are discussed, and examples from an application using these techniques, named the *Associations System*, are illustrated using the *Wall Street Journal* database. In this particular application, the features extracted are company and person names. The series of tests run on the *Associations System* demonstrate that feature extraction can be quite accurate, and that the relationships generated are reliable. In addition to conventional measures of recall and precision, evaluation measures are currently being studied which will indicate the *usefulness* of the relationships identified, in various domain-specific contexts.

1 Introduction

Information retrieval systems have traditionally been *document-oriented*; that is, IR systems have been designed to store, retrieve, and display text documents such as newspaper articles or legal summaries. Furthermore, many of today's hypertext systems have inherited this paradigm of information representation to the extent that hypertext nodes are typically short text documents, possibly derived from longer sources. In some instances, browsing of index terms associated with the text documents is also supported [1, 2], but this is generally regarded as a secondary activity in relation to the primary task of identifying relevant documents (or text nodes). From the user's standpoint, however, it is usually the information contained in the text documents that is the goal of the search, not the documents themselves.

In some application domains, the target information is well-defined, for example, financial figures, transaction dates, product types, etc. In these domains, it may be possible to construct information retrieval and hypertext-like browsing systems based on the internal information rather than exclusively on the text documents that embody it. As a result, systems like these should be able to answer important categories of queries and support alternative means of access currently impossible with standard document-based systems. For example, a traditional text retrieval system could not be expected to satisfy a real-time query which requests a list of companies with which Ross Perot had business dealings in 1988. By contrast, a feature-based system could.

The techniques described in this paper have formed the basis for the *Associations System*. This implementation is an information retrieval system which pursues a concept-oriented rather than document-oriented approach; it focuses on the recognition of domain-specific features in a textual database and relationships identified between those features.

In the following sections, we describe techniques and experiments in three major areas needed to support this application:

- *Automatic feature extraction* - techniques used to recognize features in large, free-text databases.
- *Generating direct links* - techniques for quantifying the relationship between features based on their association in free text.
- *Generating indirect links* - techniques for indexing features and identifying indirect relations between features based on shared classifications, as well as offering possible starting points for browsing in a feature network.

[1] This research was performed at the Center for Intelligent Information Retrieval at the University of Massachusetts at Amherst.

Our experiments with the company and person name recognizers use a database of one year (1987) of *Wall Street Journal* articles. It consists of 46,449 articles containing 249 words on average, and is a part of the TIPSTER document collection [3]. Subsequent recall and precision experiments with the *Associations System* use as a database a more recent year (1991) of *Wall Street Journal* articles. It contains 42,652 articles averaging 232 words each, also from the TIPSTER collection. We have found that evaluating some of these proposed techniques is more difficult than a typical information retrieval experiment, and this issue will be discussed in the sections that follow.

2 Automatic Feature Extraction

The problem of feature or fact extraction from unrestricted text has been studied by a number of researchers in the context of the Message Understanding Conferences [4] and the TIPSTER project [3]. The basic approach has been to use a variety of natural language processing and statistical techniques to extract predetermined types of facts for a specific domain. In the TIPSTER joint venture domain, for example, the goal of extraction is to identify information such as the companies forming the joint venture, the name of the new company, the location of the new company, the products of the new company, and the amount of money involved.

Accurate extraction of some types of information requires either sophisticated analysis or significant amounts of training data. There are, however, a number of important and fairly general features which can be recognized using relatively simple techniques. These include the names of companies, the names of people, locations, monetary amounts, and dates. The task of collecting this information could be described as the recognition and categorization of certain noun phrases. In other words, a feature is essentially an object which falls into a special word grouping and has certain attributes associated with it [5]. High rates of accuracy are possible because of the relative simplicity of the task. It is, for example, much easier to recognize the presence of a company name in an article about a joint venture than to identify the role that company is playing. The ability to recognize these simple features can be used to develop powerful new approaches to accessing information [6].

For the application we address in this paper, the two feature recognizers required are for company names and person names. The techniques used for these feature recognizers involve a combination of lexical scanners built using *lex* [7], or a similar tool, and table lookup.

2.1 The Company Name Recognizer

The company name recognizer scans the text for proper nouns (capitalized words) that have the appropriate format for a company name. Company names often include special words such as *Inc.*, *Corporation*, or *Pty. Ltd.* that are particularly useful for recognition [8]. In a given document, the company name recognizer will use these special words to recognize the first mention of a company name and store it in a temporary table. This table permits the recognition of subsequent uses of that company name, even if the special words are not used. In newspaper articles, for example, the first use of a company name in a story generally uses the full form.

In a simple test of the company name recognizer, we applied it to a sample of the *Wall Street Journal* database and compared the results to company names identified manually. The test database consisted of 139 articles containing 29,000 words. The manual scan of the database identified 334 company names. In this test, the precision (percentage of names identified as companies that actually were companies) was 89% and the recall (percentage of company names in the sample that were identified as companies) was 79%.[2] Many of the precision errors were caused by two difficult company name formats where names are combined using 'and' and 'of', such as in *X, Y and Z Corporation* and *X of Y Inc.* Although these can be valid formats (e.g., Mutual of Omaha), they tend to introduce too many errors. We are currently revising the company name recognizer to improve recall by introducing a company name table that will contain common names and synonyms (e.g., for American Telephone & Telegraph/AT & T and Digital Equipment/DEC). This modification is based on the observation that references to well-known companies are more likely not to use the full form of the name.

[2] The tests performed were name-based rather than occurrence-based.

2.2 The Person Name Recognizer

Many application-dependent approaches to personal name identification have been developed over the last three decades [9]. Because of our application domain, the person name recognizer relies upon techniques similar to the above, but places more emphasis on table lookup. A name is recognized when a capitalized sequence of words begins with a title such as *Ms.*, *Chairman*, *President*, and so forth. In addition, lists of first names and last names are used to identify names that do not contain titles. As in the case of the company name recognizer, subsequent references to people in the same story are recognized, even if the full name is not used. Checks are made to ensure that a recognized name is not a company name (for example, that L. L. Bean Inc., referred to later in a story as "L. L. Bean" is not recognized as a person name).

Finally, because sequences of capitalized words may contain other words in addition to a name, a stop list of common problem words is maintained. For example, the *Wall Street Journal* frequently begins sentences with constructions such as *Added Joe Smith ...* or *Investor Jane Doe ...* We are investigating whether dictionary lookup may be more effective (and general) for this purpose.

To test the person name recognizer, the same database used for the evaluation of the company name recognizer was used. The manual scan identified 269 person names in the sample text. The person name recognizer achieved 92% precision and 93% recall. Many of the errors were due to unsuitable names in the first and last name tables. As an example, we are modifying the recognizer to avoid identifying locations (e.g., Santa Monica, Carson City) as person names.

Recognizing synonymous names is also a problem. For example, Bill Clinton and William Clinton are currently recognized as two different people. For two names in different stories to be recognized as the same, we have specified that the first name, middle initial (if any) and last name have to be the same. Given that we want to make connections between people and companies, the resolution of variant names needs to be addressed. A table of common synonyms will help, but it is not the complete solution. Fortunately, the company-person connections themselves can provide significant additional evidence that two names are the same. If, for example, Roger Smith and Roger B. Smith are both highly correlated with GM, it is likely that they are the same person. This technique is used to conflate person-company links after they are generated using the techniques described in the next section. Despite the complexities involved in recognizing company and person names, our experiments show that these names can be reliably recognized in textual databases.

3 Generating Direct Links

Having identified references to companies and people in the text, the next step is to identify relationships or associations between them. These associations can be used as the basis for the links in a hypertext network of companies, people, and source texts. Associations can be identified using either direct or indirect associations. A direct association occurs when the company and/or person names occur 'close' to each other in the text. Closeness can be measured either by a simple distance measure (how far apart the names are) or using some linguistic context (for example, in the same sentence or in a subject-object relationship). In previous experiments [10], word distance has been shown to be the strongest evidence for the presence of phrasal relationships, so for this study we have concentrated on name distance as the primary measure of association.

By contrast, an indirect association occurs when the company and person names have similar types of words associated with them. In the next section, we discuss how word contexts can be used to derive these indirect associations and support the additional retrieval of companies and people.

In the direct associations study, we used text *windows* to define direct links. A window refers to the number of words on either side of a target feature (i.e., company or person name). For example, a window of size 11 would include the target feature, the five preceding words, and the five succeeding words. The window sizes used here were 51 words and 201 words. These sizes were chosen empirically, and roughly approximate average paragraph-level and document-level associations.

The strength of association between two features depends on the number of associations or co-occurrences in a text window and how common the features are in the whole database. For example, a single co-occurrence of GM and IBM in a text window is not likely to be significant, given how frequently these companies are mentioned in the database. To determine significant associations (or to rank the relationships found by presumed importance), we use two statistical measures, the expected mutual information measure (*EMIM*) [11] and phi-squared (ϕ^2) [12].

The expected mutual information measure compares the probability of observing two features, x and y, together to the probability of observing the two features independently. In this paper, we use a simplified version of this measure that ignores terms involving probabilities that features do not occur. The measure used is:

$$EMIM(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

When a strong relationship exists between the features, the joint probability (P(x,y)) will be greater than chance and $EMIM(x,y)$ will be greater than 0.

The calculation of both *EMIM* and ϕ^2 makes use of a contingency table. This table can be represented as follows:

	y	$\bar{y}$
x	a	b
$\bar{x}$	c	d

The upper-left-hand cell [a] records the number of times features x and y co-occur in a window. Cell [b] records the number of times x occurs but y does not. Similarly, cell [c] records the number of times y occurs but x does not. Finally, cell [d] records the number of times neither feature occurs. Given this table to estimate probabilities, the *EMIM* calculation is:

$$EMIM(x,y) = \log_2 \frac{a(a+b+c+d)}{(a+b)(a+c)}$$

The ϕ^2 measure has been suggested as an alternative to *EMIM* and will tend to favor high-frequency events more. This is calculated as follows:

$$\phi^2 = \frac{(ad-bc)^2}{(a+b)(a+c)(b+d)(c+d)}, \text{ where } 0 \leq \phi^2 \leq 1$$

Feature Recognizer Window Experiments

The two measures described were used to compute associations between all features (companies and people) that co-occurred in text windows for the 1987 *Wall Street Journal* collection (Table 1).

Two different window sizes were used. For window size 51, there were 226,475 windows. For size 201, there were 54,464 windows. Since the second window size is close to the average size of a document, the number of windows is similar to the number of documents in the database.

Table 2 lists the number of pairs of features (n) that co-occurred m times within windows of size 51 and 201 respectively, for $m = 1$ to 10. The tables also list the maximum number of co-occurrences.

Evaluating the associations produced is difficult. The top ranked associations for *EMIM* and ϕ^2 appear to be very similar. For example, 58 of the top 68 ϕ^2 scores for company-person associations are also among the top 68 *EMIM* scores. When the top ranked associations are examined for a given company or person for either *EMIM* or ϕ^2, one observes that, although there are not large differences between the measures, the ϕ^2 measure does seem to favor high frequency associations. Since this often corresponds with intuition about associations, the ϕ^2 measure is probably the better choice.

The question remains, however, just how reliable or useful these discovered relationships are. It is not clear whether it would be possible to develop a list of "relevant associations" as is done with relevant documents in IR tests. Certainly, it is clear that the approach works to the extent that obvious relationships are found (such as between the CEO of a company and the company) and that all the relationships discovered have a basis in the text documents. Spurious relationships only occur at low frequency co-occurrences (mainly $m = 1$) and with larger window sizes. The data for the window size

Database:	WSJ87	WSJ91
Number of words indexed in the collection:	11,550,222	9,885,466
Number of person names identified by recognizer:	45,285	38,058
Number of company names identified by recognizer:	25,816	30,615

Table 1: Statistical Information for the 1987 and 1991 *Wall Street Journal* databases (WSJ87 & WSJ91)

Number of times (m) a pair of features co-occurred	Number of pairs of terms (n) that co-occurred m times					
	Window size = 51			Window size = 201		
	person-company and company-person	person-person	company-company	person-company and company-person	person-person	company-company
1	74,950	81,890	102,404	135,112	129,430	197,642
2	28,880	28,924	20,622	60,902	65,382	36,442
3	11,084	11,654	9,206	23,584	26,944	15,112
4	6,336	7,264	4,780	16,576	22,122	9,646
5	3,512	3,790	2,534	8,204	9,898	5,616
6	2,618	2,776	1,774	7,960	10,600	4,986
7	1,740	1,908	1,182	4,544	5,410	3,228
8	1,362	1,434	916	4,572	5,390	2,824
9	1,118	992	670	2,890	3,578	2,134
10	852	720	562	2,632	3,078	1,626
maximum value of m	149	145	150	200	200	199

Table 2: WSJ87 Co-occurrence Data

of 201 shows that many more co-occurrences are found. Some of these will be useful and some will be chance co-occurrences. The most useful window size will probably depend on the application and the users.

As an indication of the type of relationships discovered, the following are results from our *Associations System* graphical user interface.[3] It is presented as an illustration of how our techniques might be integrated into a single application. Another reason we show it here is as a means of examining whether the relations found are reasonable.

From the system's entry window, the user has the option of selecting the statistical measure of association (*EMIM* or ϕ^2) and setting the co-occurrence threshold.[4] For example, if a user is only interested in names which co-occur ten or more times, then the co-occurrence threshold would be set to 10. The default values for these parameters are ϕ^2 and 2 respectively.

The first interaction shows the companies that are associated with "Donald Trump" (Figure 1). There are four regions of the screen: *Query Entries*, *Selections* (user options), *The Listings* (containing the output produced by the query), and *History*. The first column of the output listing shown contains the association measures (ϕ^2 in this case); the second column contains the frequency of co-occurrence. Note the high degree of association between "Donald Trump" and "Golden Nugget" (a hotel-casino owed by Mr. Trump). In the second interaction (Figure 2), we see additional individuals associated with the "Golden Nugget" enterprise, one of them of course being "Donald Trump," but the list also points in new directions. This second list also illustrates how the ϕ^2 association measure tends to favor higher frequency relations. From this juncture, the user has the choice of either continuing to browse through the feature network or calling up relevant documents—by using a pair of related names in a structured query, for instance.

4 Generating Indirect Links and Entry Points for Browsing

In the last section, company names and person names were used as initial queries for the system, and the associations generated were based on direct links identified by way of the co-occurrence of the features in the documents. It is also possible to develop a more general approach to identifying associations between

[3] Our GUI is constructed using the *Tcl* interpretive command language and its X toolkit, *Tk* [13].

[4] Because the *Associations System* has been integrated with *INQUERY* [6], the user can also specify the number of documents to retrieve, as well as the size of its proximity operator (e.g., for paired names used in structured queries).

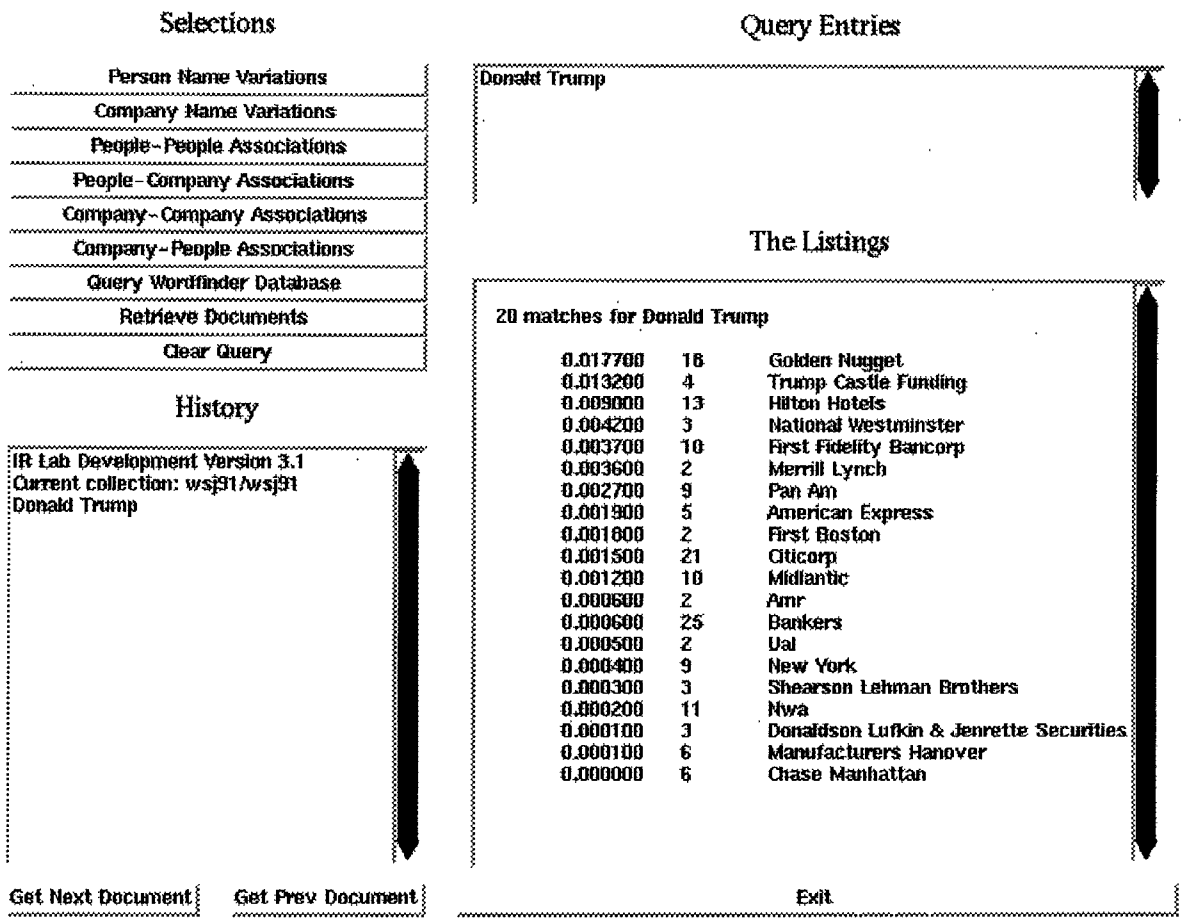

Figure 1: Companies associated with Donald Trump

features, by means of a search based on shared "attributes." This same general approach can be used to provide entry points into the system's feature network.

The basic technique involves identifying all the paragraphs in the entire database in which a given feature occurs. For each feature, a pseudo-document is created consisting of all of the paragraphs in the original database in which that feature is found. For example, if *Transamerica* (Corp.) is mentioned in 35 paragraphs in the database, the words in those paragraphs are used to create a pseudo-document for *Transamerica*. In this sense, the feature itself is treated as a special type of document and a "feature database" can be created, analogous to a document database. This approach and its resultant pseudo-documents are thus distinct from current research efforts which use local content when retrieving from large text files [14]. The types of linkages generated from this component of the system are different from normal hypertext linkages (which usually relate small text segments and, in our case, the direct links discussed above). These indirect links have provided our system with a powerful auxiliary tool, not only in terms of identifying entry points, but also in supplementing discoveries from direct-link retrieval with other new findings. In the experiments described here, the *INQUERY* text retrieval system [6] was used to store feature representations and to retrieve features.

The words that are used to represent features are all the non-stopwords in the paragraphs, weighted by their frequency of occurrence in the paragraphs and in the database as a whole. Currently, no limit is put on the number of words in the representation of a feature. Frequently occurring people and companies, therefore, may have thousands of words in their representations. It is an open research issue to determine if there are optimal representation sizes and whether the entire database or some subset of it should be used to derive these representations.

Once a feature database has been generated, at least two query mode options exist. In the first case, another feature (i.e., person or company) is used as a query, thus accessing features with similar representations. In this manner, *indirect* company-company, person-person, or company-person relations

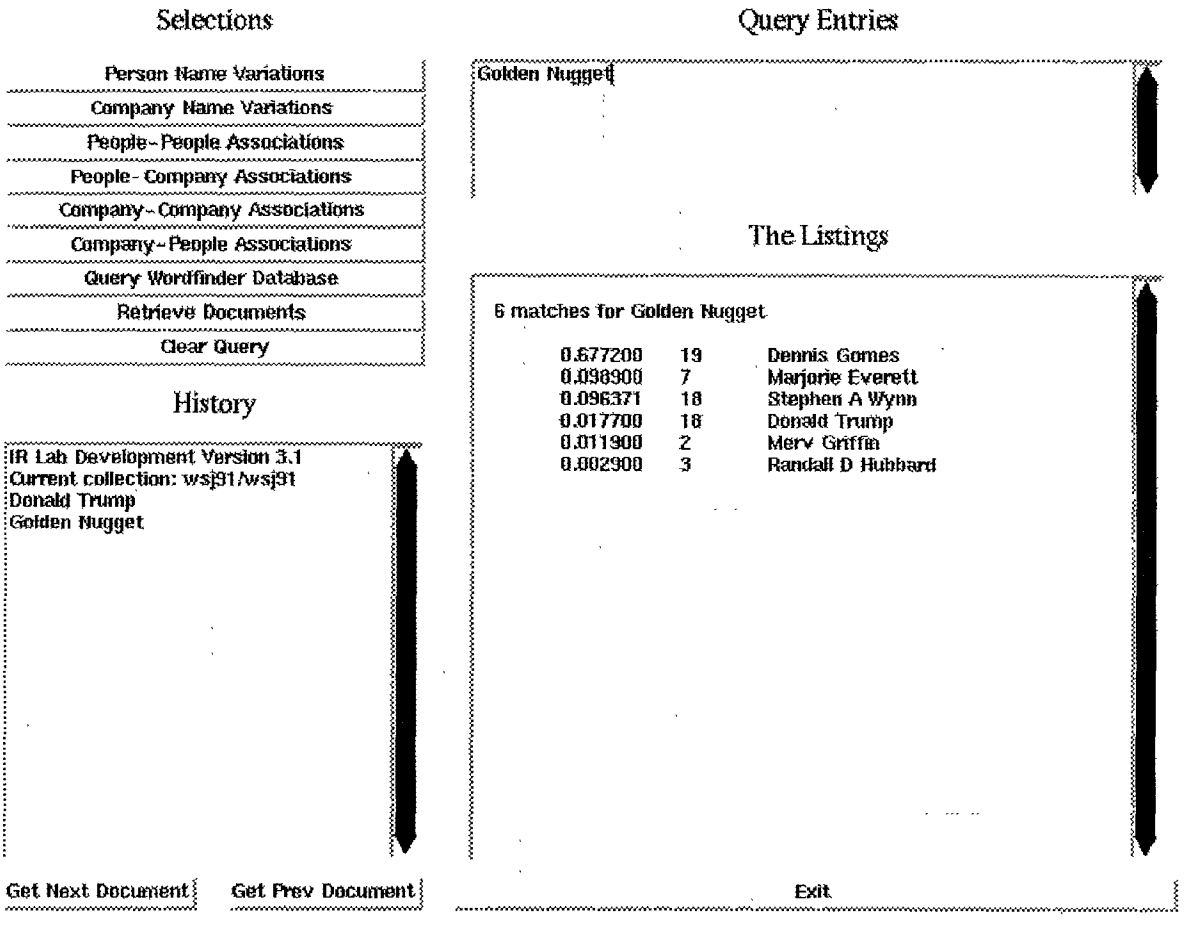

Figure 2: Persons associated with Golden Nugget (Hotel-Casino)

can be determined. Such relations could be requested by a user at any point during the user's interaction with the system.

The second mode corresponds to the use of natural language queries. Such an option provides the user with a convenient means of establishing a suitable entry point to the "feature network" at the beginning of an information search session. Natural language queries could be invoked by the feature retrieval system (in our case, *INQUERY*), and ranked lists of company and person names would be produced. The features retrieved would be those whose representations contain the words in the query with high weights. We could, for example, ask for "companies or people active in biomedical research and technology" and retrieve companies and individuals whose names have co-occurred frequently with the phrases such as "biomedical research" or "biomedical technology" in paragraphs in the database.

A feature retrieval system was generated for the WSJ92 database (with 2,275,134 indexed words) using the *INQUERY* system. The features retrieved for test cases were reasonable, based on the results of our precision experiments. That is, the company and person names retrieved using natural language queries were judged to be relevant to the topics mentioned in the queries. The output of this component of the system, however, was more difficult to evaluate for performance. More rigorous evaluation would require the participation of users in particular application domains.

Figure 3 provides an illustration of the operation of the feature retrieval system. Note that the format of the output of this system is quite different from the direct-link *Associations* program shown above. The first query is an attempt to find people and companies who are associated with the biomedical field. The first number on each line is the rank, the second is the similarity to the query, and the numbers in parentheses after the feature are related to its frequency of occurrence in the database. Note that this list contains both companies (or "enterprises") and people.

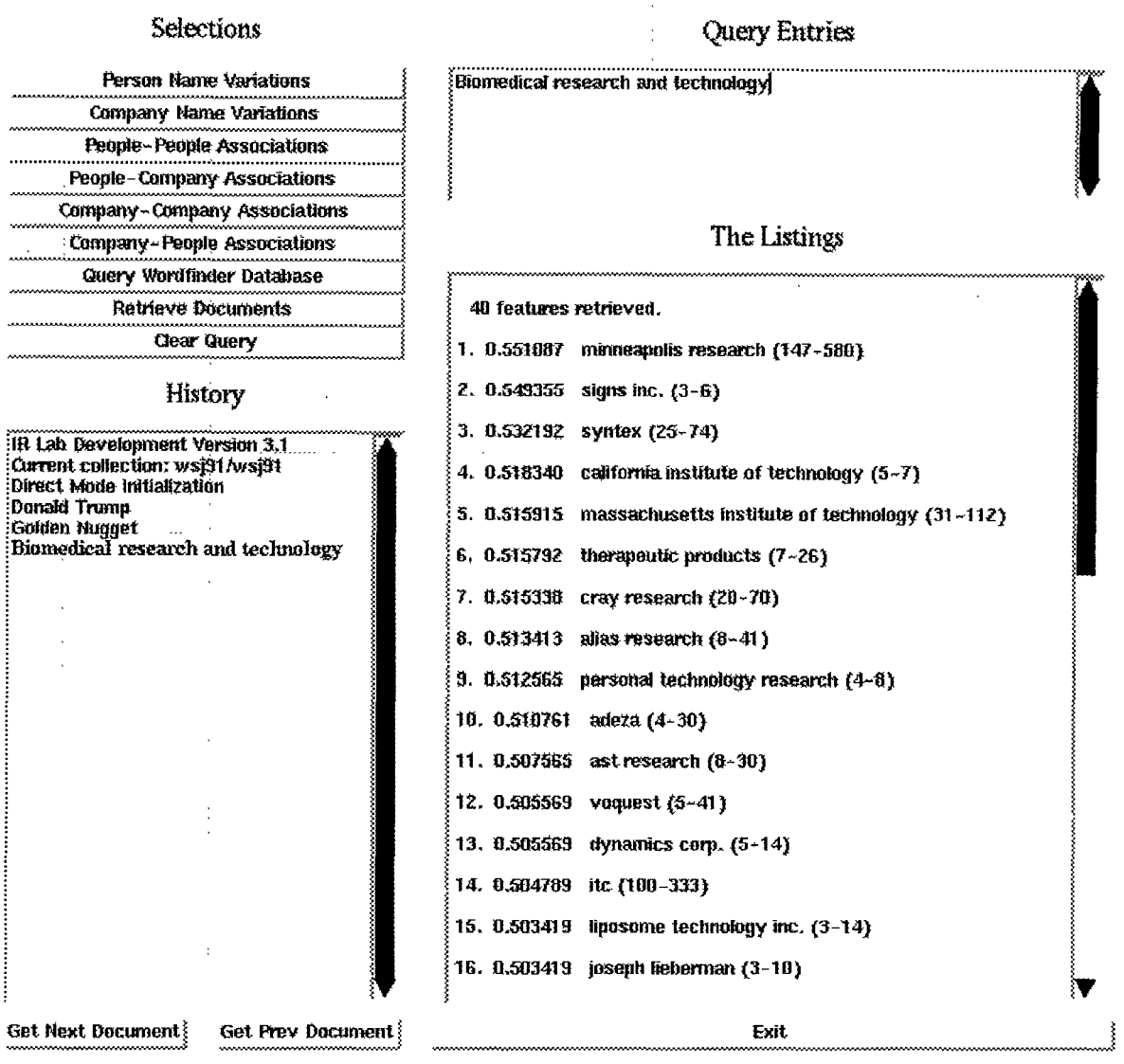

Figure 3: Enterprises and Persons associated with Biomedical research & technology

5 Development of a Feature-based System like *Associations*

The following is a summary of the processes we have developed for a system based on extracted features and relationships.

- Direct-link Generation
  1. Identify (i.e., index) features while parsing the document database. Record the position of each feature in the document/database.
  2. Compute the statistical measure of association (e.g., EMIM or ϕ^2) for direct (proximity-based) links between each of the features identified in a reasonably select co-occurrence window.
  3. Retrieve pairs of strongly associated features in response to queries consisting of features in the same domain.
- Indirect-link Generation
  1. Index all non-stopwords while parsing the document database.
  2. Generate a "feature database" consisting of pseudo-documents created using the words which are indexed near the given feature in the original database.

3. Harness a retrieval engine to provide ranked features in response to a variety of queries (e.g., either features themselves or natural language).

- System Integration—Integrate the above functionality, along with a conventional document retrieval system, to produce a hypertext-like feature browser and text retrieval system.
- Simultaneous Processing—Portions of the first two tasks above (e.g., parsing) can be performed simultaneously.

Note that given a stable parsing vehicle, these steps are free from any particular domain dependencies.

In our implementation of the *Associations System*, we made a significant effort to optimize our processing of those steps which were initially computationally expensive. We can currently parse over 100 megabytes of *Wall Street Journal* text in approximately 30 minutes using a Sun 490. This is roughly equivalent to a database the size of WSJ88, which contains 39,906 articles. The computation for *Associations*' statistical measures for WSJ87, containing 46,449 articles, took approximately 4 hours. We are currently focusing on the next phase of our system development, which is to to generate one large composite *Associations* database consisting of all six years of the TIPSTER *Wall Street Journal* collection.

System Experiments

Precision Tests—To answer the question of whether the associations generated are correct, 15 company names and 15 person names from a variety of fields were chosen from the WSJ91 database (Table 1). Both company and person matches generated by the *Associations System* were recorded. Using the *INQUERY* retrieval engine, documents were then retrieved and the correctness of the relations produced was corroborated (i.e., either Yes or No).[5] In the cases studied, representing 286 company matches and 262 person matches, a precision figure of 90% was found for both categories of matches (i.e., for company names and person names), increasing to 95% when feature recognizer errors were removed (Tables 3 and 4).

In using the *Wall Street Journal* database, we have observed that in some cases up to one half of the features associated with a given company or person fall into the categories of either investment firms or their financial analysts. Whether or not these types of relations are useful depends on the information requirements of the user. As an implementation decision, some of these "financial" features could be filtered from the output, since the set of such firms is limited and identifiable. In either case, we acknowledge the need for domain-specific users to provide their assessments of the "usefulness" of these discovered relationships.

	Matches w/ Persons	%	Matches w/ Companies	%	Combined Matches	%
Initial Results	47/58	81.0%	226/249	90.8%	273/307	88.9%
Compensating for Recognizer Error	47/47	100%	226/239	94.6%	273/286	95.5%

Table 3: Precision Values for Company Name Matches

	Matches w/ Persons	%	Matches w/ Companies	%	Combined Matches	%
Initial Results	200/221	90.5%	53/53	100%	253/274	92.3%
Compensating for Recognizer Error	200/209	95.7%	53/53	100%	253/262	96.6%

Table 4: Precision Values for Person Name Matches

[5] The relevance judgements here were made by the authors.

Recall Tests—To determine whether the system misses any relations, two companies (Revlon Inc. and William Blair & Co.) and two business figures (Charles E. Exley [of NCR] and Philip E. Benton [of Ford]) were chosen. These names were also used in the precision tests above. Again using *INQUERY* as the retrieval engine, documents were retrieved, using each of these four names as queries, and a list of related persons and companies was generated for each.[6] The *Associations System* produced 49 company matches and 29 person matches for these same names. The results produced by the system and by manual verification do not include one time only co-occurrences which as often as not are due to chance (Tables 5 and 6). In addition to the observations cited earlier, we found that another reason some companies are not recognized is because they may never occur in the database with a suffix indicating their corporate status. This finding suggests that a correct combination of company name recognizer and lookup capability will produce a system which is not only more thorough in terms of recall, but more domain independent as well.

	Documents Retrieved	Matches w/ Persons	%	Matches w/ Companies	%	Combined Matches	%
Revlon Inc.	65	7/11	64%	19/28	68%	26/39	67%
William Blair & Co.	39	1/1	100%	19/26	73%	20/27	74%

Table 5: Recall Values for Company Name Matches

	Documents Retrieved	Matches w/ Persons	%	Matches w/ Companies	%	Combined Matches	%
Philip E. Benton	12	6/7	86%	4/6	67%	10/13	77%
Charles E. Exley	34	15/19	79%	7/17	41%	22/36	61%

Table 6: Recall Values for Person Name Matches

It is not completely clear how one would proceed to evaluate thoroughly a system such as *Associations*. The tests we have performed to date are preliminary. The evaluations presented above serve to illustrate some of the types of experiments we believe are essential for an initial assessment. A number of our results have been promising, especially with respect to precision, and encourage further refinement and enhancement of these techniques. An additional evaluation is currently being investigated, beyond precision and recall, which provides a measure of output *usefulness*. It is a parameter which is client-need dependent, and we think this is a part of what is required to contribute to the assessment process.

6 Conclusion

The approach presented here provides a new means of accessing information contained in textual databases. By developing techniques for extracting features and identifying relationships between features, we have begun to address types of information needs which queries in typical text retrieval systems could not handle. For instance, we may learn through the *Associations System* that NCR Corp. is related to Robert E. Allen, who is in turn strongly related to the AT & T Co. Discovering the relationship between the NCR and AT & T companies may have been difficult in a text retrieval system because the two companies are not necessarily mentioned in the same articles, and users who must formulate the query will probably not know that they are even looking for articles about the NCR and AT & T companies until they see that this relationship exists.

In the *Associations System* framework, the person-company application addresses a demonstrated user need. This is only one example, however, among many potential applications. The techniques we have described can be similarly applied to other domains such as medical or legal data systems. The implications for such a feature-supported system are far-ranging. A similar system could be developed,

[6] It is unlikely that the retrieval engine identified every document in the WSJ91 collection containing the given base names, but in the absence of a more authoritative source on these occurrences, the results produced by INQUERY are considered a reasonable starting point.

for example, to serve a public health network, providing access to a combination of information, based on ailments, billings, and dates—in a real-time capacity not addressed by current systems. Further, the influence that such a system would have on domains which still rely upon manual indexing or unintegrated software is potentially revolutionary, in terms of time and cost savings. The challenges that we currently face appear to be in designing new evaluation measures which focus on the *usefulness* of the relations generated. Measures like these are essential in order to evaluate and tune the performance of systems like *Associations*.

Acknowledgments

This work was supported in part by the NSF Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts. Bruce Croft initiated the approach used in the *Associations System* and guided its development. Mark Pezarro suggested applying these techniques to the person-company domain. Yufeng Jing contributed significantly to the implementation of the indirect link experiments described in the paper.

References

[1] R. H. Thompson and W.B. Croft. Support for browsing in an intelligent text retrieval system. *International Journal of Man-Machine Studies*, 30:639–668, 1989.

[2] P. D. Bruza and Th.P. van der Weide. Two level hypermedia. In *Proceedings of the International Conference on Database and Expert Systems Applications*, pp. 76–83. Springer-Verlag, 1990.

[3] D. Harman. The DARPA tipster project. *ACM SIGIR Forum*, 26(2):26–28, 1992.

[4] W. Lehnert and B. Sundheim. A performance evaluation of text-analysis technologies. *AI Magazine*, pp. 81–94, 1991.

[5] D. D. Lewis. Text representation for intelligent text retrieval: a classification-oriented view. *Text-based Intelligent Systems*, ed. Paul S. Jacobs, pp. 179–197, LEA Press, 1992.

[6] J. P. Callan, W.B. Croft, and S.M. Harding. The INQUERY retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert Systems Applications*, pp. 78–83. Springer-Verlag, 1992.

[7] M. E. Lesk and E. Schmidt. Lex—a lexical analyzer generator. In *UNIX Programmer's Manual*, Bell Telephone Laboratories, Inc., 1979.

[8] L. F. Rau. Extracting company names from text. In *Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications*, 1991.

[9] C. L. Borgman and S.L. Siegfried. Getty's Synoname ™ and its cousins: a survey of applications of personal name-matching algorithms. *JASIS*, 43(7): 459–476, 1992.

[10] W. B. Croft, H.R. Turtle, and D.D. Lewis. The use of phrases and structured queries in information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 32–45, 1991.

[11] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Meeting of the ACL*, pp. 76–83, 1989.

[12] K. W. Church and W.A. Gale. Concordances for parallel text. In *Seventh Annual Conference of the University of Waterloo Centre for the New OED and Text Research*, pp. 40–62, 1991.

[13] J. K. Ousterhout. *An Introduction to Tcl and Tk*, Addison-Wesley Publishing Company, Inc., 1994.

[14] G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 49–58, 1993.