Web Intelligence Meets Brain Informatics at the Language Barrier: A Procrustean Bed?

Nick Cercone

Faculty of Science & Engineering
York University
Toronto, Ontario, Canada
ncercone@yorku.ca

Abstract. We take it for granted that computers hold answers to our questions, our information requirements, our needs over the past twenty five years we have learned much about language, about databases, and about how people interact with computers; researchers have made great strides in the construction of human computer interfaces which (relatively) seamlessly integrate modalities, for example, speech and written language, natural language and menu systems, and so on. The next generation of interfaces and browsers, in order to be considered successful, must do more: they must individualize frameworks of meaning in order to provide relevant timely responses to information requests. I want to make several points, perhaps circuitously, but directed as examining some basic tenets regarding our faith in machines. I direct your attention to several problems inherent in representation(s) required to place information into machines for easy (individualized) access, followed up by some larger questions about the inherent capabilities of machines (versus humans).

1 Introductory Remarks

I am bemused. In part, because reflection as broadly as is necessary to make sense what it means for Web intelligence to meet brain informatics, to individualize and comprehend frameworks of meaning in our rapidly changing environment, requires time. In part, because the same blessing that allows each and every one of us to reach out for information instantly is also the curse of responsibility: we must decide how, why and when to use this marvelous blessing.

When I was 8 years old and growing up in Pittsburgh, my friends and I would often sleep on our enclosed front porches in the swelter of the summer's evening heat, gently rocking to sleep on the large porch swing suspended by

N. Zhong et al. (Eds.): WImBI 2006, LNAI 4845, pp. 59–72, 2007.

chains from the porch roof. Sometimes we would wake in the middle of the night and climb the neighbor's mulberry and peach trees for midnight snacks, only to always have our actions discovered by our mothers in the morning. It took some time for us to realize that mulberries often left telltale purple marks around our mouths after a late night feast. On other nights, I would wander, by myself, to Moore Playground a kilometer away and lie in the centre field grass in the big playground ball field where the semipros played. At midnight no games were being played and the stands were deserted. I wondered then, as I looked up at the stars (yes the renaissance had occurred in Pittsburgh by then after years of industrial neglect of the environment and you could actually see the stars at night). I pondered the questions then that have basically bothered me ever since: How far was far and how big was big? I came to believe that as a grown-up I would ultimately know the answers to these questions. Much later, as a young man at a small Franciscan College in Ohio, I took just about every philosophy course that would fit into an engineering science curriculum that was possible, still hoping to find some answers to these and similarly important questions.

How true it is that language creates special worlds. And 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 (artificial) intelligence areas 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.

Still, 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.

The editors of this volume would have us believe that "the synergy between Web intelligence (WI) with brain informatics (BI) will yield profound advances in our analyzing and understanding of the mechanism of data, knowledge, intelligence and wisdom, as well as their relationship, organization and creation process." Our use of language should put this hypothesis to the test.

2 Language and Artificial Intelligence

Most languages are inaccessible to most of us most of the time[1] - we believe that the language of the Eskimo and Inuit describing the many states of snow

[1] Lewis Carroll once wrote: "I'm so glad that I don't like asparagus" said the small girl to a sympathetic friend. "Because, if I did, I should have to eat it, and I can't bear it."

is inaccessible[2], but we need not feel a loss. Generally what holds for language holds for life also. It is in this sense that language can serve as a mirror in our investigations into the nature of cognitive capabilities so necessary for Web intelligence. For it is not so much why something happens, or how it occurs, as it is to understand why we perceive things to be the way that they are or how we plan activities to occur or even what we ruminate in between all other thoughts that generally holds our interest.

However, for most of us, the world is a world of matter - wysiwyg.[3] The superiority of physics to, say, interpersonal communication, massage, etc. derives from the assumption that if we are able to explain the physical, we may be in a position of explaining everything else.

So where is artificial intelligence, Web intelligence, brain informatics situated? In the days of good old fashioned AI [GOFAI], the quest was to find a general intelligence. Representation schemes were recognized as important, serving as the structure(s) by which the systems we built (we did build large programs rather routinely then) could be extended to cover more and more of a particular domain and, if we were lucky, extend to another domain as well. Then something curious appears to have happened. A necessary step in the name of progress in artificial intelligence was to stop experimenting, and start becoming smarter about what systems we were going to build. We began designing various logics for specific purposes, never yet getting back to connecting them all up with the original quest for finding a more general intelligence (surely this is what Turing had in mind when he proposed that extremely boring party game known now as the Turing test - why I haven't even heard that term used for about 10 years now). Is history repeating itself?

Consider the fine earlier work in theorem proving research: after a while researchers gave up trying to use results of this research as the basis of natural language understanding programs. Now we find theorem proving research results embedded in constraint logic programming systems which are at the heart of many fine natural language efforts. When computer science was born, arguably from numerical analysis research, every computer science department

[2] Geoffrey K. Pullam's book "The Great Eskimo Vocabulary Hoax and Other Irreverent Essays on the Study of Language", 1991, University of Chicago Press] on the great eskimo vocabulary hoax, notwithstanding. Pullam actually wrote"Once the public has decided to accept something as an interesting fact, it becomes almost impossible to get the acceptance rescinded. The persistent interestingness and symbolic usefulness overrides any lack of factuality. . . . In the study of language, one case surpasses all others in its degree of ubiquity, and the present chapter is devoted to it; it is the notion that Eskimos have bucketloads of different words for snow. . . . But the truth is that the Eskimos do not have lots of different words for snow, and no one who knows anything about Eskimo (or more accurately, about the Inuit and Yupik families of related languages spoken by Eskimos from Siberia to Greenland) has ever said they do. Anyone who simply insists on checking their primary sources will find that they are quite unable to document the alleged facts about snow vocabulary (but nobody ever checks, because the truth might not be what the reading public wants to hear).

[3] What you see is what you get.

had numerical analysis courses, teachers and researchers. It would be heresy not to have them. Then the unthinkable happened. The numerical analysts became part of applied mathematics departments in a large part of the university systems and pursued more and more fine grained results studying error analysis to the mth degree. Not only could your space vehicle miss its intended destination, but we could tell you exactly by how much you missed it. Over the last decade, numerical analysis has been reborn and its importance reaffirmed, especially as the pendulum of university research swings to more and more applied research.

Is this what is happening to the quest for generality. Sadly, I think the situation is different in this case. Essentially, I am becoming a skeptic, not of the value of artificial intelligence but of the time frame we have given ourselves to produce an artificial intelligence, hence the importance to Web intelligence, and of the need I am convinced of, of encouraging truly multidisciplinary teams of researchers to tackle some of our problems. Let me cite some examples from natural language understanding, admitting that in each case, we could probably devise a system to tackle the particular problem, but generalizing the solution may well prove elusive.

It is common in the best of the Yiddish tradition to answer a question with a question, often the very same question with different emphasis and intonation. For example, in response to the question "Did you buy flowers for your mother on her birthday?" the response would be quite different if a different word were emphasized in the answer. Thus the answer "*Did* I buy flowers for your mother on her birthday?" is quite different from "Did *I* buy flowers for your mother on her birthday?" which is different from "Did I *buy* flowers for your mother on her birthday?" and so on.

Imagine the reasoning required to interpret the following passage and correctly ask the question which Mr. Rosenberg asks of Murray Goldwag at the end of the passage.

> One fine Friday afternoon, Murray Goldwag leaves his place of employment in Brooklyn to catch the bus to Wappinger's Falls upstate to spend the weekend with his finance Lennie Rosenfeld. During the trip and elderly gentleman, Mr. Rosenberg, returning to his home in Wappinger's Falls from a visit with his brother in Brooklyn, strikes up a conversation with Murray. "So, you're up to the Falls for the weekend"? "Yes sir", replied Murray. "By yourself?" "Just visiting friends", replied Murray. Mr. Rosenberg then reflects for a while. He is visiting friends, leaving early enough to arrive in time for a late dinner and to get himself a motel room. He must be visiting a girl friend since no one would go out of their way so much for another boy friend. Now who could he be visiting? He is a handsome lad. Could it be the Goldberg twins? - No, Meryl is away at University out of state and Fran is out of town visiting relatives. Could it be Maxine Kriebel? - No, she has a boyfriend. What about Melinda Eaman? Probably not, she is very rich and would not be going out with any young man who had to travel by bus in order to see her. Well, what about Sarah Lavie? That could be the one, Sarah has been on cloud nine recently and acting mysterious, visiting all the shoppes and making preparations for a big day soon. She is not graduating from College, nor celebrating a promotion or new job. That's it, thought Mr. Rosenberg, Murray must be coming to Wappinger's Falls to see Sarah and make plans for their upcoming wedding. Mr. Rosenberg turns to Murray and says "Congratulations, son, on your upcoming

wedding to Sarah Lavie.” “wha–what”, said Murray. “How did you know, we haven’t even told her parents yet.” “My boy”, said Mr. Rosenberg, “its obvious.”

These two examples of sublanguages of English occur with alarming frequency and we are equipped to handle these and other statements which we have never heard before as examples of “simply miraculous machines.” Pity the poor computer, however. If you do not like these examples, I am sure that I could supply hundreds of other examples, perhaps more straightforwardly, but equally obtuse to our efforts thus far in natural language understanding. Consider the following excerpt from Erle Stanley Gardner’s “The Case of the Demure Defendant”:

“Cross-examine,” Hamilton Burger snapped at Perry Mason.

Mason said, “Mr. Dayton, when you described your occupation you gave it as that of a police expert technician. Is that correct?”

“yes sir.”

“What is an expert technician?”

“Well, I have studied extensively on certain fields of science that are frequently called upon in the science of criminology.”

“That is what you meant by an expert technician?”

“Yes sir.”

“Now what is a police expert technician?”

“Well that means that... well, it all means the same thing.”

“What means the same thing?”

“An expert technician.”

“An expert technician is the same as a police expert technician?”

“Well I am in the employ of the police department.”

“Oh the police employ you as an expert witness, do they?”

“Yes sir, ...I mean no, sir. I am an expert investigator, not an expert witness.”

“You are testifying now as an expert witness are you not?”

“Yes sir.”

“Then what did you mean by saying you were an expert technician but not an expert witness?”

“I am employed as a technician but not as a witness.”

“You draw a monthly salary?”

“Yes.”

“And you are being paid for your time while you are on the stand as an expert witness?”

“Well, I’m paid for being a technician.”

“Then you won’t accept any pay for being a witness?”

“I can’t divide my salary.”

“So you are being paid?”

“Of course - as part of my employment.”

“And are you now employed by the police?”

“Yes.”

“And are you an expert witness?”

“Yes.”

“Then you are now being employed as an expert witness.”

“I guess so. Have it your own way.”

“When you described yourself as a police expert technician that means your testimony is always called by the police. Isn’t that so?”

“No, sir.”

“Who else calls you?”

“Well, I ...I could be called by either party.”

“How many times have you been on the witness stand?”

“Oh, I don’t know. I couldn’t begin to tell you.”

“Dozens of times?”

“Yes.”

“Hundreds of times?”

“Probably.”

“Have you ever been called by the defense as a defense witness?”

“I have not been directly subpoenaed by the defense. No, sir.”

“So that you have always testified for the police, for the prosecution?”

“Yes, sir. That’s my business.”

“That was what I was trying to bring out,” Mason said. ...

Mr. Dayton needs to understand the subtleties of noun phrases such as “police expert technician”, to answer Mr. Mason’s questions. Understanding such phrases are troublesome to automate since “police”, “expert” and “technician” are all nouns.

Generalizing semantic considerations for such constructions have proven evasive. The compositional approach to natural language understanding favored by logic grammarians becomes combinatorially explosive. Many researchers represent noun-noun constructions as single lexical entries, constraining the computation required to disambiguate them, and circumventing an annoying semantics problem.

When the domain of discourse is well specified and the number of such phrases is small, this approach works adequately. But is it practical? Consider “western region outage log” employed by telecommunications service personnel. Would the designer of their system resort to separate lexical entries for “eastern region outage log”, “southern region outage log”, “northern region outage log”, “northeastern region outage log”, ..., “western district outage log”, ..., “western prefecture outage log”, ..., “western region service log”, ..., “western region outage record”, ...?

Imagine further, the processing required by Perry Mason. Not only must the subtleties of language understanding realized by Mr. Dayton be mastered but also the reasoning capabilities of Mr. Mason and extraction of relevant and salient features of the conversation be identified in order to generate the appropriate next question. Actually, Mr. Mason’s task is simpler than Mr. Dayton’s - to generate an utterance which conveys a presumably preexisting thought. Mr. Dayton’s task as listener is to decide what Mr. Mason must have been thinking in order to motivate his utterance in the particular context in which he uttered it.

Language is difficult; humans are amazing. By the time you have completed reading this sentence you will have understood its meaning. Your achievement and success in understanding is most impressive. The speaker’s task is much simpler - to generate an utterance which conveys a presumably preexisting thought. Your task as listener is to decide what the speaker must have been thinking in order to motivate his utterance in the particular context in which he uttered it. In general, understanding a natural language (NL) is simply miraculous.

NL represents an important modality for human computer interactions, from simple NL interfaces to databases to machine translation to more general answer-extraction and question answering systems. Other important modalities, e.g., speech, pointing devices, graphical user interfaces, etc. remain. The perfection and integration of multimodal systems takes on new importance when we transpose previous solutions to the Internet. Systems which can communicate in natural ways and can learn from interactions are key to long term success transferring *computational to Web intelligence to brain informatics.*

3 Traditional Natural Language Applications

How long will it be before we have systems that can process language as illustrated in the three examples above? It will be instructive to look at the past.

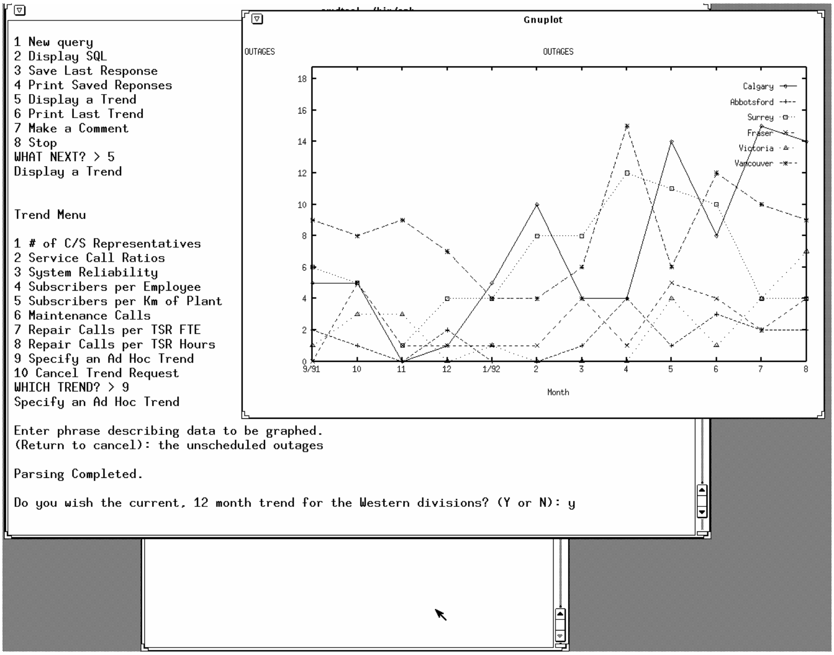
At Roger's Cablesystems Ltd., the vice president for customer service enters the following into his computer terminal, "Give me the Western region outage log for June". Within seconds SystemX [1,2] presents him with a neatly formatted table (or graph) of the data retrieved from Rogers' relational database. He could have said, "What's the outage log for the Western region for June?", or "Tell me the June regional outage log for the West." or "Find the Western outages for June.", etc. SystemX can determine that whichever phrase he uses, he means the same thing. Such flexibility in parsing, applying the logical rules of grammar to determine meaning, is nontrivial. SystemX's parsing techniques are described in [3]. After parsing, SystemX reformulates the question in SQL (structured query language) and data is extracted for presentation from Roger's large central database.

The nontrivial problem described in the preceding paragraph is but one of a large number of very difficult problems of understanding NL by computer. Fortunately, a NL interface is simpler to comprehend. Although one ultimately encounters problems comparable to the unconstrained NL understanding situation, the domain of discourse, and thereby the context, is highly constrained by the database schema. General analysis of language phenomena and much of the ambiguity inherent in NL understanding is limited but complexities arise when building NL capabilities into database interfaces. One quickly comes to realize that domain knowledge is required in order to interpret queries, in order to answer queries, and that modeling the user is important as well.

An example of SystemX accepting an English query from Rogers' vice president, translating the query into SQL, retrieving data from Rogers' database and displaying the data in the format (table or graphical trend) specified by the user in the query is shown in Figure 1. SystemX is able to display responses to requests for trends in statistical data graphically. The user has the choice of inputting his trend request using English, using menus (in the case of "canned" trends) or using a combination of English and menu responses. Various input modalities are provided as a convenience to users. The "canned" trends display data that is predictably desired on a reasonably frequent basis. They may be accessed for a minimum of keystrokes. The "canned" trends are those available through the first eight menu items in the Trend Menu in Figure 1.

Specifying a request for a trend in English may become quite cumbersome if default parameters (specifying timing and so forth) are not employed. The complex statements required are difficult to formulate and demand patience on the part of the users while waiting parsing. The system therefore allows the users to request ad-hoc trends using a combination of English and responses to menus. This combination of modality reduces the task of specifying a complex query into a set of simple tasks that are accomplished in sequence. The system accesses the database in order to be able to present tasks to users in as helpful a manner as possible.

Despite the many search engines available, searching for a relevant site remains difficult. One major reason for this difficulty is that search engines do not



```
1 New query
2 Display SQL
3 Save Last Response
4 Print Saved Reponses
5 Display a Trend
6 Print Last Trend
7 Make a Comment
8 Stop
WHAT NEXT? > 5
Display a Trend


Trend Menu

1 # of C/S Representatives
2 Service Call Ratios
3 System Reliability
4 Subscribers per Employee
5 Subscribers per Km of Plant
6 Maintenance Calls
7 Repair Calls per TSR FTE
8 Repair Calls per TSR Hours
9 Specify an Ad Hoc Trend
10 Cancel Trend Request
WHICH TREND? > 9
Specify an Ad Hoc Trend

Enter phrase describing data to be graphed.
(Return to cancel): the unscheduled outages

Parsing Completed.

Do you wish the current, 12 month trend for the Western divisions? (Y or N): y
```

Fig. 1. The SystemX main menu and a trend menu

analyze queries semantically; in contrast, most search engines perform keyword matching.

How can our use of NL semantics improve Internet searching? SystemX was one common application that provided a NL "front-end", which enables users to access database information without any need to know database structure or any query language, and with no need for query transformation to some other representation. A NL "front-end" to Internet search engines, which allows users to utilize search engines without finding appropriate search terms, is presented in [4,5]. For a search for: "I want to book a flight ticket" or "Show me some sites on online reservation of flight tickets" or phrases like "online reservation of flight tickets", these queries would yield the same search results.

NLAISE [4] allows users to choose the search engine best suited for their search and enter the query in English. The NL query is analyzed both syntactically and semantically in order to select the most appropriate keywords describing sought information. Keywords are interpreted to provide more meaningful search terms by using keyword synonyms in conjunction with Boolean operators supported by specific search engines.

In NLAISE, the NL query, along with the choice of search engine, is preprocessed in order to transform the query into a form suitable for input to the parser. The parser, in turn, has a description of grammar rules for capturing the constraints of English and a lexicon that contains the words permitted as input. The Head Driven Phrase Structure (HPSG) parser generates a complex feature structure representing the query. The semantic content of such a complex feature structure is extracted, interpreted and transformed into a form suitable for the search engine that was selected. In a test NLAISE was asked to parse the phrase “I want to schedule a trip to Japan” and generate appropriate keywords for search engine examination. NLAISE was also requested to use Infoseek as the search engine. Inspection of the 1,473 Web pages returned verified that 80% were relevant. Note the choice of keywords “Japan” and “travel” indicates the level of sophistication of NLAISE’s semantic interpretation of the original input phrase.

EMATISE [5] extended NLAISE in 3 user-oriented ways: (1) whereas NLAISE was tied to a single “travel” domain, EMATISE greatly enhanced semantic interpretation to eliminate much ambiguity and toil over multiple domains; (2) EMATISE sent out term expanded queries to multiple search engines in parallel and reranked results returned from these search engines into a single relevant high precision list for the user; and (3) EMATISE’s higher level of abstraction above conventional search services presented the user with a single, central and natural search interface with which to interact.

Consider the following scenario. Imagine picking up the phone in Toronto, dialing your Japanese program co-chairman in Tokyo to explain several papers lost in the shuffle of email systems. You speak English and she speaks Japanese. Fortunately it is 2010 and the English you speak in Toronto is automatically translated into Japanese in the time it takes to transfer your words over the phone lines. Impossible, - probably not. The world of machine translation has both fascinated and frustrated researchers for over 50 years. Recent success in statistical, nonlinguistic and hybrid systems provide hope that we will not be confined to traditional dominant direct, transfer and intralingual approaches. An informative critique of these approaches is given in [6]. We provide an approach following from CS methodology, generate and repair machine translation. (GRMT).

GRMT (Figure 2) is composed of 3 phases: “Analysis Lite Machine Translation (ALMT)”, “Translation Candidate Interpretation (TCI)” and “Repair and Iterate (RI)”. “ALMT” generates translation candidates (TC) by considering syntactic and semantic differences between language pairs without performing any sophisticated analysis. This ensures that the TC can be generated quickly, simply and efficiently. Next, the system interprets the TC to see if it retains the meaning of the SL. If so, that TC will be considered a translation. If not, that TC will be repaired based on the diagnosis that is indicated in the second phase, TCI. Subsequently the repaired TC will be re-interpreted 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 TCI and RI stages ensure the accuracy of the translation result. They also guarantee the accuracy of the translation back from the TL to SL.

GRMT treats SL and TL separately and is aware of the differences between languages. Therefore, if languages can be grouped according to various characteristics, e.g., auxiliary verb, continuous tenses, passive voice, etc., which they

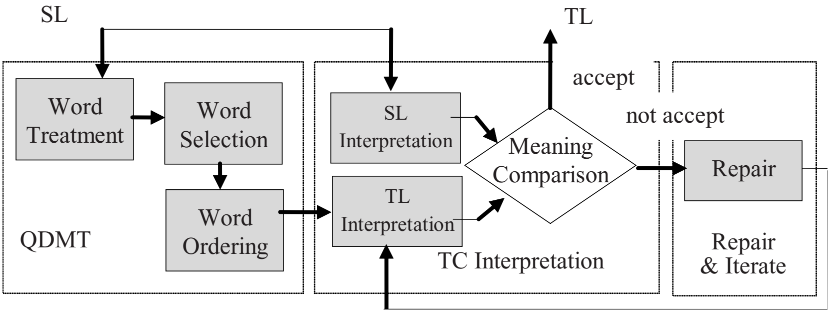


Fig. 2. Organization of Building Block

have in common, then the translation between groups can be performed more simply by GRMT. For example: Group 1 consists of English, French and Spanish, Group 2 consists of Chinese, Japanese and Thai. To perform the translation between these two groups, the transfer approach requires six SL analyzers, six TL generations and 18 sets of transfer rules while GRMT requires six SL TCIs, six TL TCIs and 2 sets of constraint applications.

Table 1. Example of a Generated TC

Example: An old woman lived in the cottage, with a fat black cat and a plump brown hen.

TC: ผู้หญิงแ ก่ คนหนึ่งไ ด้ อยู่ ใน กระท่อม หลังน ั้นก ับแ มว สีดำ อ้วน ตัวห นึ่งแ ละ ไก่ สีน้ำตาล อวบ ตัวห นึ่ง

CT: ผู้หญิงแ ก่ คนหนึ่งไ ด้ อยู่ ใน กระท่อม หลังน ั้นก ับแ มว สีดำ อ้วน ตัวห นึ่งแ ละ ไก่ สีน้ำตาล อวบ ตัวห นึ่ง

(phûujiˇ- woman) (k`- old) (khon- clas) (ny`- an) (dâj- past) (ju`u- live) (naj- in) (krathˆm- cottage) (laˇ- clas) (nán- the) (kàb- with) (mw- cat) (siˇidam- black) (?ûan- fat) (tua- clas) (ny`- a) (l`- and) (kàj- hen) (siˇinámtaan- brown) (?ùab-plumb) (tua- clas) (ny`- a)

Experiments of ALMT (English to Thai) indicate that TCs can be generated with relative accuracy. Table 1 shows an example of applying ALMT.

4 Brainstorms

Successes mentioned earlier, and others like them, represent contemporary computational intelligence solutions. How do we adapt them to become Web intelligence and brain informatics solutions? We briefly describe current work designed to make useful solutions to computational intelligence problems amenable to such use. Some of this work takes advantage of newer technologies already beginning to show up in Web applications (agent architectures. recommender systems, information extraction tools, etc.). This current work represents an intermediate step along the way to Web intelligence/brain informatics. It necessarily leads to the realization that more adaptable and more general machine learning strategies need developed and incorporated into every aspect of systems. One glaring example would be learning the meaning of unknown or undefined words, for machine translation and general speech and NL processing.

Java Parsers, Just-in-time Subgrammar Extraction, Modular HPSG's

Stefy is a NL parser implemented in Java, based on HPSGs [7], It is part of a larger project to implement a NL processing system for Internet information retrieval (IR). This IR task requires Java applets capable of parsing a NL. Earlier we discussed work on developing HPSG parsers. However, Stefy is one of the first implemented in Java. Java was chosen for two reasons. Java supports dynamic class loading and object serialization, which are important features necessary for our concept of distributed NL processing. Java is a good prototyping language, compared to C++ for example, and facilitates easy experimentation with various approaches, which makes this shift in programming language paradigm less drastic.

A drawback of our implementation is that it is not suitable for development of the grammar and lexical resources. Other systems, like ALE [8] and LKB [9], are more appropriate for this task. After a grammar or a lexicon is developed in one of those systems, it is translated into a Java description and used in Stefy.

Stefy represents a new precise and compact description of the HPSG formalism, which is especially suitable for implementation of HPSG parsers in low-level languages. Stefy represents an important step towards applying HPSG formalism in the area of distributed NLP and answer extraction.

Stefy's approach is similar to the filtering techniques, which are a recognized way to improve parser's performance. However, Stefy is different because we insist that the filtered, i.e., extracted, knowledge is in the form of a grammar. This approach is sound, and in practice it provides a clean interface between subgrammar-extraction part and the parser. Keselj gives more arguments for this separation of the subgrammar extraction and parsing [10].

An important part of the HPSG subgrammar extraction is the extraction of the corresponding type sub-hierarchy out of the original hierarchy. Efficient type operations and representation of the types are used in approximate algorithm for subgrammar extraction for HPSGs. Recently, there has been a lot of research activity in the area of grammar modularity. Some of the motivational factors for this work are the following:

- managing complexity. The NL grammars used in NL processing are large and complex. The difficult problems are designing, creating, testing, and maintaining them. Using smaller modules that are combined into larger grammars addresses the complexity problem.
- parsing efficiency. Parsing with a large, wide-coverage grammar is typically not efficient. Quickly extracting a small subgrammar module, and then using it to parse the text can reduce the running-time and space requirements.
- context-based disambiguation. By having a larger grammar we achieve a better coverage, but in the same time it becomes susceptible to ambiguities. Any NL is very ambiguous, and it is well known that humans use world-knowledge and contextual knowledge to do disambiguation. Extracting a subgrammar based on the text to be processed can be viewed as creating a context that can improve disambiguation.

Recommender Systems using ELEM2

Recommender systems suggest information sources, products, services, etc., to users based on learning from examples of their preferences, dislikes, etc. There are two predominant methodologies employed in such systems. *Collaborative (social) filtering methods* base recommendations on other users preferences, e.g., when you order books from Amazon.com, the recommender system may detect other customers who ordered the same books and determine other orders placed by these customers to then enquire whether you may also be interested in acquiring similar material. *Content-based methods* use information about the item ordered/specified in order to make further suggestions to the user. Advantages of content-based methods include the ability to recommend previously unrelated items to users with unique interests and to also provide explanations for recommendations.

For collaborative (social) filtering, we plan to merge information sources to permit more fine-grained analysis and subsequent recommendations. For example, use of the Statistics Canada database on wealth demographics in Canada, which they categorize from richer to poorer by postal code, could conceivably recommend products/services based not only on social preference but also by wealth demographics at the same time.

We especially wish to develop content-based methods since this will provide a new application for ELEM2 [11,12]. Content-based recommender systems provide another unique application for embedded ELEM2. Briefly, a set of documents (Web pages, newsgroup messages, etc.) would have information extracted from an information extraction (word extraction) phase to develop a set of examples. We randomly select a set of examples and choose a subset of these examples from which we determine from a user, positive and negative examples. These positive and negative examples serve as a training set for the user. We apply ELEM2 rule induction process to extract a “user’s profile” and then rank the rest of the examples accordingly. Top ranked examples then serve as an item list for recommendation.

Agents and Agent Architectures

The Internet is a large, distributed, and heterogeneous source of information primarily consisting of on-line World Wide Web documents. It is perceived through a set of applications based on the point-to-point communication links provided by the TCP/IP protocol. Many applications frequently end up with the problem that we want to find a relevant document, relevant item, or, generally, a relevant point in the information space consisting of Telnet sites, news groups, news group postings, FTP (File Transfer Protocol) sites, and WWW documents (pages, movies, radio broadcasts). How can we find out if someone has an e-mail address and how can we find that address? Finding interesting mailing lists is a still better example.

The Internet can be imagined as a low-level structure activated with considerable manual (human) participation. Such an intelligence-assuming environment requires computational intelligence management techniques. The most obvious example is a simple Web page. If we want to automatically use its content in a fashion more sophisticated than collecting keywords, or collecting embedded links for further navigation, then the most flexible, robust, and appropriate way to do this is to understand some of its content and to reason about it. This is the realm of computational intelligence.

"Agent" has become a computational intelligence term, and a frequent buzzword having a wide range of definitions. Nevertheless, there are some common characteristics that describe an agent. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment. Furthermore, the development of multi-agent systems is based on work in two areas artificial intelligence and distributed systems.

The combination of NL processing and multi agent system's is still quite novel and often the terms are used independently. Consider the use of NL processing for information retrieval (IR) over the Internet. This is an attempt to match the meaning of the user's query to the meaning of retrieved documents. Since this approach relies on higher levels of NL processing, it is difficult to implement. Issues include deciding what is a concept, how to extract concepts from NL texts, and how to do concept matching. The inefficiency of existing NL processing systems is a major obstacle to using them in IR. If we want to use an NLP system to analyze the documents in a large document collection, it has to be efficient and robust to be useful in practice.

A positive approach is to implement distributed NL processing so that the processing cost is widely distributed in the same way as are Internet resources. Multi agent systems are appropriate for this task.

5 Concluding Remarks

Web intelligence/brain informatics requires further research and development into the technologies discussed above and other technologies as well. Adapting existing computational intelligence solutions may not always be appropriate for Web intelligence for a number of reasons, e.g., the magnitude of information available on the Internet and the additional requirements for speedy processing. Computational intelligence solutions which may be adapted must incorporate a more robust notion of learning in order for these solutions to scale to the Web, in order for these solutions to adapt to individual user requirements, and in order for these solutions to personalize interfaces.

We have only briefly touched on a few, albeit important, issues that will be the mainstay of Web intelligence in the near term future. Users will demand access to the Internet that is simple (multimodal interfaces), with language/speech capabilities - both comprehension and, when needed, translation - and personalized (multi agent architectures) Internet use which "learns".

How soon might we expect to see breakthroughs? One way of considering this question is to recognize that research progress is highly incremental, thus, we are seeing progress every day. I, for one, have great hopes for the future of Web intelligence.

Acknowledgments

The author wishes to acknowledge the support of the Natural Sciences and Engineering Research Council, of Canada.

References

1. Cercone, N., Han, J., McFetridge, P., Popowich, F., Cai, Y., Fass, D., Groeneboer, C., Hall, G., Huang, Y.: SystemX and DBLearn: easily getting more from your Relational Database. invited for Integrated Computer-Aided Engineering. In: Golshani, F. (ed.), vol. 1(4), pp. 311–339 (1994)
2. Popowich, F., McFetridge, P., Fass, D., Hall, G.: Processing Complex Noun Phrases in a Natural Language Interface to a Statistical Database. In: Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992), Nantes, France, pp. 47–52 (1992)
3. McFetridge, P., Cercone, N.: Installing an HPSG Parser in a Modular Natural Language Interface, Computational Intelligence III, pp. 169–178. North Holland, Amsterdam (1991)
4. Mahalingam, G., Cercone, N.: Finding Information Easily is Important for e-Business. In: Kou, W. (ed.) Data Warehousing and Data Mining for Electronic Commerce, pp. 135–168. IBM Press (1999)
5. Hou, L., Cercone, N.: Extracting Meaningful Semantic Information with EMATISE: an HPSG-based Internet Search Engine Parser. In: NLPKE Symposium of the IEEE SMC Conference, Tuscon, AZ (2001)
6. Naruedomkul, K., Cercone, N.: Generate and Repair Machine Translation. Computational Intelligence 18(3), 254–270 (2002)
7. Keselj, V.: Stefy: Java Parser for HPSGs, Version 0.1, Technical Report CS-99-26, Department of Computer Science, University of Waterloo, Waterloo, Canada (2000)
8. Carpenter, B., Penn, G.: ALE, the attribute logic engine, User's Guide (1999), `www.sfs.nphil.unituebingen.de/~gpenn/ale.html`
9. Copestake, A.: The (new) LKB system, Version 5.2 (1999)
10. Keselj, V.: Just-in-time subgrammar extraction for HPSG, Technical Report CS-2001-08, Computer Science, University of Waterloo, Waterloo, Canada (2001)
11. An, A., Cercone, N.: ELEM2: A Learning System for More Accurate Classifications. In: Mercer, R.E. (ed.) Advances in Artificial Intelligence. LNCS, vol. 1418, pp. 426–441. Springer, Heidelberg (1998)
12. An, A., Cercone, N.: Discretization of Continuous Attributes for Learning Classification Rules. In: Zhong, N., Zhou, L. (eds.) Methodologies for Knowledge Discovery and Data Mining. LNCS (LNAI), vol. 1574, pp. 509–514. Springer, Heidelberg (1999)