FactBrowser Demonstration

Scott Miller, Sergey Bratus, Lance Ramshaw, Ralph Weischedel, and Alex Zamanian
BBN Technologies
70 Fawcett St
Cambridge, MA 02138
1-617-873-2078
{szmiller, sbratus, lramshaw, weischedel, azamanian}@bbn.com

ABSTRACT

The FactBrowser demonstration illustrates automatic database update from live feeds based on information extraction from text and the ability to browse the resulting database for unexpected connections. The technology used has four interesting features:

1. The demonstration employs a **light architecture** based on the Web; using an XML-based client-server architecture, the graphical user interface requires only Internet Explorer 5.0 or higher. No application code resides on the client.
2. A **permanent database** grows based on cross-document entity tracking and accumulating facts.
3. The database is **updated daily** based on automatic processing of documents distributed by the Foreign Broadcasting Information Service (FBIS). Document capture and database update are fully automatic, requiring no human intervention.
4. The following key components: name finding, parsing, and pronoun resolution are all based on the trained, language-independent **statistical modeling techniques**.

The strategic focus throughout the design of FactBrowser has been on producing high precision output so as to maintain quality in the data base.

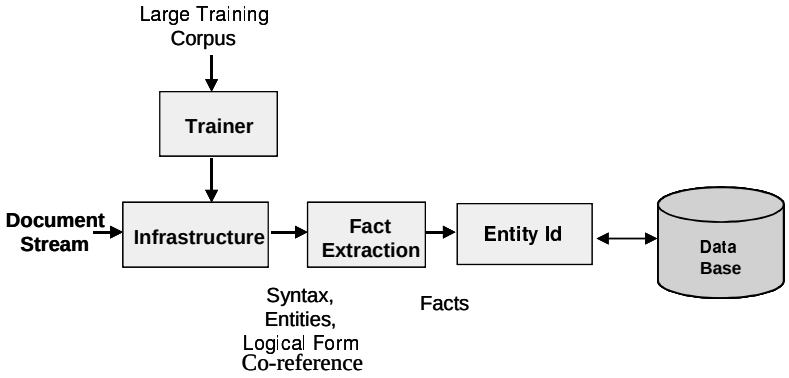

Figure 1: Information Extraction System Structure

1. INTRODUCTION

FactBrowser analyzes a daily stream of world news documents, extracting information about entities and relations between them. Extracted information is stored in a database and viewable through tables that list all entities and all the relations found in the collection. The types of entities currently extracted are *Person, Organization,* and *Location*. For each entity, the table lists both the type and the different names and descriptions of that entity, allowing the user also to display every mention of an entity in their source document context. The system currently tracks three relations:

OfficeHolder, i.e., who plays what role in what organization

OrganizationAt, i.e., where an organization is located

Attribution, i.e., who is quoted as saying what.

In addition to automatic updating of the database daily, a browser interface enables exploring the database. By following threads across the fact and entity tables, the user can determine, for example, other people who work at the same organization as a given speaker. FactBrowser produces versions of the documents annotated with the results of the analysis, including a full syntactic parse of the text with additional semantic information like name types and descriptor co-reference links attached to the appropriate parse tree nodes.

The extraction system consists of three subsystems, as illustrated in Figure 1.

2. DOCUMENT LEVEL ENTITY EXTRACTION

The bulk of the analysis for each document can be done independently of other documents in the collection and so can be easily parallelized. The path of a document through this stage of processing is as follows:

Loading: A document is loaded, sentence-broken, and tokenized (in the style of the UPenn Treebank). If the input document

contains section, paragraph, or topic markup, this markup is preserved.

Name finding: BBN's IdentiFinder™ (Bikel et al., 1999), a statistical name finder, is used to mark name mentions of the three semantic types. These mentions, marked as spans of tokens, are called *base name mentions*.

Parsing: Constrained syntactic parsing is performed, using a lexicalized probabilistic context free grammar (Miller, et al., 2000). BBN's statistical parser is trained on the UPenn Treebank and run in a constrained mode that ensures that the constituent brackets that it identifies will not conflict with the base name mentions previously found by IdentiFinder.

Parse merging: The name information from IdentiFinder is merged into the syntactic parses, which sometimes involves inserting additional nodes into the parse tree. Since the parser follows Treebank style, it produces relatively flat parses for NPs. For example, a name with premodifiers like "small town America" would parse as a single, three-element node. In such cases, the system introduces a new node covering just the name portion. Futhermore, if the name is that of a *Person* and the premodifiers include what appears to be a title, as in "Microsoft President Bill Gates", the system also introduces an NP node over the title, making it available for descriptor finding in the next stage.

Descriptor finding: A statistical model classifies the NP-type nodes in the parse trees, in order to identify those that are probable descriptors of persons or organizations. The model is based on the head word of the NP, the head word of its parent constituent, and any left modifier. The model is trained on newswire data in which the descriptor types were marked by hand. The resulting descriptors are referred to as *base descriptor mentions* and are appropriately labeled in the parse.

Structured mention analysis: Some local co-references between the base name and a description can be recognized with high reliability. For example, NPs that contain appositives or that contain a base name mention with a post-modifying clause are marked at this stage as *structured mentions*, and the name co-referenced with the description.

Document-level name co-reference: The system finds non-local co-reference relations between the name mentions in the document. Rules specific to the name's type are used to generate a list of possible alternate or abbreviated forms for each name. For example, the person name "John Smith" would generate the alternate forms "Smith" and "Mr. Smith", while the company name "Smith Enterprises" would generate "Smith Enterprises, Inc." and "SE". Any of those alternate forms that occur elsewhere in the document are then linked together with the source name.

Pronoun resolution: A generative statistical model resolves pronouns. The estimated probability of a link between the pronoun and its antecedent relies on features like number, gender (determined by heuristics), and a distance measure based on the Hobbs [1977] tree search that outlines the order in which NP-type nodes are considered as possible antecedents for a pronoun.

Entity creation: Sets of co-referring mentions (names, locally-linked descriptors, and pronouns) are *entities*, and the corresponding metadata for each entity is added to the document.

3. FACT/RELATIONSHIP EXTRACTION

The FactBrowser demonstration system extracts three types of relations: 1) person has role in organization, 2) organization is located in place, and 3) statement is attributed to person. To maintain database integrity, the system is intentionally biased toward high precision, at the cost of some recall.

Relations are identified by recognizing syntactic patterns in parse trees produced by the statistical parsing component. The entities mentioned in the relations are resolved to underlying database entries by the name co-reference and pronoun resolution components. Thus, pronominal mentions and shortened versions of names are resolved to the most descriptive known strings for those entities.

Fact/relationship extraction is the final stage in document level processing. Once complete, the system has marked the entities that the document is about, the places in the document where each entity is mentioned, and the relations in which they are said to participate.

4. COLLECTION LEVEL PROCESSING (CROSSDOC)

The third component connects the entities found in document-level processing with entities previously encountered in other documents. This stage is less amenable to parallelization, and requires growing resources as the collection size grows.

In the current system, only the simplest heuristic is implemented. The connection between a document-level entity and a *global entity* is established on the basis of *canonical name mentions*. For each document-level entity, a *canonical name mention* is constructed from the tokens of the mentions in its mention set. This process includes removal of all punctuation, case normalization, and removal of any parenthesized groups of tokens. After such processing, the longest base name mention is chosen as the canonical one to represent the document-level entity. A database query using this canonical name as a key then returns all records from the global database that may match the entity. Finally, a decision is made as to whether the document-level entity matches one of these records, in which case its mentions join that record, and that database record is updated. If none of the existing database records match, a new record is created and introduced, based solely on the mentions of the document-level entity.

After the cross-document stage, each mention in the processed document text is marked with its global co-reference information, which can then be used for generating cross-linked views.

5. VISUAL DISPLAY

The FactBrowser interface displays tables of entities (people or organizations) and of facts. The entry for each entity points to all of the locations in each of the documents where that entity was mentioned, and the system can display the source text surrounding any of the mentions.

FactBrowser thus enables database level, rather than sentence level, analysis. In Figure 2, for example, although the sentence identifies Bangaru Laxman as the chief of "BJP", the spreadsheet view correctly shows Laxman as head of the Bharatiya Janata Party. (The longest name, or the longest description, if the entity

has no name in the data base, is used for display in the spreadsheet browser.)

6. CONCLUSIONS

This effort is still in its early stages. Much has been learned from transitioning from processing a file of data as an experiment to processing documents as a continuous stream, from assimilating information across documents, from updating an existing data base of entities, and from the challenge of maintaining a 24 by 7 portal into the data. Yet there is much to be done. The fundamental challenges still remain: significantly reducing the error in extracted data (reducing both missed data and incorrectly extracted data), improving cross-document correlation of both entities and facts, and massively reducing the amount of training data required to achieve high performance.

7. ACKNOWLEDGEMENTS

The work reported here was supported in part by the Defense Advanced Research Projects Agency under contract numbers N66001-99-D-8615 and N66001-00-C8008. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

8. REFERENCES

[1] Bikel, D., Schwartz, R., and Weischedel, R. "An Algorithm that Learns What's in a Name," Machine Learning 34 (1999), 211-231.

[2] Miller, S., Ramshaw, L., Fox, H., and Weischedel, R. "A Novel Use of Statistical Parsing to Extract Information from Text", In Proceedings of 1st Meeting of the North American Chapter of the ACL, (Seattle, WA, 2000), 226-233.

[3] Hobbs, J. R., "Resolving Pronoun References", reprinted in 1986 in Readings in Natural Language Processing, B. Grosz, K. Jones, and B. Webber, eds., Morgan Kaufmann, (1977).

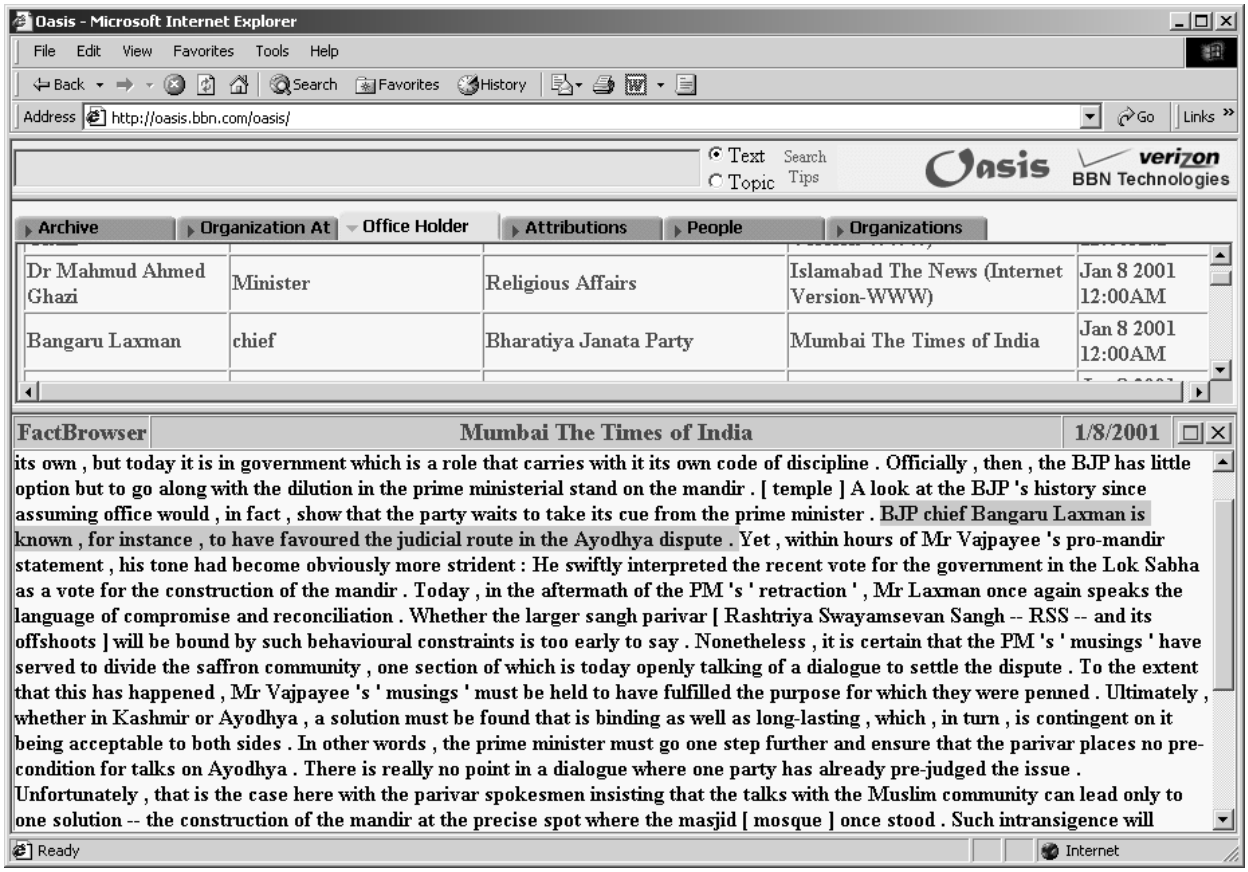

Figure 2: Screen shot illustrating demonstration