INTEX at IBM

Max SILBERZTEIN

Introduction

INTEX is a development environment that allows users to rapidly construct, test and maintain descriptions of specific patterns that occur in texts written in natural language.[1] Each description is represented by a *local grammar*, usually entered via the INTEX graph editor. Local grammars (or *graphs*) can be used to represent:

- character-based patterns, for the recognition of phone numbers (e.g. "*sequence of 3 digits, followed by a space or an hyphen, followed by 4 digits*"), email or Internet addresses, hours or dates expressed numerically, reference or serial numbers, sentence endings, etc.
- orthographical patterns, for the recognition of spelling variants (e.g. "*centre* or *center*"), company names and their variants ("*International Business Machines Corp.*, *Big Blue*"), etc.;
- morphological patterns, for the recognition of families of derived words (e.g. "*France*, *French*, *Frenchmen*, *frenchify*") and inflected forms (conjugation of verbs, inflection of nouns);
- families of lexical entries, for the recognition and indexing of *related* terms and concepts (e.g. "credit card, debit card, MasterCard, visa card...");

[1] See an overview of the system in SILBERZTEIN 1999; the instruction manual is SILBERZTEIN 2000.

✉ Max SILBERZTEIN, IBM T.J. Watson Research Center
e-mail: ms1@us.ibm.com

- morphosyntactic patterns, for the recognition of frozen or semi-frozen expressions, such as complements of dates and times (e.g. "on Monday the 15th at 3PM", "two days ago in the early afternoon"), of locations, addresses, etc.;
- other morphosyntactic patterns for the recognition and co-indexing of *transformed* syntactic constructions (e.g. "N_0's trip to N_1 = N_0 went, traveled to N_1").

One important characteristic of INTEX is that each local grammar can be easily re-used in other local grammars. Developers typically construct simple, elementary graphs that are equivalent to finite-state transducers (FSTs), and re-use these elementary graphs to construct more complex graphs. This process is similar to the method by which engineers build "black boxes" with Computer Aided Design systems to design for instance simple logical operators (AND, XOR) that are subsequently reused in elementary arithmetic operations (ADD), reused in large numbers in more complex arithmetic operations (ADD_{64}), in ALUs, processors, etc. INTEX provides tools to help design, test, debug, refine and maintain large numbers of local grammars in libraries.

Another characteristic of INTEX is that all the objects processed (grammars, dictionaries and texts) are internally represented by FSTs. Therefore, all the functionalities provided by the system are expressed as a limited number of operations on FSTs. For instance, applying a grammar to a text is performed by computing the union of the grammar FSTs, and then the intersection of the resulting FST and the text FST. This architecture allows for very efficient algorithms (e.g. when applying a deterministic FST to indexed texts) and gives INTEX the power of a Turing machine (thanks to the ability to cascade FSTs).

At the T.J. Watson Research Center, several groups are using INTEX for different projects.

Textract (TALENT group, Roy Byrd)

The TALENT ("Text Analysis and Language ENgineering") group has developed a series of text analysis tools that process document collections to automatically feed databases that represent the domain

vocabulary, compute links between terms and concepts that are used to navigate within the documents, retrieve relevant information, compute summaries, etc.

Textract: extraction of vocabularies, relationships, and document structure
Summarizer: extraction of salient sentences
TextLiner: inserting document highlights
Context Thesaurus: find vocabulary items related to a query
Lexical Navigation: explore lexical networks
Prompted Query Refinement: suggest query improvements
TopCat: document categorization

These tools are based on the *Textract* program which processes each individual document of the collection.

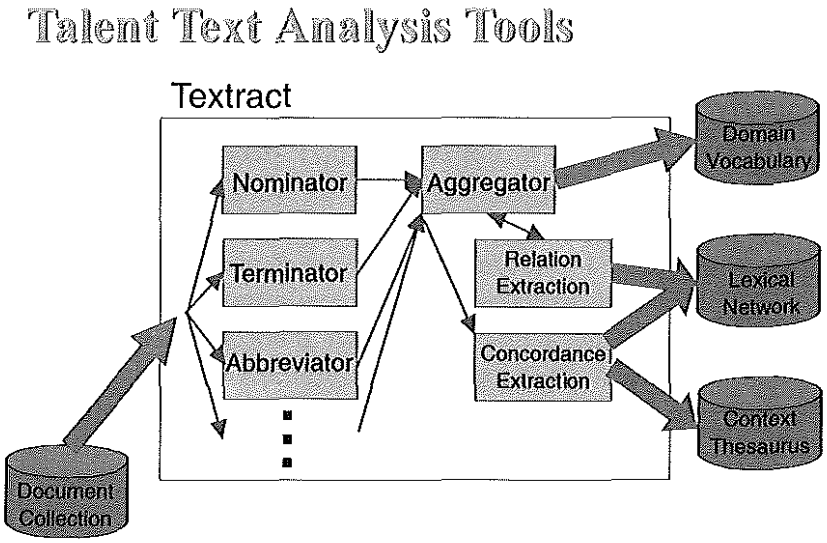


Fig. 1: The Textract *program*

Several of the *Textract* functionalities are based on finite state recognizers that identify relevant items; each type of items corresponds to a specific program module:

SentSep splits the text-file into tokens, sentences and paragraphs;

Text Pattern Recognizer recognizes simple sequences of characters (phone numbers, dates expressed numerically, etc.);
Nominator recognizes single- and multi-word proper names of persons, places and organizations;
Terminator recognizes multi-word technical terms;
Abbreviator recognizes abbreviated terms and acronyms, and links them to their full form.

RAVIN & KAZI 1999 describes *Nominator*; BOGURAEV & NEFF 2000b contains a general description of *Textract*.

These modules, written in C/C++, can only be corrected or updated by their authors. They cannot be easily reconfigured to process different types of corpora (i.e. articles of general newspapers vs highly technical reference documents).[1] In any case, they only process English texts, and in order to adapt them to other languages, one would probably have to completely rewrite them.

Most of these modules' behavior could be simulated by finite state devices that would **recognize** sets of patterns (e.g. *Mr.* or *Miss* followed by a firstname, followed by a single uppercase letter, a period and a word in uppercase) and **produce** the corresponding information (e.g. *Proper name*). INTEX transducers could be used to produce the same results. For instance, the following transducer [Fig. 2] behaves similarly to the *SentSep* module.

INTEX enables users to develop local grammars for the recognition of sentences, proper names, terms and abbreviations for other types of texts or for other languages. Local grammars can be constructed in minutes via the INTEX graph editor; then, a dozen tools to help test, refine and maintain these grammars are available.

Our goal is to add to the *Textract* system the capability of processing any INTEX grammar, in order to produce results similar to the ones produced by the original *Textract* modules.

[1] One might claim that such reconfiguration is not an urgent necessity, as long as we can assume that the syntax of names, terms and abbreviations is stable accross different types of corpora.

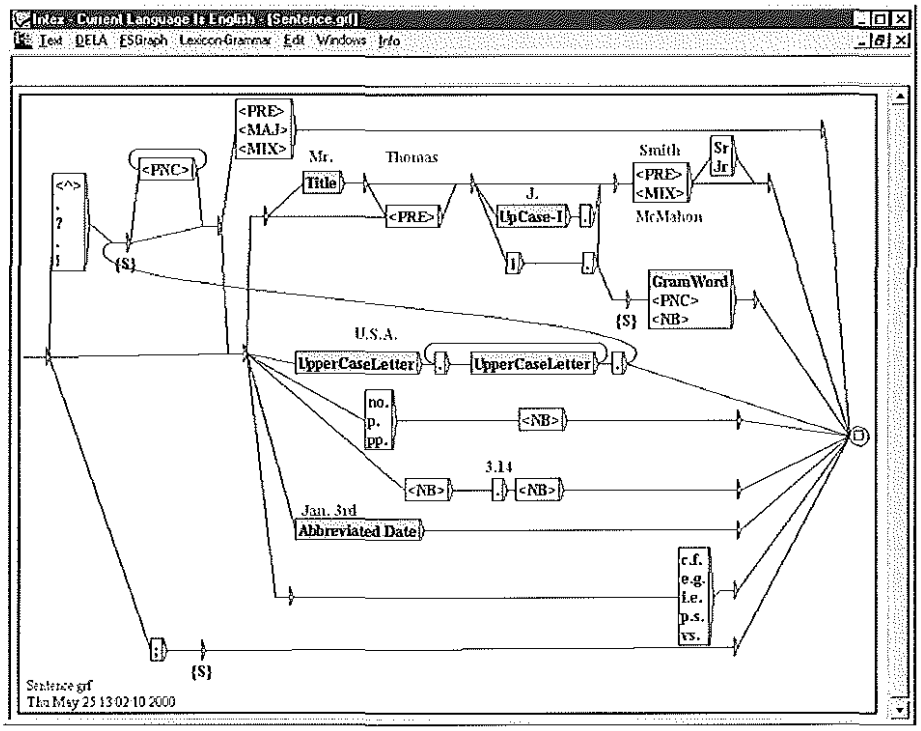


Fig. 2: An Intex graph for the recognition of sentences

Towards finite state analysis of lexical cohesion (TALENT group, Bran Boguraev)

The TALENT group has developed an automatic summarizer that computes a reduced set of sentences extracted from a given document, in order to allow users to get the content of the document very quickly (described in BOGURAEV & NEFF 2000a).

The program is based on the following principle: the summary of a document is constructed by extracting from the document a *coherent* series of *relevant sentences*. *Relevant sentences* contain a minimal number of *relevant information*. Information is relevant for the text if it occurs a significant number of times in a noun phrase and all its possible variants in the text. Considered variants can be morphological variants, synonymous terms and expressions (e.g. *secret* and *clandestine*), a pronoun and the corresponding noun phrase, etc. Sentences of

a series are coherent if they share a minimal amount of information with their predecessors and successors.

The essential component of this summarization model is the calculation of a *salience* measure for each noun phrase in the document. This is crucially enabled by a full configurational syntactic parse of the text. Unfortunately, syntactic parsers do not scale well for real-time processing of large document collections with gigabytes of data. Instead, INTEX is being used to develop a set of phrasal extractors which, when cascaded, emulates a syntactic parser to the extent that one can recover rich configurational information concerning phrase composition and inter-phrasal relationships from the cascade application.

The resulting shallow parser is implemented entirely as a cascade of finite state transducers (FSTs); each FST recognizes specific sequences in the text. The overall organization of the cascade implements the following steps:

1. the first FST recognizes simple NPs, AdjPs and some verbal groups
2. prepositional phrases, post head-noun and verb adjuncts are then recognized
3. Complex NPs: appositives, NP lists
4. Clauses: subordinating, modifying, wh-
5. Sentence Subjects
6. Sentence Objects

The implementation breaks the processing into 12 levels of cascading, realized by means of over 60 INTEX graphs.

Story analysis (Knowledge Socialization group, Andrew Gordon)

In some circumstances, it is valuable to be able to find information concerning particular procedures or activities within a large text collection. In particular, many knowledge management goals can be achieved by locating stories in narrative collections that describe the experiences of people engaged in particular tasks. Andrew Gordon is interested in using the INTEX system to design patterns that can be used both to identify the breadth of activities discussed in a narrative

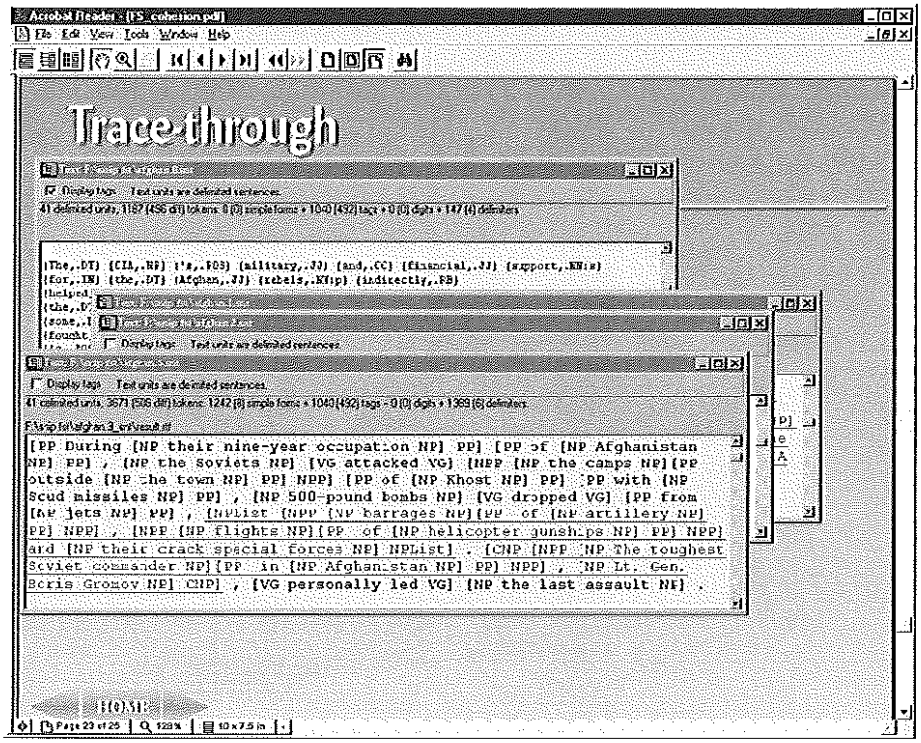


Fig. 3:
A cascade of 3 transducers is used to recognize a complex Noun Phrase

collection, and to recognize references to activities that are commonly understood within a particular community (see GORDON 1999).

Some examples of the kind of interesting statements sought include the following from a collections of stories written in the 1930's: "*I'm* **doing a** *three-act with a couple of kinkers*" and "*He was* **going for** *the daughter like a hungry pooch after meat*".

Local grammars are used to automatically recognize characteristic terms and expressions of activities. For instance, the following graph recognizes the sequences "*doing a*" and "*going for*", which occur in many verbal expressions of activities.

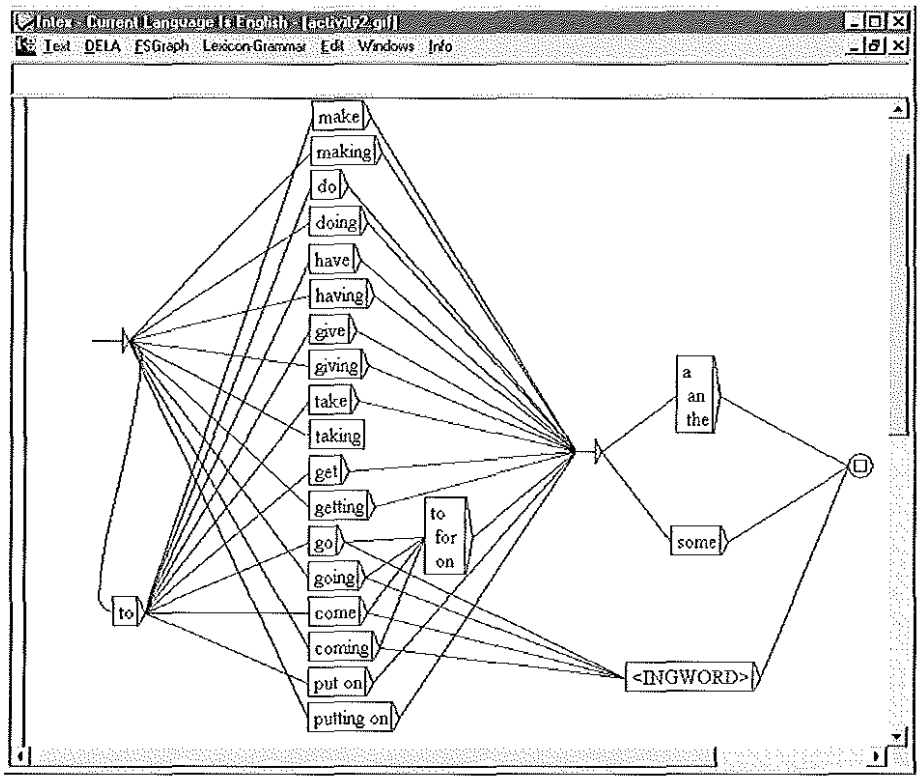


Fig. 4: An INTEX graph used to recognize verbal constructions of activities

Web-wide oncology information for the lay researcher (Applied Learning Sciences Group, Linda Tetzlaff)

People with medical conditions increasingly access the Web for comprehensible and timely information on their disorders. Linda Tetzlaff's project intends to improve access for patients with cancer to Web information generally and then focus on a subset of Web pages that reflect personal experience. The goal is to generate comprehensive and authoritative information utilizing a web crawl, post-crawl analyses and a computer-assisted review process.

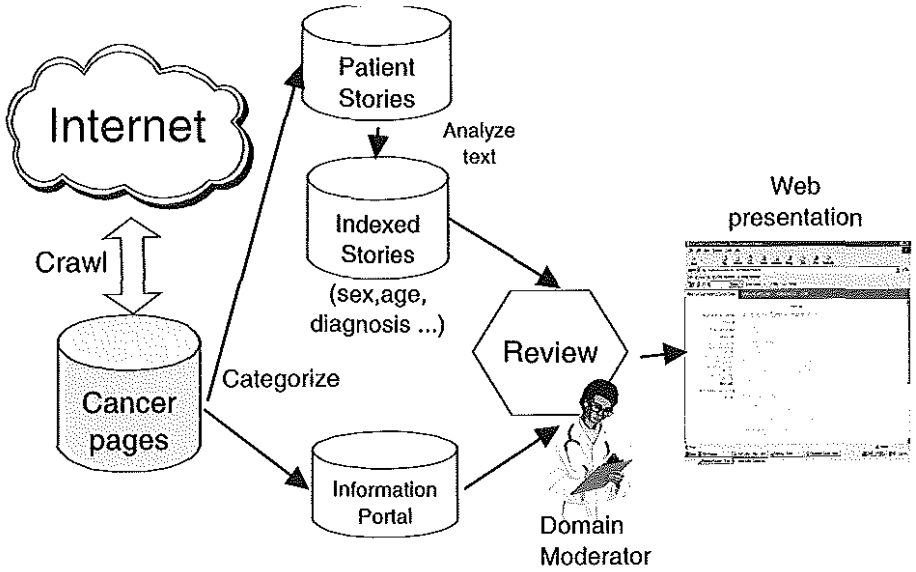
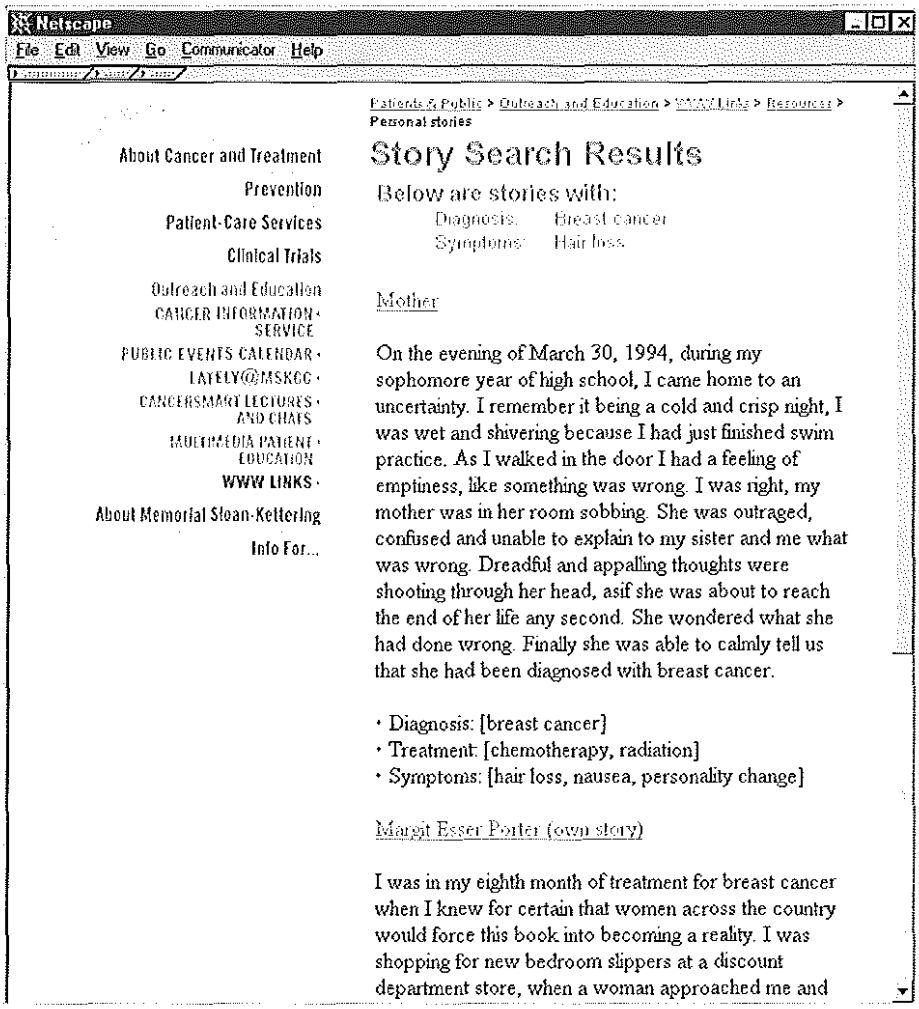


Fig. 5: Architecture of the Oncology system

INTEX, as one of the components of analysis, is used to extract patient-salient information from the stories, such as author, sex, diagnosis, treatment, and complications. The results will feed a Web-based interface from which the patient can find stories of personal relevance.

The first step in the process is to crawl the Web for all pages relating to cancer. Pages are categorized as stories or not-stories. Not-stories are further categorized, using another process, into a topical, hierarchical tree. Stories are processed using INTEX graphs to extract patient-salient information, such as probable diagnosis, and treatment. While much of this information appears in canonical form (e.g. chemotherapy, hair loss), much is also loosely or indirectly imbedded in the narrative (e.g. 'Chemo #7, my hair quits with nary a 'by your leave'.'). INTEX output is used to make further inferences about the story. For example, stories heavily weighted with first person pronouns are assumed to be first person narratives. In conjunction with other evidence from the text, this supports identification of the story as a personal story, written by the patient. The analyzed features

are stored in a database which can be accessed from the Web. End users can specify their perferences in a simple checkbox interface, and salient stories are returned. Stories are represented by their intial sample text and, a summary of features is automatically constructed from the INTEX analysis (see below).



LittleHelper (Wearable Device Platform Group, Michael Olsen)

E-mail was originally designed as a communications application. Now it is being used for a wide variety of additional purposes:

- work task management
- document delivery and archiving
- sharing names and addresses
- sending reminders and scheduling appointments.

These additional usages are denoted as *E-mail overload*, which:

- causes people to spend and waste more time processing their E-mail
- causes frustration in categorizing all this information.
- causes some level of anxiety because E-mail is becoming a media for distribution of time-critical, personal, and work related information.

In an effort to minimize transcription of information found in E-mail text to calendar and address book, Michael Olsen is developing an application, ***LittleHelper***, which identifies calendar and address book items in E-mail text. INTEX is an integral part of this analysis:

- INTEX aids in conditioning the text by expanding contracted terms, e.g. "Joe's" is either "Joe is", "Joe has" or "Joe's", thus enabling *LittleHelper* in properly identifying pronouns and verbs;
- INTEX aids in spotting loosely put dates, times and combined date-times, e.g. "*saturday morning*" and "*a week from next Monday*" which are in turn translated into precise dates and times in *LittleHelper*.

For more information please contact Michael Olsen (cmolsen@us.ibm.com).

Predictive Annotation in Question Answering (Knowledge Structure Group, John Prager)

Question Answering is an exercise in which a user enters a question in a computer (e.g. "*How tall is the Matterhorn?*"), then the computer looks up a large collection of documents (or WEB sites) to try to

locate the answer, as expressed somewhere in the collection (e.g. "*The institute revised the Matterhorn's height to 14,776 feet 9 inches, citing a recent survey...*").

The architecture of the Question Answering system developed by John Prager is the following:

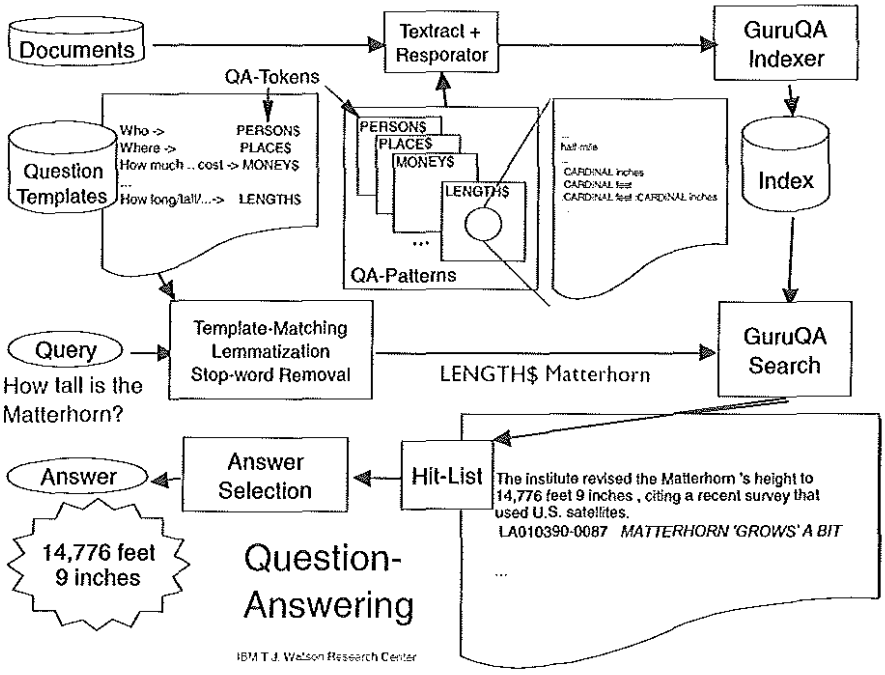


Fig. 6: Architecture of the Question Answering System

- A series of templates is first constructed; each template represents, on the one hand, a typical question and its variants (e.g. 'how long is', 'what is the distance from ... to', 'how many miles between', etc.); on the other hand, a set of patterns that can be used to recognize possible answers (e.g. '3 miles'). All the templates (e.g. LENGTH$, PERSON$, PLACE$, etc.) are indexed in the collection of documents;
- the user query is parsed; the analysis involves lemmatizing the text, removing stop words and matching the question against the series of templates. The result is a bag of words that contains relevant words

and templates, e.g. (LENGTH$, Matterhorn). A search engine is then used to apply the bag of words to the index of the documents;
- a specific ranking system selects in the hit list the passages of the documents in which the answer is most likely to appear (for instance, if the indexed words and templates occur in the same sentence).

See PRAGER, BROWN & CODEN 2000 for more information on the system. John Prager's approach to Question-Answering requires identifying in a text potential answers to questions: people (answers to *Who*), places (answers to *Where*), times and dates (answers to *When*) and so on. Local grammars can be developed to parse questions in Natural Languages, and to perform term expansions.

Usually, local lexical analysis is not sufficient to distinguish different kinds of proper names from each other. John Prager plans to use INTEX to identify patterns such as "Shakespeare wrote ..." or "... written by Shakespeare" to establish Shakespeare as person, which in turn will allow us to identify "Shakespeare's Hamlet" as a work of literature.

Perspectives

The functionalities of INTEX that are most appreciated by the different groups are the capability to rapidly construct elementary graphs that process specific patterns, and the possibility of reusing these graphs in other, more sophisticated grammars.

With Cédrick Fairon's help, we are going to build a *Textract*-like module that will use INTEX graphs to recognize sentences, proper names, technical terms and abbreviations in French texts.

Other groups within the T.J. Watson research center are considering using INTEX in various projects: INTEX could significantly enhance statistical processing; INTEX could be used as a front end to develop taggers and CF parsers, etc.

Already, several general grammars, e.g. to recognize phone numbers, dates, beginnings of noun phrases, verb groups, etc. are being developed independently by different groups. The capability of INTEX to centralize and accumulate the linguistic data in large libraries could

allow researchers to benefit from each other's effort, and build reliable and fast natural language-enabled applications.

Aknowledgement

This article describes some of the uses to which INTEX is being put in IBM T.J. Watson Research Center. I would like to acknowledge the assistance of the people whose work is described here: Roy Byrd, Bran Boguraev, Andrew Gordon, Linda Tetzlaff, Michael Olsen, John Prager.

References

BOGURAEV (Branimir), NEFF (Mary): 2000a, "The effects of analysing cohesion on document summarization", in *Proceedings of the 18th international conference on Computational Linguistics. COLING'2000* (Saarbrücken).

BOGURAEV (Branimir), NEFF (Mary): 2000b, "Discourse segmentation in aid of document summarization", in *Proceedings of Hawaii international conference on system sciences. HICSS-33* (Maui, Hawaii).

GORDON (Andrew S.): 1999, *The Design of Knowledge-rich Browsing Interfaces for Retrieval in Digital Libraries*. PhD Dissertation (Northwestern University, Department of Computer Science).

PRAGER (John), BROWN (Eric), CODEN (Anni): 2000, "Question-Answering by Predictive Annotation", in *Proceedings of SIGIR' 2000* (Athens).

RAVIN (Yael) and KAZI (Zunaid): 1999, "Is Hillary Rodham Clinton the President? Disambiguating Names accross Documents", in *Proceedings of the ACL '99 Workshop on Coreference and its Applications*, June 1999. Textract.

SILBERZTEIN (Max): 1999, "Text Indexation with INTEX", in *Computer and the Humanities*, 33 (Amsterdam: Kluwer Academic Publishers).

SILBERZTEIN (Max): 2000, *Manuel d'utilisation INTEX 4.3* (Paris: LADL, Université Paris 7).

Downloadable from www.ladl.jussieu.fr/INTEX.