

Real-world Requirements for Natural Language Interfaces for E-Commerce

From: AAAI Technical Report WS-99-01. Compilation copyright © 1999, AAAI (www.aaai.org). All rights reserved.

Mallory Selfridge

Department of Computer Science and Engineering
University of Connecticut
Storrs, CT 0624
mal@engr.uconn.edu

As e-commerce continues to expand, the potential utility of natural language communication with e-commerce applications becomes increasingly apparent. While clickable web pages will continue to support the great majority of e-commerce interactions for some time to come, increasing numbers of on-line customers want to speak with a person. If a natural language interface could successfully assume some of this customer-service burden, requirements for human staff would be reduced.

Potential over-the-web natural language application domains vary widely in complexity. At the lowest level, a customer may have questions regarding the operation of the web-site. If accessing an on-line auction, for example, ignorance as to how to post a bid might motivate a call to customer service. At an intermediate level, a customer may have questions regarding the transaction that was mediated by the web-site. Perhaps the product was not received, or the customer was incorrectly invoiced. A call to customer service might well be appropriate. At the highest level, the site may have been designed to generate interest in some external entity, such as a school or company, and such interest produces questions. While a comprehensive tour of the site may provide some answers, often site visitors have unanticipated questions, and a call is an obvious way to answer them.

Current attempts to develop over-the-net natural language technology is only in the early stages. While several 'natural language' applications have been deployed, none yet approach the performance needed for the type of applications mentioned above. So, the question obtains: how can one construct a robust natural language interface for real-world e-commerce? Obviously, answering this question would involve addressing many of the standard areas of natural language research.

The first, robustness, goes almost without saying. However, frailty due to shallowness of understanding is still common. Despite decades of NL research, robustness remains an elusive goal. Second, people routinely switch contexts with little or no warning,

when another question "pops into their heads." For example, in one sentence a customer might ask about their bank balance, then (perhaps in response to the unexpected low balance!) ask about loans. Such context switching must be handled accurately and naturally. Third, multi-level understanding of customer goals is critical, since people often ask questions that are related to unstated goals. For example, the question "What is the ratio of men to women at your College?" may reflect a broader desire to understand the overall social life. Note that "goal identification" is the requirement, not necessarily "goal inference". Inference is only one technology for addressing identification (another is for the system to just ask the customer, i.e. "There are 5 women for every 4 men on campus, but men are more likely to commute. Are you interested in other aspects of social life on campus?") Finally, recognition of non-understanding allows an interface to degrade smoothly. Few things are as annoying as a system (human or otherwise) that thinks it knows what you want despite your protestations. A system that recognizes that it does not understand is important for two reasons. One, it allows the system to apologize and try another strategy (i.e., connect the customer to a human being), and two, recognition of incompetence is a prerequisite to overall system improvement, whether by machine learning techniques or human intervention.

Overall, these requirements can only be met in the empirical world of real on-line systems: the imperative of real-world utility suggests that "people use it" is the only ultimate criterion for success. To meet this criteria, such a system would employ a single uniform architecture to span several levels of domain complexity, and would know how to detect when its boundaries of competence were reached. It would be limited by vocabulary and by domain, but would not require any special knowledge of its customers. It is currently an open question as to whether such a natural language capability can be built using some combination of existing technology, or whether new approaches must be developed.