Wlodek Zadrozny, M. Budzikowska, J. Chai,
N. Kambhatla, S. Levesque, and N. Nicolov

Natural Language Dialogue for Personalized Interaction

Technologies that successfully recognize and react to spoken or typed words are key to true personalization. Front- and back-end systems must respond in accord, and one solution may be found somewhere in the middle(ware).

The pragmatic goal of natural language (NL) and multimodal interfaces (speech recognition, keyboard entry, pointing, among others) is to enable ease-of-use for users/customers in performing more sophisticated human-computer interactions (HCI). NL research attempts to define extensive discourse models that in turn provide improved models of context-enabling HCI and personalization. Customers have the initiative to

JEAN-FRANCOIS PODEVIN

express their interests, wishes, or queries directly and naturally, by speaking, typing, and pointing. The computer system provides intelligent answers or asks relevant questions because it has a model of the domain and a user model. The business goal of such computerized systems is to create the marketplace of one. In essence, improved discourse models can enable better one-to-one context for each individual.

Even though we build NL systems, we realize this goal cannot be fully achieved due to limitations of science, technology, business knowledge, and programming environments. These environments include:

- Limitations of NL understanding;
- Managing the complexities of interaction (for example, when using NL on devices with differing bandwidth);
- Lack of precise user models (for example, knowing how demographics and personal characteristics of a person should be reflected in the type of language and dialogue the system is using with the user), and
- Lack of middleware and toolkits.

We would add another problem to this list: *Our repositories of knowledge are not designed for NL interaction.*

We view the use of NL as a compelling enabling technology for personalization, since each user can interact with the system in exactly his or her own words, rather than use one of a small number of preset ways to interact. Of course, we assume the system's behavior will be intelligent. Such behavior will require personalization because personal contextual data is a condition for smooth interaction. For instance, gender information is a useful bias and context cue when speaking about buying clothes.

The following scenarios illustrate current technology, which we contrast with a few scenarios we believe are beyond state of the art and require solving some fundamental problems.

Vanilla Scenarios

A bank customer at home and online is interested in buying a car. In the NL Search window he types the text "car loan." The system responds with a choice between "new car loans," "used car loans," and "existing automobile loans." The user points to "new car loans" and types "can I apply online?" The application form appears, a voice message is played explaining the application process.

Other scenarios for personalized service within reach of state-of-the-art technology include complex searches on specific Web sites. For example, "black pants without cuffs" would work very well with personalization. The system knows your size, sex, and preferences, and is capable of understanding simple negative modification. Systems can also be built for extracting relevant positive and negative events in the stock market. The system knows your interests, your portfolio, and has limited understanding of causal chains (profit warnings, interest rates influencing market, and so on). Personalized text categorization and translating and summarizing Web pages are two other scenarios. The system knows about your previous interactions, and only shows you new information in the language of your choice. You guess the meaning of mistranslated passages or occasionally use a dictionary.

We should keep in mind that NL understanding is a spectrum of technologies, starting from keyword searches to fully contextualized understanding of intentions. We are still on the left-hand side of the scale. Thus, we can add synonyms to keyword searches, or search text based on templates and answer queries like "How tall is Mount St. Helens?" However, no intelligent database can properly respond to these statements:

"I'm replacing Bill on this sales call on widgets this afternoon. What do I have to know?"

or

"I just inherited 50K. What do I do with it?"

Here are some other scenarios we contend are beyond current technology.

Investment advice. A bank customer looking at her stock portfolio, interacting with a research system capable of replying to queries such as "Which technology companies with high price-to-sales ratio are likely to merge in the next few months?" Even interpreting this natural query is difficult because it requires improving the state of the art in NL to cover semantics of modifiers such as "high," "likely," and "next few" with a prediction mechanism that would interpret "likely." Typically, projects that involve creating more than one advancement in a state of the art fail.

General investment advice is too difficult because the number of conceptual entities is large, and includes types of financial instruments as well as peoples goals, expectations, psychological profiles, and so on. Moreover, the relationships between the entities are complex. Finally, there are legal risks. How would one certify such a system and under what circum-

stances can its license be taken away?

Other beyond-the-state-of-the-art systems include machines for automated real-time transcription of meetings that would work with 97% accuracy, or for automated translation of phone conversations due to the state of speech systems. It is likely that further substantial progress in speech technology will be achieved when speech systems take semantics into account. In our opinion, this will require some fundamental breakthroughs. Similarly, it is not likely that you will see an intelligent secretary embodied in a software package or a Web site providing personalized intelligent travel planning. However, both domains seem susceptible to substantial gradual improvements.

How Do Dialogue Systems Work?

Current dialogue systems work in steps:

- Get text (if speech is the input medium, decode the speech into words);
- Extract relevant chunks of information from the resulting text;
- Using a stored template, check if all parameters have been given;
- If not, ask for missing parameters. Send a message to perform an action to a back-end engine;
- Provide some help, if necessary; and
- Present a back-end message to the user.

The first three steps are about language understanding (and speech recognition), the rest are about dialogue management. The final step also touches on NL and multimodal generation.

If we look at the possible scenarios mentioned here, we see the domain of interaction is limited, the back-end is a combination of a relational database and transaction system, and the dialogue patterns are pretty much fixed. On the other hand, the impossible scenarios require deeper understanding of the textual content and of users intentions. It would also be difficult to classify the patterns of interaction into a small number of dialogue templates.

Given the primitive state of the art, we should ask: Why does dialogue work? It enhances the robustness of communication: a partial match is enough to make progress and leads to successful interaction through negotiation. It also increases the efficiency of communication: lacking pieces of knowledge are deduced from encoded knowledge (back-end) or personalized data (front-end).

Engineering a dialogue system with its discourse model consists of choosing an appropriate domain, and the relevant set of parameters for all the steps. The choice of domain depends on business factors, technology factors and the relationships between the two. A few of the many relevant factors include the channel(s) of interaction (phone vs. Web), the importance of the number of concepts and relations in the domain of discourse, the availability of dictionaries, and the capability to recognize users requests and connect them with an appropriate execution mechanism (for example, what should count as a "matching shirt" or "inexpensive laptop" is a combination of user expectations and business decisions. It might be easier to interpret the last expression about a laptop than to deal with millions of ways in which outfits match or not).

Personalization makes this task both more difficult and easier. The increase in difficulty is obvious: We have to account for more factors. But these factors also make dialogue management more manageable. For instance, users' requests can be better understood based on preferences and profiles; global and local discourse history make inferencing more reliable; given explanations and acknowledgments from a system can be tailored based on personalized profile context data fitting an individual's preferences, thereby reducing misunderstanding.

For example, consider a product advisor, or more specifically, someone who advises buyers of PCs and accessories. If I am a novice user—but the system knows my demographic profile and has access to data linking demography with products—my request for "a computer under $2000" can be answered immediately with a picture of a specific machine, a list of other possible choices, and a justification: "This is our best-selling model. Users like it because … By the way, you might consider these upgrades." Similarly, if I already have a laptop, and I'm looking for additional memory, the dialogue simply consists of choosing the amount.

It is important to note that NL understanding and personalization does not necessarily imply customizing the dialogue to a specific user. It can mean customizing to a style or modality of interaction, to a group of users or to a channel of interaction (Web, phone, among others). Mixed initiative dialogue systems can be viewed as a form of personalization, where the dialogue strategy of the system is adapting in response to the perceived needs and style of users. This approach provides a middle ground between perfect personalization of dialogues, which is difficult to implement and no personalization, which is hard to endure.

What Holds Us Back and How Can We Make Progress?

As we mentioned in our opening remarks, the impediments to progress lie on several planes. They include language issues, in particular, semantics. Except for very restricted domains, we do not know how to com-

pute the meaning of a sentence based on meanings of its words and its context. On the other hand, it is possible to extract useful data in more restricted domains, and approximate the meaning of sentences that way.

Another issue where more progress would help is the lack of precise user models. Let us assume we can have any piece of information about a person. How could we use this knowledge to make this person's interaction with a dialogue system most effective and pleasant?

We are going to avoid discussing these two problems here. Instead, we address the issue of managing the business complexities of dialogue systems (for example, using NL dialogue on devices with differing bandwidth) by describing a piece of middleware called *Dialogue Moves Markup Language* (DMML).

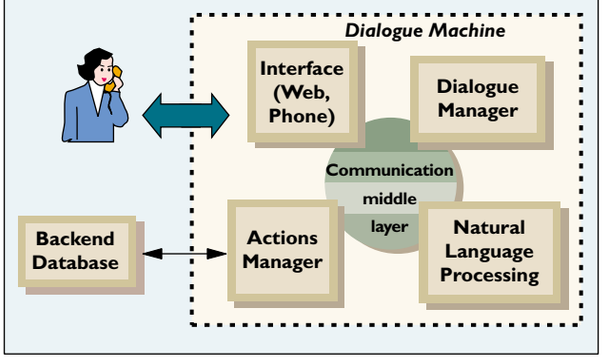
DMML Middleware and Universal Interaction

One of the key aspects of engineering is to design the system layer that makes the front- and back-end systems interwork—the middleware. By the front-end system, we mean those systems that interact with the user, for example, a window through which the user and the system negotiate delivery of groceries. The back-end typically consists of a database of documents and/or a transaction system, for example, a database of documents and a system for accepting payments for these documents. As we can see in Figure 1, the middle layer typically contains a dialogue manager, an action manager, a language-understanding module and a communication layer for conveying messages between them.

Now, consider a bank that must provide access to the same transaction system through different means: phone, Web, PDA, Web phone, the teller, the customer service representative on the phone, and perhaps a few others. Obviously, users want personalized consistent service through whichever devices they choose or through a human. If all access paths are engineered by different teams the chance for satisfying the customers is poor. Hence, the idea of *Universal Interaction.*

Universal Interaction uses the same mechanism for different communication devices (such as phones, PDAs, and desktop computers). This means conveying the same content through different channels by suitably modifying the way it is represented. Given the differences in bandwidth among these devices, it is not at all obvious that Universal Interaction is feasible. On the other hand it is obviously desirable, for it would save us the effort of engineering for each communication channel and each device separately, and ensure the consistency of style. Universal Interaction and dialogue management are connected because whatever can be done in one turn on a desktop computer, might require a few turns of phone dialogue or interactions with a PDA. When the bandwidth of a device is low, we have to compensate by increasing the number of turns. Similarly, if there is a great deal of information to be conveyed to or by the user. Dialogue management issues include such elements as what to ask the user and in what order, what information to convey by text, what by voice, and what by picture.

Figure 1. The architecture of conversational systems.



Once we remember the NL capability to choose from a large number of possibilities, we realize that there is no effective Universal Interaction without NL dialogue.

Universal Interaction architecture is tailor-made for personalization, the actual interface for each user can be specifically constructed for him or her based on geography-specific, user-specific, and style-specific transformations.

How do we transform the content to fit into different representations? A potentially good idea is to use a XML/XSL-based architecture for the separation of form, content, style, and interactions. To make the idea more specific, imagine how one can represent a dialogue move in a stock trading scenario.

DMML—inspired by the theory of speech acts and XML—is an attempt to capture the intent of communicative agents in the context of NL dialogue management. The idea is to codify dialogue moves[1] such as *greetings, warnings, reminders, thanks, notifications, clarifications,* or *confirmations* in a set of tags connected at runtime with NL understanding modules, which allows us to describe participants' behaviors in terms of dialogue moves, without worrying about how they are expressed in language. For instance, "any positive identification must be followed by a confirmation." The

[1] A dialogue move is a unit of interaction in dialog. For example, in spoken dialogue it consists of one or more phrases (for example, "Hello, what is my checking balance").

Figure 2. A dialogue tagged with DialogMoves Markup Language.

Request Dialogue Move	Response Dialogue Move
USER: Hi! I want to buy one hundred shares of IBM	SYSTEM: Hello, Steve! You said you wanted to buy 100 shares of IBM. Do you want to use your checking account? At what price do you want to buy these shares?
<DIALOG_MOVE AGENT_FROM="USER" AGENT_TO="CDM_STOCKS" MODALITY="TEXT" CHANNEL="WEB-PHONE"> <ASSERT> <GREET>Hi</GREET> </ASSERT> <REQUEST REQUEST_TYPE="ACTION"> **<BUY_STOCKS> <SPECIFIED>** **<COMPANY>IBM</COMPANY>** **<QUANTITY>100</QUANTITY>** **</SPECIFIED> </BUY_STOCKS>** </REQUEST> </DIALOG_MOVE>	<DIALOG_MOVE AGENT_FROM="CDM_STOCKS" AGENT_TO="USER" MODALITY="TEXT/GRAPHICS" CHANNEL="WEB-PHONE"> <ASSERT> <GREET>Hello, Steve!</GREET> </ASSERT> <RESPONSE> RESPONSE_TYPE="REQUEST"> <REQUESTED REQUEST_TYPE="ACTION"> **<BUY_STOCKS> <SPECIFIED>** **<COMPANY>IBM</COMPANY>** **<QUANTITY>100</QUANTITY> </SPECIFIED> </BUY_STOCKS>** </REQUESTED> <REQUEST REQUEST_TYPE="CONFIRMATION"> **</BUY_STOCKS> <SPECIFIED BY="USER">** **<COMPANY>IBM</COMPANY>** **<QUANTITY>100</QUANTITY> </SPECIFIED> <INFERRED>** **<ACCOUNT>checking</ACCOUNT>** **<INFERRED> <UNKNOWN> <PRICE"/> </UNKNOWN> </BUY_STOCKS>** </REQUEST> </REPONSE> </DIALOG_MOVE>

tags can also encode other parameters of the dialogue, such as the type of channel and personal characteristics. Thus, the dialogue can reflect the channel characteristics, which connects DMML and Universal Interaction.

Figure 2 shows the modularity of the approach: normal black font represents dialogue moves (DMML), bold font represents domain-specific constructs, like parameters of stock-trading transactions. DMML illustrates the concept of a communication middleware in dialogue systems and is very well suited for personalization.

Designing for Interaction

The final point in this article is about knowledge engineering, with NL dialogue in mind. The key idea is it might be worth designing our repositories of textual knowledge and some relational databases for NL interaction. Note that textual documents are already designed for printing—that is, for a specific type of interaction with humans that is mediated by a printer—and more recently, for the Internet. Similarly, Web pages are designed for interaction mediated by a browser.

This is not sufficient since neither provides the level of granularity we often desire, that is, access to specific information in a limited context. Furthermore, both are passive. Thus, it is a logical next step to make the knowledge active and adaptable to the user. The problem is not simple because it involves both rethinking the format in which data is stored, and creating dialogue interfaces that incorporate some knowledge of the domain. However, for some types of data, such as instruction books or catalogs, it might be solvable by extending current technologies. For example, catalogs are a good starting point because they are typically structured and amenable to classic knowledge engineering techniques. One such system is the previously mentioned product advisor.

For other perspectives on this topic, readers may be interested in the following resources: *Natural Language Understanding,* by James Allen, Addison-Wesley, is a classic introduction to language processing. A recent update is given in *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition* by Daniel Jurafsky and James H. Martin, Prentice Hall. The proper starting point for exploring Internet resources is the Web page of the Association for Computational Linguistics—see www.aclweb.org; information on XML and related topics can be found at www.xml.org, and, finally, pointers to our own work appear at www.research.ibm.com/compsci/linguistics/index.html. C

Wlodek Zadrozny (wlodz@us.ibm.com) is manager, conversational machines; **M. Budzikowska** (SM1@us.ibm.com) is a research staff member; **J. Chai** (jchai@us.ibm.com) is a research staff member; **N. Kambhatla** (nanda@us.ibm.com) is a research staff member; **S. Levesque** (slevesqu@ca.ibm.com) is IT/Architect, IBM Canada; and **N. Nicolov** (nicolas@us.ibm.com) is a visiting scientist. All, except Levesque, are located at IBM T.J. Watson Research Center in Hawthorne, NY.