Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Biomedical Informatics 35 (2002) 236–246

Journal of Biomedical Informatics

www.elsevier.com/locate/yjbin

Information extraction for enhanced access to disease outbreak reports

Ralph Grishman,* Silja Huttunen, and Roman Yangarber

Computer Science Department, Courant Institute of Mathematical Sciences, New York University, New York, NY 10003-6806 ,USA

Received 7 October 2002

Abstract

Document search is generally based on individual terms in the document. However, for collections within limited domains it is possible to provide more powerful access tools. This paper describes a system designed for collections of reports of infectious disease outbreaks. The system, Proteus-BIO, automatically creates a table of outbreaks, with each table entry linked to the document describing that outbreak; this makes it possible to use database operations such as selection and sorting to find relevant documents. Proteus-BIO consists of a Web crawler which gathers relevant documents; an information extraction engine which converts the individual outbreak events to a tabular database; and a database browser which provides access to the events and, through them, to the documents. The information extraction engine uses sets of patterns and word classes to extract the information about each event. Preparing these patterns and word classes has been a time-consuming manual operation in the past, but automated discovery tools now make this task significantly easier. A small study comparing the effectiveness of the tabular index with conventional Web search tools demonstrated that users can find substantially more documents in a given time period with Proteus-BIO.

1. Introduction

Keyword-based document search, which is at the heart of almost all document search tools today,[1] is a remarkably robust and effective tool. But its shortcomings are also well recognized. It is better at locating documents about topics than documents that report specific relationships. Because information is not normalized, keyword retrieval is particularly weak for searches involving numerical relations (“List executives who were indicted more than three times in 2002.”), locative relations (“List companies with headquarters in Pennsylvania which declared bankruptcy.”), etc.

If we are interested in repeated searches of a collection in a limited domain, it is possible to provide much more powerful search tools. If the collection centers around a small number of relations or event types, it is possible to automatically extract and normalize these relations, and present the relations as a tabular database, with links back to the original documents. This allows for much more precise search, homing in on documents expressing particular relations or reporting particular events.

The capability of analyzing natural language text and extracting such relations is provided by an *information extraction* system. Such systems have been developed for a number of news topics under the auspices of the Message Understanding Conferences [1,2]. The need for analyzing large volumes of hospital medical records has led to the development of extraction systems for specific types of medical reports [3–5]. In addition, the rapid growth of the biomedical and genomics literature has prompted the development and application of such systems to scientific articles in these areas [6–8].

We report in this paper on a system, Proteus-BIO, providing a capability for searching for documents on the Web about infectious disease outbreaks. The system gathers Web pages, extracts information about outbreaks, and presents the extracted information in a tabular form with links back to the documents. We first present the overall system, and then focus on the extraction system and the tools which acquire the

* Corresponding author. Fax: 1-212-995-4123.
E-mail address: grishman@cs.nyu.edu (R. Grishman).

[1] Possibly augmented by additional information, such as proximity, frequency with which a document is referenced, etc.

1532-0464/02/$ - see front matter © 2003 Elsevier Science (USA). All rights reserved.
doi:10.1016/S1532-0464(03)00013-3

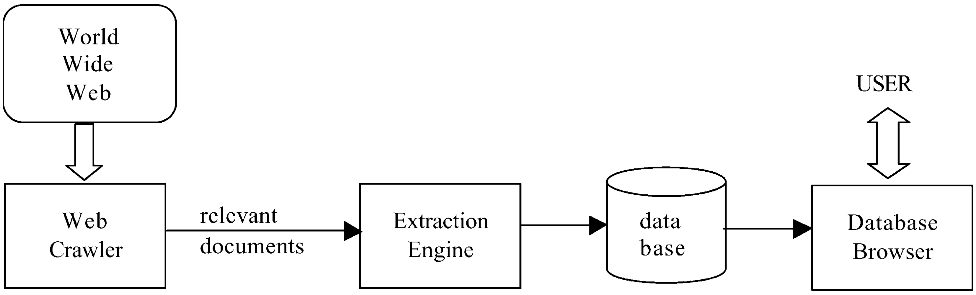


Fig. 1. Proteus-BIO system structure.

knowledge required by the extraction system. Finally, we report on a brief assessment of the effectiveness of the overall system for document search.

2. Overall system structure

The system has three basic components: a Web crawler which finds new relevant Web pages; an extraction engine which builds the database from the documents; and a browser which allows the user to search the database and associated documents (Fig. 1).

The Web crawler traverses portions of the Web each night, looking for new, relevant Web pages. It searches the Web trees starting from the root nodes of selected general and medical news sources, looking for new Web pages. The current prototype system visits two medical news sites, ProMed-mail of the International Society for Infectious Diseases[2] and the Disease Outbreak News of the World Health Organization[3]; one general newspaper, the Chicago Sun-Times, and one general news search engine, Northern Light. In the case of the medical news sources, all new Web pages are sent to the extraction engine. For general news, we apply a simple filter which requires that a document contain at least one word or phrase relevant to an infectious disease outbreak. This filter is intended only to improve efficiency; since most general news stories are not relevant to infectious diseases, it is not worthwhile sending them all to the extraction engine.

The Web crawler has a second function: finding the text body within the Web page. A typical Web page has lots of information besides the actual text of a story: headlines, links to other stories, sponsorship information or advertisements, etc. For most Web pages, the crawler uses the HTML markup to locate the relevant text. For the ProMed Web pages, which contain primarily text without HTML markup, the crawler uses specific text tags and other layout indicators (blank lines, capitalized lines, etc.).

The extraction engine analyzes the text of the story and identifies instances of infectious disease outbreaks. For each outbreak report, it captures specific pieces of information: the location and date of the outbreak, the disease, the number and type of victims, and whether they died. For each such report, the engine adds one row to the database of outbreaks, and links the row back to the document. Roughly speaking, the engine operates by looking for linguistic patterns in the text, such as “outbreak of <disease> killed <victims>,” and uses the variables in the pattern, such as <disease> and <victims>, to fill slots in the database. The database and extraction engine are discussed in detail in the following sections.

The database browser provides the user interface for the system. It presents the extracted information in tabular form, and allows the user to sort and select rows as one would with a typical database or spreadsheet interface. Each row is linked back to the corresponding passage in the document and the relevant items are highlighted in the document.

3. The domain and database

The basic task definition, and the extraction system, were developed as part of an Integrated Feasibility Experiment, IFE-BIO, organized by the Defense Advanced Research Projects Agency[4] to permit the monitoring of a large number of news sources for reports of infectious disease outbreaks around the world.[5]

The task definition led to basic agreement on the types of information (database attributes) to be included: the name of the disease, the time and location of the outbreak, the number of affected victims (infected and dead), and type of victims. However, organizing the information in the text into a set of distinct database entries proved to be problematic. For example, a sentence may report both the number of new cases and a cumulative total (“Since Tuesday the Ebola outbreak has claimed 15 lives in Uganda, bringing the total for this year to 27.”). These

[2] http://www.promedmail.org

[3] http://www.who.int/disease-outbreak-news/

[4] The MITRE was the lead developer for IFE-BIO.

[5] The Web crawler and browser were added to provide a stand-alone functionality separate from the Integrated Feasibility Experiment.

figures contain overlapping information since the new 15 cases are *included* in the total of 27: "this year" contains the time span "since Tuesday." The first part of the event is temporally contained in the second part. Should we have a single database entry with both figures?

Closer examination of additional reports revealed other types of inclusions, such as inclusion by status, as in the following example:

> A total of 2549 cases of meningococcal disease, of which 186 were fatal, was reported to the national health authorities between 1 January and 31 March 2000.

Of the 2549 cases, 186 are deceased. An example of inclusion by case descriptor is:

> On 23 September, 4 birds tested positive for West Nile-like virus, including 3 exotic birds from a local zoo and 1 crow from Westchester county.

Other types of inclusions are geographical inclusion and inclusion by disease type. These types of inclusion relationships complicate the representation of events in this domain.

To handle different types of nesting, we distinguish an *outbreak*, which may include information from several locations and wider spans of time, from the atomic information, which we refer to as an *incident*. An incident involves a single time period, a single location, and a single number of victims, either sick or dead. These incidents may then be connected by various inclusion relations, such as inclusion by time (relating current and cumulative counts), inclusion by location, etc.

Splitting up the description of an outbreak into incidents makes it possible to represent the information in a natural and intuitive way [9]. Each incident becomes a separate row in the database. As a consequence, the database schema is simpler, but there are typically many records per document. The resulting database schema (or "template") is as follows:

Document number	
Disease name	
Date	actual string from text
Normalized date	[start date, end date], where each is of the form year–month–day
Location	town, village, area, etc., as mentioned in text
Country	as mentioned in text, or computed from location
Victim descriptor	short description of victims, as mentioned in text
Victim count	numeric count
Victim status	infected, sick, dead
Victim type	human, animal, or plant
Parent incident	
Inclusion type	(only if parent incident is not empty)

The last two items in the record (parent incident and inclusion type) link the incidents and record the relationships between the incidents, in order to be able to reconstruct the outbreaks.

Dividing the outbreaks into incidents affects the process of extraction, since we can now focus on looking for smaller atomic pieces first, and then connect them through inclusion relations. We are addressing the latter as a separate problem in the overall process of information extraction [10]. The linguistic cues, or cohesive devices, may connect a set of incidents at the sentence level, or across sentence boundaries, depending where the facts appear.

We have identified several linguistic cues that signal the presence or absence of an inclusion relationship between two incidents. These cues can be specific lexical items, e.g., adverbs, verbs, prepositions, or connectives. They can also be nouns or noun phrases that are semantically related, e.g., "bird" is a hypernym of "crow" in the preceding example. We are investigating possible means of locating the cues in the text, to automatically recover relationships between incidents. Only a few relations, marked by local syntactic cues, are recovered in the present system.

4. The extraction system

The Proteus extraction system used for this application has been developed over several years at New York University and was used for several prior evaluations of information extraction, including Message Understanding Conferences 6 and 7 [11,12]. The basic structure of the sentence analyzer is a cascaded set of finite-state transducers; this is now the most commonly used structure for information extraction systems [13].

The heart of the sentence analyzer is a set of *event patterns*. Each pattern corresponds to one way in which the information about an outbreak incident may be expressed. For example, the pattern '*disease* killed *victim*' would match the string "cholera killed 7 inhabitants." If a pattern is matched, a logical structure we call an 'event' is generated, and the event structures are later transformed into rows of the database.

The constituents of an event can themselves be complex phrases; for example, "a new outbreak of cholera killed 7 of the 20 patients in the hospital." Rather than build all of this complexity into the event patterns, we provide a set of preprocessing steps which identify the major linguistic constituents: names and dates; noun and verb groups; and some noun phrases.

Often information about an incident must be drawn from several sentences in the document. Typically, some of this information, such as the disease name or the location, does not appear in the sentence reporting the incident and so must be recovered from context. In

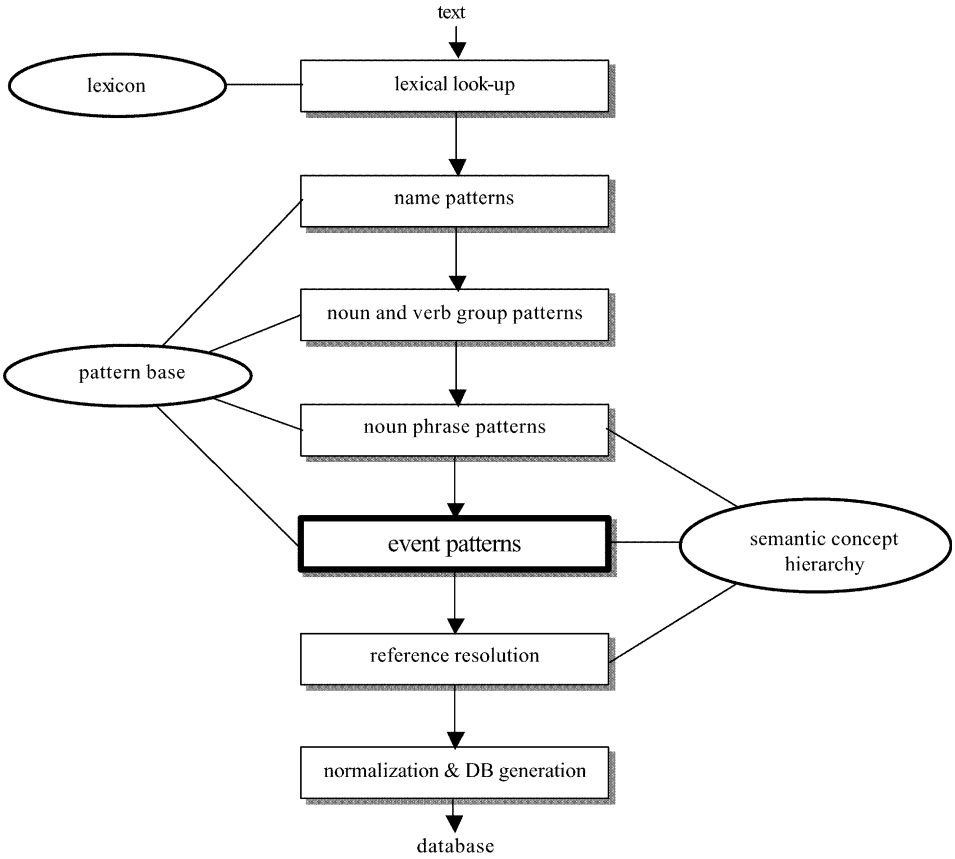


Fig. 2. Structure of the Proteus extraction system.

other cases, an explicit anaphoric phrase (such as a pronoun) is used. Such cases are handled by a reference resolution component and a number of rules which are applied to the output of the event patterns. These rules are task-specific; they are used, e.g., to normalize the values in some of the incident slots. Finally, database (DB) entries are generated for the incidents.

The result is a system structure as shown in Fig. 2. We describe now each of the major components—preprocessing, event patterns, postprocessing, and normalization—in somewhat greater detail.

4.1. Preprocessing

The goal of the initial stages of processing is to identify and classify the phrases which may be the constituents of event patterns.

Processing begins with lexicon (dictionary) look-up. We use both a general English dictionary and several specialized dictionaries. The general dictionary, Comlex Syntax (developed at New York University [14]), provides part-of-speech information and syntactic features for approximately 40,000 English base forms. The specialized dictionaries include lists of disease names, names of major locations (countries, states, major cities) and organizations.

Following dictionary look-up, the system performs several stages of pattern matching, which identify successively larger constituents. The *name patterns* identify names of people, organizations, and locations beyond those explicitly listed in the dictionaries. Thus a name preceded by a title ("Mr. Ripley Rumpole") or beginning with a common first name ("Fred Whatshisname") would be recognized as a person, while a name with a typical corporate ending ("Blascom Associates") would be recognized as an organization. These patterns are applicable to most news stories, although additional patterns may be required for specific tasks. As a result of these patterns, the matched text is tagged with the corresponding *semantic type*, e.g., DISEASE or LOCATION; further pattern matching can then proceed in terms of these, more general semantic labels, rather than the surface word forms.

The *noun group patterns* recognize units consisting of a noun and its left modifiers, such as "the 20 American tourists." The *verb group patterns* recognize a verb and its associated auxiliaries, so that "will be reported" is recognized as a passive form of "report." Both of these pattern sets rely only on syntactic features, and so are applicable across tasks and domains.

Finally, the *noun phrase patterns* selectively associate right modifiers with a noun or noun group. In contrast to the noun and verb group patterns, making the correct attachment for right modifiers requires some semantic knowledge. These patterns are therefore stated in terms of the semantic classes of the domain (for example, PERSON, DISEASE, LOCATION) and are only applied to create noun phrases relevant to the extraction task. Some of these patterns are discussed in the following section, along with the event patterns.

4.2. Event patterns

There are 74 clause- or sentence-level linguistic patterns that capture incidents from the text. For example, the pattern

```
np(DISEASE) vg(KILL) np(VICTIM)
```

will match a clause like

Cholera killed 23 inhabitants.

Here "np(DISEASE)" matches a noun phrase of semantic type DISEASE—generally, a noun phrase whose head is a word of the DISEASE class. These classes are defined by a domain-specific semantic concept hierarchy, which is one of the knowledge bases of the extraction system. Similarly, "vg(KILL)" matches a verb group whose head is in the KILL class and "np(VICTIM)" a noun phrase whose head is in the VICTIM class (which includes people, animals, and plants).

The use of word classes and noun phrase and verb group patterns broadens the coverage of a single pattern. Coverage is further extended by a meta-rule which is automatically applied to each clause pattern; this meta-rule generates the syntactic transforms of the basic active clause, such as the passive, relative, and reduced relative, and adds them as patterns. For the pattern just above, for example, it would generate a passive pattern:

```
np(VICTIM) vg-passive(KILL) by
np(DISEASE)
```

which would match the clause

23 inhabitants were killed by cholera.

The meta-rule also inserts a 'sentence adjunct' (modifier) SA in each position before and after the subject, verb, and object, so the last pattern would in fact be

```
SA* np(VICTIM) SA* vg-passive(KILL) SA*
by np(DISEASE) SA*
```

This is useful to capture modifiers such as temporal and locative information, which can appear almost anywhere in the clause:

last week 23 inhabitants were killed by cholera;
23 inhabitants were killed *last week* by cholera;
23 inhabitants were killed by cholera *last week*.

For a particular extraction task, the sentence adjunct may be used to capture additional modifiers; for example, in Proteus-BIO it is also used to capture disease names when they appear as modifiers in prepositional phrases. For example, the sub-pattern

```
(of | by | from | with | due to | because of)
np(DISEASE)
```

will match sentences like "10 people were hospitalized *with/due to Ebola*." SA is an example of a *sub-pattern*, i.e., it is used only within an event pattern (not by itself), and may be used to fill one or more slots in the event pattern. The "*" of "SA*" indicates that 0 or more sentence adjuncts may occur at each position: "10 people were hospitalized [*with Ebola*] [*in Rwanda*] [*last week*]." There are 20 such sub-patterns special to Proteus-BIO.

Other examples of event patterns are

```
np(DEATH-TOLL) vg(RISE) to <Q>
```

(where <Q> is a number) which will match text like

the number of deaths in China has risen to 67

and

```
np(DISEASE-ACTIVITY) vg(CAUSE) np(DEATH)
```

which matches the text

the Ebola epidemic has caused 4 deaths

In addition, there are 26 *entity patterns* that do not create an incident, but label complex expressions (generally, noun phrases) as single concepts. For example, in the text "the outbreak of Ebola has claimed 4 victims," first an entity pattern will mark "the outbreak of Ebola" as a single noun phrase of type DISEASE-ACTIVITY, and then the above event pattern will match to create an incident.

In summary, the current pattern distribution is as follows:

Top-level patterns	74
Entity patterns	26
Sub-patterns	20
TOTAL	120

4.3. Reference resolution

Frequently the information about a single incident must be assembled from multiple sentences. Sometimes there is an explicit reference to prior information, such as a pronoun ("*He* died two weeks later."). Reference resolution replaces such references with an antecedent noun phrase of the appropriate type (in this case, a person).

The reference resolution procedure first examines a noun phrase to determine whether it is likely to be coreferential with a previously mentioned entity. Phrases beginning with indefinite determiners or quantifiers (e.g., "a," "some," "several") are generally not coreferential, and so are assumed to introduce new entities. For other phrases, including pronouns, the concept hierarchy is used to identify possible antecedents. Each noun phrase is classified with respect to the concept hierarchy based upon both the head of the phrase and the context of the phrase. Given a noun phrase, reference resolution searches for the most recently mentioned prior entity which has the same class in the concept hierarchy or a superclass or subclass, and which has no conflicting modifiers (for example, quantified by different numbers). If such an entity exists, the new noun phrase is taken to be a reference to this entity; otherwise, it introduces a new entity.

In many cases the contextual references are implicit: information about an incident is omitted and must be recovered from context. To handle such cases, we mark certain arguments as semantically *essential*. If they are not explicitly given in the clause containing the incident, reference resolution will treat the argument just as if there was a pronominal reference: it will determine the semantic class of the argument, and will search for an antecedent of that class in the text. This process is used to fill in dates, locations, and disease names. For example, the pattern

```
np(VICTIM)     vg-passive(hospitalize)
[with np(DISEASE)]?
```

matches phrases like "5 victims were hospitalized with Ebola." Since the final prepositional phrase is optional (marked with "?"), the pattern will also match "5 victims were hospitalized." Marking the `DISEASE` argument *essential* will force the system to look for a disease name mentioned earlier in the text. (If no such name is found, the event is discarded.)

4.4. Normalization

To maximize the utility of the database, certain information in the extracted templates needs to be normalized for the operations of selection and search.

Consider the dates extracted by the event patterns. They are natural-language expressions appearing in text, like

a. "as of last week", or

b. "from January 1st to June 30th of 2002."

These text segments become the fills in the incident templates. However, if the user needs to select the records from the database which refer to a particular range of dates of interest, date information is not useful in this form. To remedy this problem, Proteus computes the *normalized date*—the actual date, or date range, from such textual descriptions, either in absolute terms (as in b), or relative to the date of the publication of the report (as in a). The normalized date is a string in "yyyy.mm.dd" format, stored as an additional field in the incident template. This makes sorting and selection possible.

A similar problem exists for locations. Proteus extracts the location mentioned in the text to fill the template. However, a specific location may be a small, little-known village, or even the name of a neighborhood in some city, which would be useless for database search. For this reason, we include an additional slot in the incident template, called "Country," and Proteus normalizes each location, by filling this slot with the corresponding country. Locations are currently normalized in one of three ways: by using local patterns to identify country names corresponding to the location (e.g., "in London, England"); by looking up the country name in a limited on-line gazetteer corresponding to several hundred better-known cities; or by applying heuristics to guess the most likely country from the global context of the document, e.g., using the nearest mentioned country name. (If the database were connected to a rich geographical information system, the normalized location could include the actual coordinate range of the affected region, irrespective of country borders.)

At present Proteus normalizes dates and locations. In the future we also plan to normalize the disease name. A disease can be referred to by a number of names, aliases, variants or acronyms: e.g., "Ebola," "Ebola haemorrhagic fever," "EHF," etc.; anthrax can be referred to as "Ragpicker Disease" or the "Siberian Plague." For more effective record selection, we plan to designate one "canonical" name for each disease, and normalize all variants to that.

5. Knowledge acquisition

As we saw in the previous section, developing an extraction system for a new application requires construction of task-specific patterns, lexical entries, word classes, and rules; these are the 'knowledge bases' of the extraction system. In practice, this has involved the analysis of a corpus to identify the relevant linguistic constructs, and then the manual creation of patterns and classes.[6] This is an expensive process which has limited the application of information extraction.

The corpus analysis involves the review of texts to identify the many ways in which the facts of interest (or bits of the facts) are expressed linguistically. The most common patterns are simple enough to discover through the manual analysis of a small corpus, bringing the system to some baseline level of performance. However, finding the less frequent expressions to improve performance further is difficult and very time-consuming (as predicted by Zipf's Law).

Once the linguistic expressions have been identified, they must be converted into the appropriate form for the extraction system knowledge bases. This is itself a delicate and error-prone task. Discovered patterns need to be combined and generalized to maximize coverage;

[6] These two sides of the customization problem were covered in detail in [15].

changes to the semantic class hierarchy must be analyzed for their effect on the patterns which reference the classes being changed.

A number of tools have been developed to address these problems. These tools range from computer-aided human construction of patterns to automated construction of patterns with minimal human input. We briefly describe several of these, focussing on those developed at New York University and used in the development of Proteus-BIO.

5.1. Machine-aided knowledge acquisition: PET

PET is a set of graphical tools for customizing knowledge bases for a new extraction task [16]. The intended product of PET is a complete, ready-to-run extraction system.

The modules of PET at present are organized into two principal groups:

- Knowledge Base editors: for customizing each of the Proteus knowledge bases:
  - the Lexicon editor;
  - the Concept Hierarchy editor;
  - the Predicates editor;
  - the Pattern editor.
- Document/Template Browsers and Evaluation tools: for browsing textual documents, for applying the IE system to the documents, for scoring the system, and for browsing the resulting templates in a convenient graphical form.

Typically, user interaction centers around two main components of PET: the Document Browser and the Pattern Editor. The user invokes the Document Browser to choose a document from a collection, and to apply the IE system, in its current state, to the entire training collection, to the selected document, or to specific sentences from within the document. The browser is connected to a scorer—a performance-scoring tool—so that if the document is part of a training corpus (i.e., if it has an associated answer key) the user can check how well the system performs on this document.

When the user finds problematic segments or sentences in the document, the user can paste them into the Pattern Editor, and examine in a detailed, level-by-level fashion exactly what the system does when processing the sentence. The sentence can be used as an example from which the Pattern Editor derives new patterns. The new patterns are added to the Pattern Base, and then in turn tested against the given sentence, the document, or the entire training corpus, in an iterative development cycle.

The Pattern Editor attempts to generalize the example text fragment as far as possible, to increase coverage. The Pattern Editor gives the user control over the patterns in the Pattern Base, and over the ordering of their application, which can be crucial to correct operation of the Pattern Base.

The remaining Editors in PET allow the user to modify the respective knowledge bases, following which, again, the performance of the IE system may be checked for improvements.

5.2. Automatic pattern discovery: ExDisco

The next question we address is how the system can aid in the search for linguistic knowledge, in particular for patterns which capture the sought events. Most of the work in this area has involved learning patterns from *annotated* documents: documents which have been marked up to indicate the relevant information, or documents for which extraction templates have been prepared by hand. Riloff [17] described a learning procedure in which documents only had to be marked for relevance to the extraction task. In [18] we introduced an algorithm for learning good-quality patterns automatically from a large, general, unannotated corpus of documents. We now give a brief description of the algorithm, ExDisco.

The main idea is to *bootstrap* the pattern base from a few seed patterns, which are strongly associated with the topic of interest; for example, in our extraction topic, we might seed the system with sentences like "disease killed N people," "victims had symptoms," etc. The algorithm then searches the general corpus for documents matching the initial patterns. (The corpus is general in the sense that it mostly contains articles unrelated to infectious diseases.) The matching documents are considered "relevant" to our interest. The relevant documents are then analyzed syntactically, to produce "candidate" patterns. Each candidate pattern is then matched against the entire corpus, to check how closely it correlates statistically with the relevant document set: the pattern is considered "good" if it matches on the relevant documents substantially more often than on the non-relevant documents. The most strongly correlated pattern is then added to the seed patterns, and the search for documents begins anew.

As a result, the procedure finds many relevant documents, and among them more good patterns, e.g., "victims were diagnosed with disease," "person tested positive for disease," etc., which may not resemble the original seed patterns syntactically, but are closely related to them semantically.

The ideas behind the algorithm are similar to those employed in automatic term expansion in information retrieval. However, rather than searching for topic-specific terms, we search for topic-specific patterns, i.e., syntactic fragments of sentences, or combinations of subject, verb, and object. These patterns may contain terms that individually are not particularly topic-specific, but which in combination are nonetheless quite topic-specific. The bootstrapping of relevant documents and good patterns in tandem rests on the intuitive notion that if a good pattern matches, that indicates that the document is relevant; conversely, if a document is relevant, it is by definition very likely to contain some good patterns.

5.3. Name and term discovery: Nomen

For proper operation of the extraction system, it is essential to have lists of names of certain kinds of physical or logical entities which are as complete as possible. In particular, in the Proteus-BIO domain we need to know as many disease names as possible; other types of essential names are names of disease agents (i.e., bacteria, viruses, fungi, algae, parasites, etc.), names of disease vectors (i.e., rats, mosquitoes, etc.), names of drugs used in treatment, etc. Of more general usefulness are names of locations (cities, countries, etc.), which are needed for many other domains as well.

The medical domain is particularly well endowed with sources of terminology, such as the ICD[7] and UMLS.[8] The emphasis of our work, however, has been on methods for building the required knowledge bases across a wide range of domains, by acquiring the knowledge automatically or semi-automatically from the texts themselves. Even when substantial term lists are available (as for diseases or drugs), these automatic methods have the benefit of being able to discover new variant names and acronyms and add them to the lists.

Over the past few years, several experiments have been reported on using bootstrapping methods for name discovery [19,20]. For the Proteus-BIO project, we developed an algorithm for discovering names, called Nomen [21], similar in spirit to ExDisco. The idea is to give the algorithm a few seed names in *several* categories of interest, and then let the algorithm grow the categories simultaneously, by iteratively collecting contexts which are typical for these names. For example, given a few disease names, the algorithm extracts all local contexts where the seed disease names occur—e.g., a few words to the left, and a few words to the right. Each such local context is considered a pattern. Each pattern is then applied to a large corpus of disease-related documents and evaluated, based on how well the pattern correlates statistically with already known disease names.

For instance, the pattern "X was reported" might occur with many diseases; however, it is also likely to occur with many non-diseases. Hence it is not particularly strongly correlated with the disease category. On the other hand, the pattern "<number> people died of X" is much more likely to indicate that X is a disease.

The strongest pattern is added to a growing set of "disease" patterns; the *new*, previously un-categorized names that it selects are added to the set of diseases. At the same time, the algorithm also learns a few more instances of location names, etc. This procedure progresses iteratively, until no further names can be classified in the corpus. Starting from as few as 10 sample names for each category, Nomen is able to learn thousands of names, with high accuracy; Yangarber et al. [10] report on an evaluation of the quality of the lists that Nomen produces. When applied to a small corpus of 26,000 sentences, among the discovered names there were more than 100 disease names which we did not find in our original manual compilation of over 2200 names of disease and infectious agents obtained from various publicly available sources.[9] These names were added to the Proteus-BIO lexical knowledge base, improving the coverage of the resulting IE database.

6. The database browser

To present the results of information extraction, we use a spreadsheet-like (tabular) interface (Fig. 3). Each incident (database entry) is presented as a separate row in the table. The user can sort the table on any column or combination of columns. The user can also select a subset of rows by placing constraints on the values of a column. Selecting a row and pressing a key brings up the document from which the row was derived. The passage (or passages) in the document which express the event are underlined and are centered in the document window. In addition, the information used to fill certain slots (the disease, the date, the location, the victim description) are highlighted and color-coded. The database browser is implemented for use through a standard Web browser.[10]

7. Assessment

We have done two types of evaluation of our system, one of the *accuracy* of the extraction system and one of the *effectiveness* of the system for document search accuracy.

7.1. Accuracy

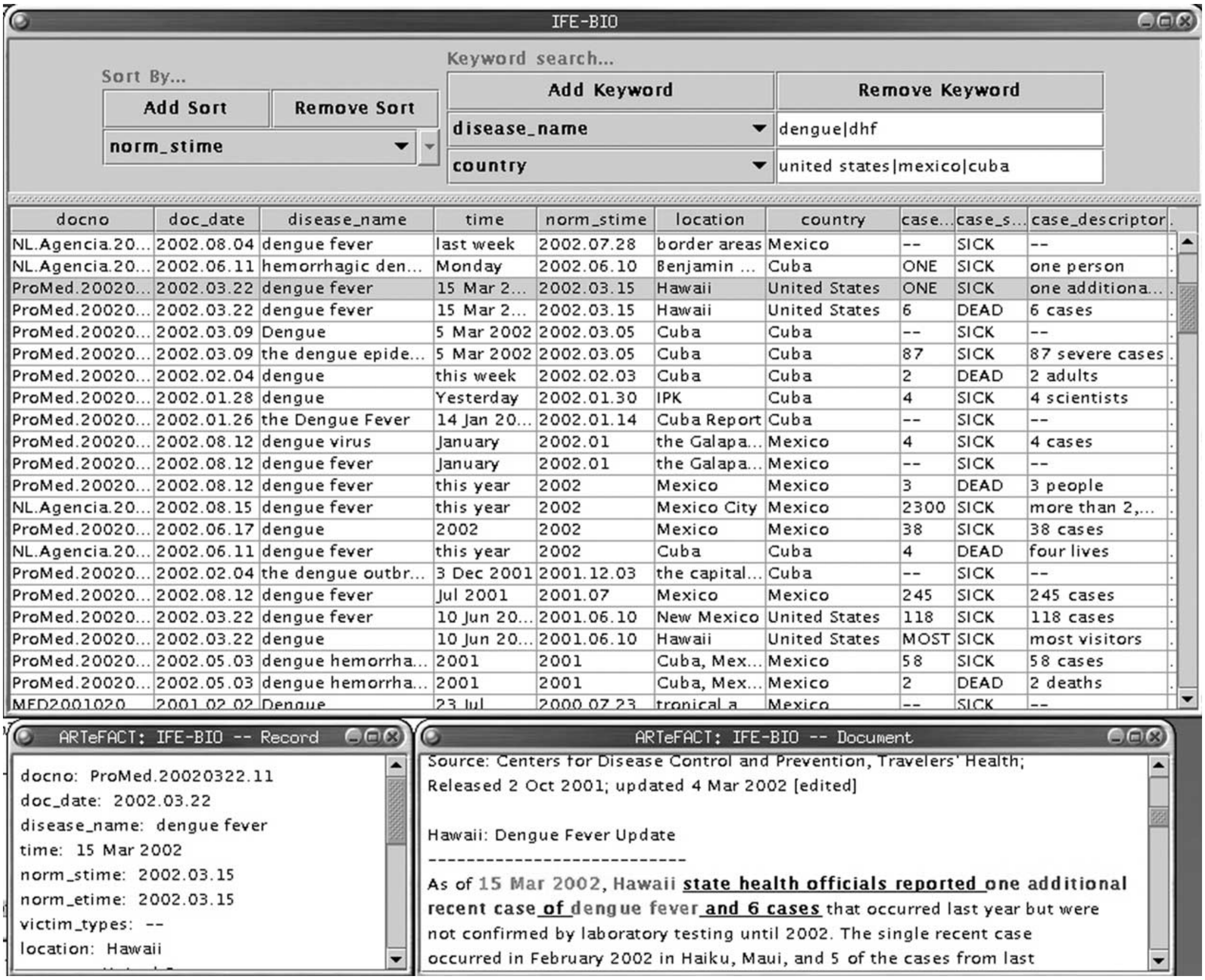
To aid in system development and evaluation, we selected two small collections of documents, a training set and a test set, and manually prepared the

[7] The International Statistical Classification of Diseases, http://www.who.int/whosis/icd10/

[8] Unified Medical Language System, http://www.nlm.nih.gov/research/umls/

[9] To get a sense of the false-positive rate of Nomen, note that after learning, the procedure identifies ~70% of names in the reference list with ~65% precision, i.e., roughly two out of three names are correct.

[10] In fact, two separate database viewers were implemented. One version, based on code written earlier by Troy Straszheim, is a client-side Java applet. Because this did not operate with all browsers, a second viewer, operating on the server side and generating HTML pages dynamically, was implemented by Josh Rosenblatt.



norm_stime

disease_name: dengue|dhf

country: united states|mexico|cuba

docno	doc_date	disease_name	time	norm_stime	location	country	case...	case_s...	case_descriptor	.
NL.Agencia.20...	2002.08.04	dengue fever	last week	2002.07.28	border areas	Mexico	--	SICK	--	.
NL.Agencia.20...	2002.06.11	hemorrhagic den...	Monday	2002.06.10	Benjamin ...	Cuba	ONE	SICK	one person	.
ProMed.20020...	2002.03.22	dengue fever	15 Mar 2...	2002.03.15	Hawaii	United States	ONE	SICK	one additiona...	.
ProMed.20020...	2002.03.22	dengue fever	15 Mar 2...	2002.03.15	Hawaii	United States	6	DEAD	6 cases	.
ProMed.20020...	2002.03.09	Dengue	5 Mar 2002	2002.03.05	Cuba	Cuba	--	SICK	--	.
ProMed.20020...	2002.03.09	the dengue epide...	5 Mar 2002	2002.03.05	Cuba	Cuba	87	SICK	87 severe cases	.
ProMed.20020...	2002.02.04	dengue	this week	2002.02.03	Cuba	Cuba	2	DEAD	2 adults	.
ProMed.20020...	2002.01.28	dengue	Yesterday	2002.01.30	IPK	Cuba	4	SICK	4 scientists	.
ProMed.20020...	2002.01.26	the Dengue Fever	14 Jan 20...	2002.01.14	Cuba Report	Cuba	--	SICK	--	.
ProMed.20020...	2002.08.12	dengue virus	January	2002.01	the Galapa...	Mexico	4	SICK	4 cases	.
ProMed.20020...	2002.08.12	dengue fever	January	2002.01	the Galapa...	Mexico	--	SICK	--	.
ProMed.20020...	2002.08.12	dengue fever	this year	2002	Mexico	Mexico	3	DEAD	3 people	.
NL.Agencia.20...	2002.08.15	dengue fever	this year	2002	Mexico City	Mexico	2300	SICK	more than 2,...	.
ProMed.20020...	2002.06.17	dengue	2002	2002	Mexico	Mexico	38	SICK	38 cases	.
NL.Agencia.20...	2002.06.11	dengue fever	this year	2002	Cuba	Cuba	4	DEAD	four lives	.
ProMed.20020...	2002.02.04	the dengue outbr...	3 Dec 2001	2001.12.03	the capital...	Cuba	--	SICK	--	.
ProMed.20020...	2002.08.12	dengue fever	Jul 2001	2001.07	Mexico	Mexico	245	SICK	245 cases	.
ProMed.20020...	2002.03.22	dengue fever	10 Jun 20...	2001.06.10	New Mexico	United States	118	SICK	118 cases	.
ProMed.20020...	2002.03.22	dengue	10 Jun 20...	2001.06.10	Hawaii	United States	MOST	SICK	most visitors	.
ProMed.20020...	2002.05.03	dengue hemorrha...	2001	2001	Cuba, Mex...	Mexico	58	SICK	58 cases	.
ProMed.20020...	2002.05.03	dengue hemorrha...	2001	2001	Cuba, Mex...	Mexico	2	DEAD	2 deaths	.
MED2001020	2001.02.02	Dengue	23 Jul	2000.07.23	tropical a	Mexico	--	SICK	--	

docno: ProMed.20020322.11
doc_date: 2002.03.22
disease_name: dengue fever
time: 15 Mar 2002
norm_stime: 2002.03.15
norm_etime: 2002.03.15
victim_types: --
location: Hawaii

Source: Centers for Disease Control and Prevention, Travelers' Health;
Released 2 Oct 2001; updated 4 Mar 2002 [edited]

Hawaii: Dengue Fever Update

As of 15 Mar 2002, Hawaii **<u>state health officials reported</u> one additional recent case <u>of</u> dengue fever <u>and 6 cases</u>** that occurred last year but were not confirmed by laboratory testing until 2002. The single recent case occurred in February 2002 in Haiku, Maui, and 5 of the cases from last

Fig. 3. Database browser.

database entries for all the incidents in the documents. The training set consisted of 26 documents from ProMed and 12 from WHO, and was regularly used by the developers to refine and extend the system. The test set consisted of 20 documents from ProMed and 12 from WHO; it was coded by someone other than the system developers and used at the end of each stage of system development to gauge system performance.

The database entries were prepared in the format used by the Message Understanding Conferences (MUCs) [1,2] and evaluated using the MUC scoring program. The basic measures of performance are recall and precision. Recall is the number of correct fields in the system response, as a fraction of the number of fields in the answer key; precision is the number of correct fields in the system response, as a fraction of the total number of fields in the system response (the parent incident and inclusion type fields are not counted in this evaluation). On the test corpus, the precision was 79% and the recall was 41%.

7.2. Effectiveness

Our system was designed primarily to facilitate retrospective search through the archives of disease outbreak reports, in order to see how particular diseases have spread, which diseases have struck particular regions, etc. We therefore felt it was important to assess how effective our system was at improving document search, when compared to a typical tool such as a Web search engine.

We selected two tasks, each of which required the user to find documents relevant to the outbreak of a specified disease in a region of the world over a particular time period.[11] The tasks are briefly described in Appendix A. We recruited six students (from computer science and speech communications, proficient in databases and Web search but not medical experts) to serve

[11] One of these tasks was provided to us by MITRE, which had used it for evaluating their own IFE-BIO system. The other was provided by an epidemiologist, Dr. Marjorie Pollack.

as test subjects. The subjects were divided into two groups, A and B. For the first task, members of group A were provided with the location of the ProMed, WHO, and Northern Light Web pages, and were given time to acquaint themselves with these sites; they were also free to use other sites for their searches. Then they were given the task description and had to use a standard Web search engine (Google) to find relevant articles. Members of group B were provided with the interface to the extracted database, as described above, and were allowed to familiarize themselves with this interface before the evaluation began (the subjects took up to 30 min to do so). Subjects were then given 60 min to find as many articles relevant to the task as they could. For the second task, the roles were reversed—group A used the Proteus-BIO interface, while group B used a Web search engine.

The subjects' performance was evaluated in terms of the number of relevant documents (Web pages) that they were able to find in the allotted time (1 h). The documents retrieved by the subjects were reviewed by one of the authors, and their relevance was determined based on the test criteria. Of the documents found on the Web, on average 84.3% were relevant; the percentage for the database was 84.5%. The subjects using the extraction database were able to find substantially more relevant documents than those using the search engine. Using the database, the median number of relevant articles was 18.5 (range: 10–29); using standard Web search, the median was 12.5 (range: 4–34). This indicates that, even though there were errors and gaps in the extracted database (due to the limitations of the extraction engine), it is a valuable tool for document search, and can be more effective than keyword search for some tasks.[12]

8. Discussion

We have described in this paper a method for automatically extracting some key relations from reports on a single topic, and have demonstrated that this extracted information can be used to speed searching of the report collection. Such an approach should be applicable whenever we have a collection of documents on a particular topic which will be frequently searched, and when one or a few relations provide the key information about this topic.

Essential to the approach is the ability to identify the linguistic patterns which express these relations, and the classes of words and phrases which constitute the arguments of these relations. We have presented tools which can partially automate this discovery process. Such automation should make it feasible to create similar search systems for a wide variety of topics and collections.

The basic ideas of this approach are nearly as old as the computer era. Harris [22] proposed using linguistic procedures to extract selected relations from scientific articles within a single domain, and to use these extracted relations for document access. In subsequent papers and books, he described how a distributional analysis of a scientific sublanguage may be used to systematically discover these relations and the associated word classes.[13] This distributional analysis is one of the foundations for many pattern and word class discovery algorithms, including those we described above. Integrating these algorithms into a unified automatic discovery procedure remains a major challenge but should open the way for much wider use of information extraction.

Acknowledgments

This research was supported by the Defense Advanced Research Projects Agency as part of the Translingual Information Detection, Extraction and Summarization (TIDES) program, under Grant N66001-001-1-8917 from the Space and Naval Warfare Systems Center San Diego, and by the National Science Foundation under Grant IIS-0081962. This paper does not necessarily reflect the position or the policy of the US Government.

We acknowledge the assistance of Winston Lin in the development of this system, including the preparation of training data, implementation of text preprocessing, and the implementation of Nomen.

Appendix A. Extraction tasks

Two extraction tasks were used to evaluate the effectiveness of Proteus-BIO. The following are summaries of the tasks—portions of the instructions given to the test subjects.

A.1. Task 1: Dengue

Dengue fever (DF) is an acute febrile, viral disease frequently characterized by headaches, bone or joint and muscular pain, rash, and leucopenia as symptoms. Dengue hemorrhagic fever (DHF) is a life threatening

[12] After the experiment was complete, we identified several factors that skewed the odds in favor of the Web-based search. Subjects browsed the Web under one operating system, while the database interface was provided under a different operating system—one with which the subjects had less facility. Further, the way in which the task guidelines were defined caused some subjects to study the documents in greater detail while searching the database than while searching the Web, thus spending more time per document and retrieving fewer results. We expect that if we were to repeat the experiments while eliminating these sources of variation, the advantage of the database would be more pronounced.

[13] See the paper by Zellig Harris in this issue.

complication of dengue fever and is characterized by four major clinical manifestations: high fever, hemorrhagic phenomena, often with hepatomegaly, and in severe cases, signs of circulatory failure. Such patients may develop hypovolemic shock, resulting from plasma leakage. This is called dengue shock syndrome (DSS) and can be fatal.

A ProMED posting (31 July 2001) describes the capture in a second Tempe, Arizona neighborhood by Maricopa County health officials of an *Aedes aegypti* mosquito. *A. aegypti* is the primary vector for dengue fever and its variants, as well as yellow fever. The mosquito's appearance marks the first time the species has been found this far north (Tempe is a suburb of Phoenix). Until now, the mosquito was found only as far north as Tucson. It is common in Central and South America.

Use the system to determine the incidence of DF, DHF, and DSS in the Western Hemisphere roughly north of 20 N 00, but also including Mexico City (19N54).

A.2. Task 2: Rift valley fever

The disease of interest for tracking is rift valley fever (RVF): its progression throughout the African Continent and its extension outside of the African Continent. There was an outbreak in Saudi Arabia and Yemen, but nothing has been heard further. Investigate the progression of the disease, in the following order of precedence:

1. Saudi Arabia and Yemen;
2. North Africa and Near East;
3. the African continent;
4. outside the African continent.

References

[1] Grishman R, Sundheim B. Message understanding conference—6: a brief history. In: Proceedings of the 16th International Conference on Computational Linguistics; 1996. p. 466–71.

[2] MUC-7 Message Understanding Conference Proceedings; 1998. Available from: http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html.

[3] Sager N, Friedman C, Lyman MS. Medical language processing: computer management of narrative data. Menlo Park, CA: Addison-Wesley; 1987.

[4] Haug PJ, Christensen L, Gundersen M, Clemons B, Koehler S, Bauer K. A natural language parsing system for encoding admitting diagnoses. In: Proceedings of the AMIA Annual Fall Symposium; 1997. p. 814–18.

[5] Friedman C, Hripcsak G, DuMouchel W, Johnson SB, Clayton PD. Natural language processing in an operational clinical information system. Nat Lang Eng 1995;1:1–28.

[6] Humphreys K, Demetriou G, Gaizauskas R. Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. In: Proceedings of the Pacific Symposium on Biocomputing; 2000. p. 505–16.

[7] Rindflesch TC, Tanabe L, Weinstein JN, Hunter L. EDGAR: Extraction of drugs, genes and relations from the biomedical literature. In: Proceedings of the Pacific Symposium on Biocomputing; 2000. p. 514–25.

[8] Friedman C, Kra P, Krauthammer M, Yu H, Rzhetsky A. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 2001;(Suppl 1):S74–82.

[9] Huttunen S, Yangarber R, Grishman R. Diversity of scenarios in information extraction. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation; 2000. p. 1443–50.

[10] Huttunen S, Yangarber R, Grishman R. Complexity of event structure in IE scenarios. In: Proceedings of the 19th International Conference on Computational Linguistics; 2002.

[11] Grishman R. The NYU system for MUC-6, or where's the syntax. In: Proceedings of the Sixth Message Understanding Conference. San Francisco: Morgan Kaufmann; 1995. p. 167–76.

[12] Yangarber R, Grishman R. NYU: Description of the Proteus/PET system as used for MUC-7 ST. In: Proceedings of the Seventh Message Understanding Conference; 1998.

[13] Appelt D, Hobbs J, Bear J, Israel D, Kameyama M, Tyson M. FASTUS: a finite-state processor for information extraction from real-world text. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence; 1993. p. 1172–78.

[14] Macleod C,, Grishman R,, Meyers A. COMLEX syntax: a large syntactic dictionary for natural language processing. Comput Humanities 1997/1998;31:459–81.

[15] Yangarber R. Scenario customization for information extraction. Doctoral Dissertation, Department of Computer Science, New York University; 2000.

[16] Yangarber R, Grishman R. Customization of information extraction systems. In: Proceedings of the International Workshop on Lexically Driven Information Extraction; 1997.

[17] Riloff E. Automatically generating extraction patterns from untagged text. In: Proceedings of the 13th National Conference on Artificial Intelligence; 1996. p. 1044–49.

[18] Yangarber R, Grishman R, Tapanainen P, Huttunen S. Automatic acquisition of domain knowledge for information extraction. In: Proceedings of the 18th International Conference on Computational Linguistics; 2000. p. 940–46.

[19] Strzalkowski T, Wang J. A self-learning universal concept spotter. In: Proceedings of the 16th International Conference on Computational Linguistics; 1996. p. 931–36.

[20] Collins M, Singer Y. Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora; 1999.

[21] Yangarber R, Lin W, Grishman R. Unsupervised learning of generalized names. In: Proceedings of the 19th International Conference on Computational Linguistics; 2002.

[22] Harris Z. Linguistic transformations for information retrieval. In: Proceedings of the International Conference on Scientific Information, vol. 2. Washington, DC: NAS-NRC; 1959 [reprinted in Harris Z, Papers in structural and transformational linguistics. Dordrecht, Holland: Reidel; 1970. p. 458–71].