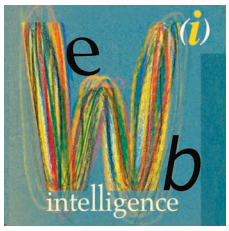
From Computational Intelligence to Web Intelligence



The authors explore three topics in computational intelligence—machine translation, machine learning, and user interface design—and speculate on their effects on Web intelligence.

Nick Cercone
Dalhousie University

Lijun Hou
University of Waterloo

Vlado Keselj
Dalhousie University

Aijun An
York University

Kanlaya Naruedomkul
Mahidol University

Xiaohua Hu
DMW Software

Systems that can communicate in natural ways and learn from interactions are key to long-term success in Web intelligence. By focusing directly on the Web, researchers in traditional computational intelligence topics can help in developing intelligent, user-amenable Internet systems.

The demands of the interactive, information-rich World Wide Web will challenge the most skillful practitioner. The number of problems requiring Web-specific solutions is large, and solutions will require a sustained complementary effort to advance fundamental machine-learning research and to incorporate a learning component into every Internet interaction.

NATURAL LANGUAGE INTERFACES

Natural language embodies important modalities for human-computer interactions, from simple database interfaces and machine translation to more general answer-extraction and question-answering systems.

In the 1990s, Simon Fraser University researchers worked on a project that culminated in SystemX, a natural language interface to relational databases.[1,2] Techniques pioneered in SystemX led to research in natural language access to Internet search engines. We describe two systems—one a natural language front end and the other a metasearch engine—to illustrate recent applications of this research.

Natural language front ends

Despite the many Internet search engines available, finding relevant sites remains difficult. One major problem is that search engines do not analyze queries semantically. Most simply perform keyword matching.

How can natural language semantics improve Internet searches? One common application of automated natural language understanding provides a "front end" that lets users access database information without needing to know the database structure or transform queries to another representation. NLAISE, the natural language access to Internet search engines,[3] is a front end that brings up the same results for a set of queries like "I want to book a flight," "Show me some sites on online reservations," or just "online reservation."

The user enters an English query that is transformed into a form for specific search engines. NLAISE lets the user choose the search engine best suited to the query, then analyzes the query syntactically and semantically. The system selects appropriate keywords for the information sought and interprets them to provide meaningful search terms, such as synonyms with Boolean operators.

NLAISE uses the Head-Driven Phrase Structure Grammar parser to generate a complex feature structure representing the query.[4] The HPSG parser extracts semantic content from the parsed query's feature structure, then interprets and transforms it into a form suitable for the selected search engine. For example, NLAISE generates the keywords "travel" and "Japan" when parsing "I want to schedule a trip to Japan."

A test inspection of 1,473 Web pages verified that

0018-9162/02/$17.00 © 2002 IEEE

80 percent of NLAISE-mediated queries were relevant.[3]

Metasearch extensions

A second system, English meta-access to Internet search engines (EMATISE),[5] extends NLAISE in three user-oriented ways:

- The search domains go beyond NLAISE's single "travel" domain and enhance semantic interpretation to eliminate ambiguity over multiple domains.
- Term-expanded queries go out to multiple search engines in parallel, and results return as a single relevant high-precision list.
- A single search interface provides users a higher level of abstraction than conventional search services.

As Figure 1 shows, EMATISE's metasearch engine has a modular design. It passes the logical query to the aggregation engine, which is responsible for concurrently dispatching the query to selected search services, obtaining initial results from each service, eliminating duplicate results, consolidating and reranking results, and creating HTML pages from the results for integrated display to the user.

As the Web grows, search services become volatile. Providers launch new services frequently and retire or replace others. They change interfaces to query input and results. EMATISE's modular design includes driver classes that provide a wrapper around service-specific information, encapsulating the services, and making modifications to them invisible to the user.

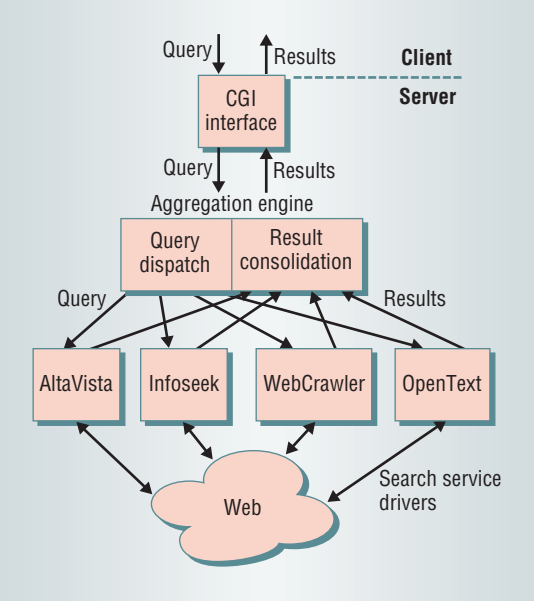


Figure 1. Modular metasearch engine architecture. Logical queries pass through a common gateway interface to the aggregation engine, which dispatches the query to selected search services.

MACHINE TRANSLATION

Traditional forms of machine translation fall into three categories: *direct*, *transfer,* and *intralingual*. These approaches either translate poorly, require resources that grow exponentially with the number of languages translated (multilinguality), or simplify language excessively. However, recent success in statistical, nonlinguistic, and hybrid machine translation suggest that systems based on these technologies can achieve better results, if a large enough annotated language corpus is available. We have developed *generate-and-repair machine translation*,[6] an extensible, modular approach to machine translation that systematically performs semantics-based translation preserving meaning between source and target languages.

As Figure 2 shows, GRMT comprises three phases: analysis-lite machine translation (ALMT), translation candidate evaluation (TCE), and repair and iterate (RI).

ALMT generates *translation candidates* by considering syntactic and semantic differences between language pairs. It generates the TCs simply and quickly, without performing a sophisticated analysis. TCE interprets the TC to see if it retains the meaning of the *source language*. If so, TCE considers the TC to be a translation. If not, it will diag-

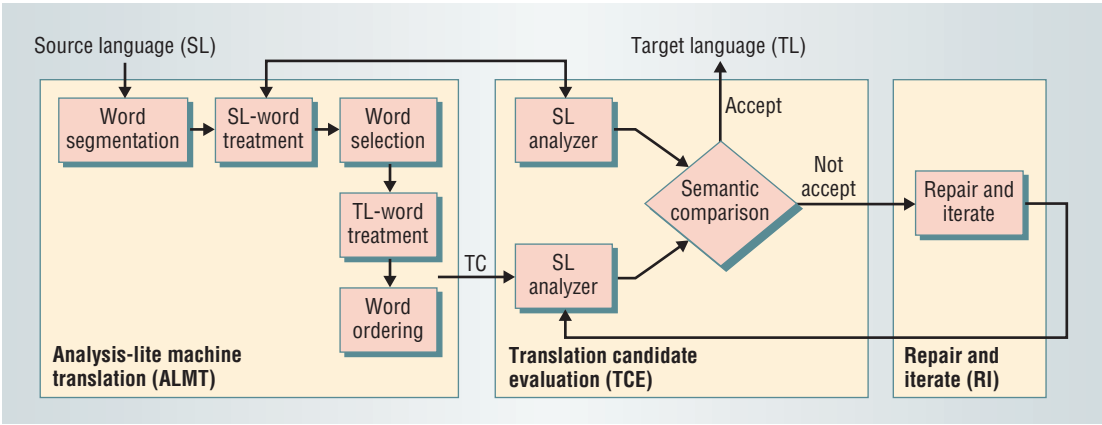


Figure 2. Generate-and-repair machine translation (GRMT) architecture. A three-phase process iteratively translates the source language to the target language.

Combining natural language processing with Internet multiagent systems may yield a new way to distribute processing costs.

nose the problem and send the TC to RI for repair. Following the repair, TC analyzer reinterprets the TC to determine if it still has a different meaning from the SL. These two processes iterate until the TC conveys the same meaning as the SL. The TCE and RI phases ensure translation accuracy.

GRMT treats the SL and *target language* separately and knows the differences between them. Therefore, if languages can be grouped according to various characteristics that they have in common, such as auxiliary verbs and continuous tenses, GRMT can perform the translation between groups. For example, let's say that Group 1 consists of English, French, and Spanish; and Group 2 consists of Chinese, Japanese, and Thai. The translation between these two groups using the traditional transfer approach requires six SL analyzers, six TL generations, and 18 sets of transfer rules. GRMT, on the other hand, requires six SL TCEs, six TL TCEs, and two sets of constraint applications.

Initial experiments (English to Thai) indicate that ALMT generates TCs with relative accuracy. For example, ALMT rearranged each phrase of an English sentence with four noun phrases, "An old woman lived in the cottage, with a fat black cat and a plump brown hen," into correct Thai grammatical order. In this case, the generated TC was also the correct translation. This is not always the case, but when TCE can diagnose an inappropriate TC, the system can replace it in the TCE and RI phases.

MACHINE LEARNING

ELEM2 is a rule-induction method that grew out of data-classification and rule-induction research at the University of Waterloo.[7-9] ELEM2 outperforms existing methods, for example C4.5, and has many applications, including Web page classification. For example, we have applied it in a data-classification, rule-induction application role for a commercial molecular compound database in which it predicts highly active molecular compounds.

ELEM2 induces decision rules from a set of observed data and classifies new examples by applying the induced rules.[7] ELEM2 is distinguished from other RI systems in its induction and classification processes. It uses a new heuristic function for evaluating attribute-value-pair relevance to a target concept during induction and for selecting the most relevant pairs for formulating rules. To handle inconsistent training examples, ELEM2 defines an unlearnable region of a concept according to the concept's probability distribution to the training data. It uses the unlearnable region as a stopping criterion for the concept-learning process, which resolves conflicts without removing inconsistent examples before rule induction. It also employs several continuous attribute discretization methods for learning classification rules and lets users choose the best one.

ELEM2 provides a new measure based on information entropy,[8] and it employs rule-quality measures to handle imperfect data by postpruning generated rules.[9] It defines measures according to relative rule distributions and chooses from four alternative kinds of distributions.

Rule-quality measures promise the greatest overall performance enhancement. We have used 12 rule-quality measures encompassing statistical or empirical formulas. Our studies show that ELEM2 provides more accurate classification results than the C4.5 and CN2 induction algorithms for both artificial and real data sets.[9]

WEB APPLICATIONS

Some of these solutions from computational intelligence are beginning to show up in Web applications.

Parsing natural language

Stefy is a natural language parser implemented in Java.[10] Based on HPSGs, Stefy is part of a larger project to implement a natural language system for Internet information retrieval. This IR task requires Java applets that can parse natural language. Java supports the dynamic class loading and object serializations that our concept of distributed natural language processing requires.

Stefy's approach is similar to the filtering techniques that improve traditional parser performance, but it differs by requiring the filtered knowledge to be in the form of a grammar. This approach is theoretically sound, and it provides a clean interface between subgrammar extraction and the parser.

Recent research on grammar modularity has focused on fast, accurate parsing of sublanguages. Motivation for this work includes potential improvements in managing the complexity of language expressions, parsing efficiency, and context-based disambiguation.

Recommender systems

Recommender systems suggest information sources, products, and services by learning from user preferences. Two methodologies dominate:

- *Collaborative (social) filtering* methods base recommendations on the preferences of users who made choices similar to the current user's; Amazon.com has used this method for years.
- *Content-based* methods use item-specified information that represents a unique niche across different product domains; wealth demographics is an example.

Future work in collaborative filtering can merge information sources to refine the analyses and subsequent recommendations.

Content-based recommender systems provide a unique application for embedded ELEM2. An application could use the information extracted from a set of documents—Web pages, newsgroup messages, and so on—during a word-extraction phase to develop a set of examples that serve as a user training set. The ELEM2 rule-induction process could then extract a user profile and rank the rest of the examples accordingly. The top-ranked examples then serve as items for recommendation. This process can help to personalize the recommender to individuals.

Agent systems

The Internet is a large, distributed, heterogeneous source of information. Users perceive it through a set of applications based on point-to-point TCP/IP communication links. Many of its applications require finding a relevant document or other point in the information space, which includes Telnet sites, newsgroup postings, FTP sites, and Web documents.

In natural language processing, we try to match the meaning of user queries to the meaning of retrieved documents. This involves deciding what a concept is, how to extract it from a natural language text, and how to match a concept with others that are similar. Existing systems are inefficient, but combining natural language processing with multiagent systems on the Internet may yield a new way to distribute processing costs—enabling natural language IR to keep pace with the growing wealth of information and resources currently available.

Adapting existing computational intelligence solutions may not always be appropriate for Web intelligence, but when it is, the solutions must incorporate a more robust notion of learning that will scale to the Web, adapt to individual user requirements, and personalize interfaces. ■

Acknowledgment

We thank the Natural Sciences and Engineering Research Council of Canada for its support of our work.

References

1. N. Cercone et al., "Natural Language Interfaces: Introducing SystemX," in *Advances in AI in Software Engineering*, T. Oren, ed., JAI Press, Greenwich, Conn., 1990, pp. 169-250.
2. P. McFetridge and N. Cercone, "Installing an HPSG Parser in a Modular NL Interface," *Computational Intelligence III*, North Holland, Amsterdam, 1991, pp. 169-178.
3. G. Mahalingam, *Natural Language Access to Internet Search Engines*, master's thesis, Computer Science Dept., Univ. of Regina, Regina, Saskatchewan, Canada, 1997.
4. C. Pollard and I. Sag, *Head-Driven Phrase Structure Grammar*, Univ. of Chicago Press, 1994.
5. L. Hou, *EMATISE: English Meta Access to Internet Search Engines*, master's thesis, Computer Science Dept., Univ. of Waterloo, Waterloo, Ontario, Canada, 1999.
6. K. Naruedomkul and N. Cercone, "Generate and Repair Machine Translation," to be published in *Computational Intelligence*, vol. 18, 2002.
7. A. An and N. Cercone, "ELEM2: A Learning System for More Accurate Classifications," *Proc. 12th Canadian Conf. Artificial Intelligence*, Lecture Notes in Artificial Intelligence 1418 (LNAI 1418), Springer-Verlag, Heidelberg, 1998, pp. 426-441.
8. A. An and N. Cercone, "Discretization of Continuous Attributes for Learning Classification Rules," *Proc. 9th Int'l Workshop Inductive Logic Programming* (PAKDD 99), Lecture Notes in Artificial Intelligence 1574 (LNAI 1574), Springer-Verlag, Heidelberg, pp. 509-514.
9. A. An and N. Cercone, "Rule Quality Measures for Rule Induction Systems: Description and Evaluation," *Computational Intelligence*, vol. 17, no. 3, 2001, pp. 409-424.
10. V. Keselj, *Modular Stochastic HPSGs for Question Answering*, doctoral dissertation, Computer Science Dept., Univ. of Waterloo, Ontario, Canada, 2002.

Nick Cercone *is a professor of computer science and dean of the computer science faculty at Dalhousie University, Halifax, Nova Scotia, Canada. His research interests include natural language processing, knowledge-based systems, knowledge discovery in databases, data mining, and design and human interfaces. Cercone received a PhD in com-*

puting science from the University of Alberta. He is a member of the ACM and the IEEE. Contact him at nick@cs.dal.ca.

***Lijun Hou** is a software developer in Waterloo. She received an M Math in computer science from the University of Waterloo. Her research interests include natural language interfaces to Internet search engines. Contact her at l2hou@ai.uwaterloo.ca.*

***Vlado Keselj** is an assistant professor of computer science at Dalhousie University, Halifax, Nova Scotia, Canada. His research interests include natural language processing, modular stochastic HPSGs for question answering, and information retrieval and extraction. He received a PhD in computer science from the University of Waterloo. Contact him at vlado@cs.dal.ca.*

***Aijun An** is an assistant professor in the Department of Computer Science at York University, Toronto, Ontario, Canada. Her research interests include data mining, machine learning, Web mining, and bioinformatics. She received a PhD in computer science from the University of Regina. Contact her at aan@cs.yorku.ca.*

***Kanlaya Naruedomkul** is an assistant professor and the committee chair of the MSc applied mathematics program at Mahidol University, Bangkok, Thailand. Her research interests center on artificial intelligence applications, including automated natural language processing, computational linguistics, and machine translation. Naruedomkul received a PhD in computer science from the University of Regina. Contact her at scknr@mucc.mahidol.ac.th.*

***Xiaohua Hu** is an assistant professor at Drexel University and founder and president of DMW Software. His research interest is data mining. Hu received a PhD in computer science from the University of Regina. Contact him at xiaohua_hu@hotmail.com.*