[*MT evaluation: basis for future directions*. Proceedings of a workshop...2-3 November 1992, San Diego]

MESSAGE UNDERSTANDING: The MUC Methodology

Beth Sundheim
U.S. Naval Command, Control, and Ocean Surveillance Center
RDT&E Division (NRaD)

Unlike machine translation, "message understanding" is not a task that has a well-defined output. The DARPA-sponsored research into English message understanding in the 1980's has stimulated the development of an evaluation methodology that views text analysis in the context of an "information extraction" task. Originally, the texts used for research and evaluation purposes were paragraph-length naval messages; now they are longer, open-source newswire articles on more broadly defined topics. The task has evolved to one in which "who"/"what"/"where"/"when" types of information are extracted with increasing attention paid to the interrelationships among the pieces of information.

The methodology is being applied to English, Japanese, and Spanish in a variety of subject areas. It requires an intensive data preparation effort and makes use of specially designed evaluation software. The metrics focus on the completeness and accuracy of the system's output and are meant to reveal tradeoffs the system is making between missing some percentage of the expected information and generating erroneous (including spurious) information. Thus, message understanding is being evaluated in the context of an application that has encouraged the development of systems that can locate interesting bits of information and tie them together, while ignoring large segments of uninteresting text.

DATA EXTRACTION: Semi-Automated Evaluation of Japanese and English Output

Lynn Carlson
U.S. Department of Defense

In the DARPA-sponsored Tipster Program, automated data extraction systems are being developed for Japanese and English in the domains of joint ventures and microelectronics. These domains and languages were chosen to fulfill the Tipster Program goal of demonstrating both domain- and language-independent algorithms. The data extraction tasks involve automatically filling object-oriented data structures, or templates, with information extracted from news articles in one of the four domain-language pairs. For each domain, a set of objects is defined, each of which contains various slots which define generic types of information to be extracted and represented in an appropriately formatted data representation. After the Tipster systems have detected whether a particular text contains relevant information, they must extract specific instances of the generic types from the text and output the information in the appropriate template slots. These slots are then scored using a semiautomated scoring program with manually produced templates as keys.

The human analysts who produce the keys are also preparing development set templates for each domain-language pair, which serve as training models for system developers. Each domain-language pair has an associated set of 1,200-1,600 documents, each with a corresponding template to be filled. In preparing these templates, the analysts follow a set of detailed fill rules developed for the domain

and language. The documents and their templates are divided into the development set and three test sets, to be used for the 12-, 18-, and 24-month evaluations. For the 12-month evaluation (September 1992), only the joint ventures domain was evaluated, with 100 templates for English and 50 for Japanese. Future evaluations will test both domains and languages using larger test sets.

The reporting task for joint ventures involves capturing information about business activities of entities who enter into a cooperative agreement for a specific project or purpose. The template consists of 11 different object types, which together capture essential information about joint venture formation and activities, including who the business partners are, what type of activity and industry is involved, and so on. For any particular object, a slot may be defined as having one of the following types of fillers:

(1) A pointer to another object: In the TIE-UP-RELATIONSHIP object, the filler for the ENTITY slot is a pointer to an ENTITY object.

(2) A set fill: In the ENTITY object, the TYPE slot is filled by selecting one element from a predefined set. For English, the set members are COMPANY, GOVERNMENT, PERSON, OTHER.

(3) A normalized form: In the TEMPLATE object, the DOC DATE slot is filled by YYMMDD, as in 851105.

(4) A string fill: In the ENTITY object, the ALIAS slot is a string fill, as in "IBM".

For the 12-month evaluation, only the top four objects of the joint ventures template were evaluated; these included the TEMPLATE, TIE-UP-RELATIONSHIP, ENTITY, and ENTITY-RELATIONSHIP objects. Criteria were established for determining correct, partially correct, and incorrect answers consistently across the two languages for the evaluation. Fully automatic scoring was done for slot fills which were either pointers or set fills. For normalized forms and string fills, semiautomatic scoring by human analysis was conducted for this evaluation. The guidelines for this interactive scoring were as follows:

(1) The scoring was conducted "blindly," in that the human analysts did not know whose system was being evaluated.

(2) Two analysis scored each system together.

(3) The scoring was context-free: the original documents were not consulted during the scoring process.

(4) Partial or full credit was given whenever it was deemed reasonable. Stricter rules will be applied for the next evaluation.

(5) The same criteria were applied to both languages.

(6) The overall guiding principle was to determine whether the response uniquely identified the same entity as indicated in the key.

Some examples of how these rules applied to actual slots from the Tipster 12-month evaluation are given below. The slot name is given, followed by the nature of the mismatch and, in the last column, the score applied (I = incorrect; P = partial match; M = full match).

Slot	Nature of mismatch	Score
NAME	Extra names included in response Expected: "Kajima CORP" Actual: "Citibank and Kajima CORP"	I
	Extra words in response Expected: "Citibank" Actual: "Yesterday, Citibank"	P
	Alias given instead of name Expected: "General Electric" Actual: "GE"	P
	All proper nouns left out of response Expected: "Asahi Glass Co" Actual: "Glass Co"	P
NATIONALITY	COUNTRY left out of response Expected: "JAPAN (COUNTRY)" ACTUAL: "JAPAN"	M
LOCATION	City missed; country correct Expected: "TOKYO (CITY) JAPAN (COUNTRY)" ACTUAL: "JAPAN"	P
	City correct; country missed Expected: "LOS ANGELES (CITY) UNITED STATES (COUNTRY)" ACTUAL: "LOS ANGELES (CITY) JAPAN (COUNTRY)"	I

DATA EXTRACTION:
The MURASAKI Project: Multilingual Natural Language Understanding

Rita McCardell Doerr
U.S. Department of Defense

MURASAKI is a multilingual data extraction application under development for the Department of Defense. It processes Spanish and Japanese newspaper articles reporting statistics about the AIDS disease. Key to MURASAKI's design is its language-independent and domain-dependent architecture. Significant progress has been demonstrated in its two-year development to date.

In the initial stages, we addressed issues of Japanese language processing and display, built Spanish and Japanese lexicons and basic grammars, and developed domain data. During this past year we focused on advanced development issues. These include a multilingual approach to grammar implementation and a new, language-independent discourse module.

Syntactic analysis consists of a parser and a grammar. Our parser is entirely language-independent. There is a separate grammar for each language, but the linguistic core of each grammar is the same. This semi-language-independent strategy provides maximum robustness while taking advantage of the insights provided by syntactic advances within the field of linguistics.

The MURASAKI discourse module, unlike other implemented discourse modules to date, is language-independent: it is data-driven, i.e., no processing code depends on language-independent facts. It consists of two discourse processing submodules (the Discourse Administrator and the Resolution Engine) and three discourse knowledge bases (i.e., the Discourse Knowledge Source KB, the Discourse Phenomenon KB, and the Discourse Domain KB). The Discourse Administrator is a development-time tool for defining the three discourse knowledge bases. The Resolution Engine, on the other hand, is the run-time processing module, which actually performs anaphora resolution using the discourse knowledge bases created by the Discourse Administrator.

Extracted Spanish and Japanese information on AIDS is stored in SYBASE database records in their native scripts. We make use of a SYBASE server that permits SQL queries to both the Japanese and Spanish databases, respectively, in order to retrieve information about the AIDS disease.

Evaluation provides crucial feedback to sponsors and developers alike. MURASAKI makes use of the MUC (Message Understanding Conference) evaluation software to score black-box data extraction results. We have extended the MUC software to handle Spanish and Japanese, and to interface with the object-oriented template structure used in the MURASAKI database.

Grammar evaluation is a new component of MURASAKI. We make use of the University of Pennsylvania Tree Bank bracketing tools to create model parses for Spanish and Japanese, and then automatically compare our system's syntactic output to the models in order to focus development efforts on key linguistic phenomena.

DATA EXTRACTION:
Measurement of Human Performance for Information Extraction

Craig A. Will
Institute for Defense Analysis
Boyan Onyshkevych
U.S. Department of Defense

As part of an effort to evaluate automated machine extraction of data from text in the DARPA TIPSTER project, analysts created a corpus of information extracted from newspaper articles in two domains: joint business ventures and microelectronics fabrication (Onyshkevych et al. 1994; Carlson et al. 1994). Each domain, in turn, used articles in two languages, English and Japanese. The corpus consisted of filled "templates," which were used both in the development of machine extraction systems and in the evaluation of the developed systems.

This rather extensive effort by analysts to encode templates made it possible to study the performance of humans for this task in some detail and to develop methods for comparing their performance with that of machines participating in the TIPSTER/MUC-5 evaluation (Will 1994; Okurowski 1994). This work complemented efforts to develop measurements that can objectively evaluate and compare the performance of different machine systems (Chinchor & Sundheim 1993) by also allowing comparisons with humans and thus providing better insight into the maturity and applicability of the technology.

The template structure was object-oriented. A template consisted of a number of objects, or template building blocks, with related information. Each object consisted of slots, which could be either fields containing specific information or pointers, i.e., references to other objects (Onyshkevych 1994).

Preliminary results are presented on the performance of humans in this task, including general baseline performance, variability among analysts, classification of errors or points of variation, and performance for different aspects of extraction. In general, the task was very difficult for both humans and machines. Substantial differences were observed between codings by different analysts, despite the fact that the analysts were highly skilled and had from 6 to 30 years of experience as professional analysts for U.S. Government agencies.

The differences could generally be divided into two categories: "errors" and "legitimate analytical differences." Errors included the following: (1) the analysts missed information contained in the article or erroneously interpreted its meaning; (2) the analysts forgot or misapplied a fill rule; (3) the analysts misspelled a word or made a keyboarding (typographical) error or an analogous error with a mouse; and (4) the analysts made an error in constructing the object-oriented structure, such as failing to create an object, failing to reference an object, providing an incorrect reference to an object, or creating an extraneous object. "Legitimate analytical differences" included the following: (1) different interpretations of ambiguous language in an article; (2) differences in the extent to which analysts were able or willing to infer information from the article that was not directly stated; and (3) different interpretations of a fill rule and how it should be applied (or the ability or willingness to infer a rule if no rule obviously applied).

To improve the quality and consistency of the extracted information, three steps were taken: first, a set of relatively detailed rules for extracting and structuring information were developed (the rules for the English Microelectronics extraction task, for example, came to a 40-page, single-spaced document). Second, a procedure was developed in which at least two analysts participated in coding most of the articles, with the different codings then reconciled to produce a final version. Third, software tools were developed that helped analysts to minimize errors, including a template-filling tool developed at New Mexico State University that provided a graphical interface that allowed analysts to easily visualize the relationships among objects and thus avoid errors in linking objects together. In addition to the templates created for the development and testing of machine systems, a small number of articles were coded by all analysts for the purpose of studying human performance on the extraction task.

The results of analysis of these codings on the English Microelectronics task showed that analysts produced about a 33% error rate when evaluated by comparing an analysts' first coding with the reconciled version produced by a different analyst. In contrast, the best machine system performed with an error rate of about 62%. Both the human and machine scores are highly reliable, as confirmed by statistical measures (Will 1994).

This level of performance suggests that machine extraction systems are still far from achieving high-quality extraction with the difficult texts and extraction problems characterized by the TIPSTER corpus. However, machine performance is close enough to the human level to suggest that practical extraction systems could be built today by careful selection of both the text and the extraction task, and perhaps by making use of integrated human-machine systems that can harness the abilities of both humans and machines for extraction rather than depending upon a machine-only system.

References

Onyshkevych, Boyan, Mary Ellen Okurowski, and Lynn Carlson. 1994. Tasks, Domains, and Languages for Information Extraction. *Proceedings of the TIPSTER Text Program, Phase One.* San Francisco: Morgan Kaufmann.

Carlson, Lynn, Boyan A. Onyshkevych, and Mary Ellen Okurowski. 1994. Corpora and Data Preparation for Information Extraction. *Proceedings of the TIPSTER Text Program, Phase One.* San Francisco: Morgan Kaufmann.

Will, Craig A. 1994. Comparing Human and Machine Performance for Natural Language Information Extraction: Results from the TIPSTER Text Evaluation. *Proceedings of the TIPSTER Text Program, Phase One.* San Francisco: Morgan Kaufmann.

Okurowski, Mary Ellen. 1994. Information Extraction Task Overview. *Proceedings of the TIPSTER Text Program, Phase One.* San Francisco: Morgan Kaufmann.

Chinchor, Nancy, and Sundheim, Beth. 1993. MUC-5 Evaluation Metrics. *Proceedings of the Fifth Message Understanding Conference.* San Francisco: Morgan Kaufmann.

Onyshkevych, Boyan A. Template Design for Information Extraction. *Proceedings of the TIPSTER Text Program, Phase One.* San Francisco: Morgan Kaufmann, 1994.

Will, Craig A., and Reeker, Larry H. "Issues in the Design of Human-Machine Systems for Natural Language Information Extraction." Presented at the 18-month TIPSTER meeting (Williamsburg, Va., February 22-24, 1993). Paper available from the authors.