1

Conversational Agents

James Lester

Karl Branting

Bradford Mott

CONTENTS

Abstract Conversational agents integrate computational linguistics techniques with the communication channel of the Web to interpret and respond to statements made by users in ordinary natural language. Web-based conversational agents deliver high-volumes of interactive text-based dialogs. Recent years have seen significant activity in enterprise-class conversational agents. This chapter describes the principal applications of conversational agents in the enterprise and the technical challenges posed by their design and large-scale deployments. These technical challenges fall into two categories: accurate and efficient natural-language processing; and the scalability, performance, reliability, integration, and maintenance requirements posed by enterprise deployments.

1.1 Introduction

The Internet has introduced sweeping changes in every facet of contemporary life. Business is conducted fundamentally differently than in the pre-Web era. We educate our students in new ways, and we are seeing paradigm shifts in government, healthcare, and entertainment. At the heart of

X-XXXX-XXXX-X/XX/$X.XX+$X.XX

these changes are new technologies for communication, and one of the most promising communication technologies is the conversational agent, which marries agent capabilities with computational linguistics.

Conversational agents exploit natural language technologies to engage users in text-based information-seeking and task-oriented dialogs for a broad range of applications. Deployed on retail websites, they respond to customers' inquiries about products and services. Conversational agents associated with financial services' websites answer questions about account balances and provide portfolio information. Pedagogical conversational agents assist students by providing problem-solving advice as they learn. Conversational agents for entertainment are deployed in games to engage players in situated dialogs about the game-world events. In coming years, conversational agents will support a broad range of applications in business enterprises, education, government, healthcare, and entertainment.

Recent growth in conversational agents has been propelled by the convergence of two enabling technologies. First, the Web emerged as a universal communications channel. Web-based conversational agents are scalable enterprise systems that leverage the Internet to simultaneously deliver dialog services to large populations of users. Second, computational linguistics, the field of artificial intelligence that focuses on natural language software, has seen major improvements. Dramatic advances in parsing technologies, for example, have significantly increased natural language understanding capabilities.

Conversational agents are beginning to play a particularly prominent role in one specific family of applications: enterprise software. In recent years, the demand for cost-effective solutions to the customer service problem has increased dramatically. Deploying automated solutions can significantly reduce the high proportion of customer service budgets devoted to training and labor costs. By exploiting the enabling technologies of the Web and computational linguistics noted above, conversational agents offer companies the ability to provide customer service much more economically than with traditional models. In *customer-facing* deployments, conversational agents interact directly with customers to help them obtain answers to their questions. In *internal-facing* deployments, they converse with customer service representatives to train them and help them assist customers.

In this chapter we will discuss Web-based conversational agents, focusing on their role in the enterprise. We first describe the principal applications of conversational agents in the business environment. We then turn to the technical challenges posed by their development and large-scale deployments. Finally, we review the foundational natural language technologies of interpretation, dialog management, and response execution, as well as an enterprise architecture that addresses the requirements of conversational scalability, performance, reliability, "authoring," and maintenance in the enterprise.

1.2 Applications

Effective communication is paramount for a broad range of tasks in the enterprise. An enterprise must communicate clearly with its suppliers and partners, and engaging clients in an ongoing dialog—not merely metaphorically but also literally—is essential for maintaining an ongoing relationship. Communication characterized by information-seeking and task-oriented dialogs is central to five major families of business applications:

- *Customer service*: Responding to customers' general questions about products and services, e.g., answering questions about applying for an automobile loan or home mortgage.
- *Help desk*: Responding to internal employee questions, e.g., responding to HR questions.
- *Website navigation*: Guiding customers to relevant portions of complex websites. A "Website concierge" is invaluable in helping people determine where information or services reside on

a company's website.

- *Guided selling*: Providing answers and guidance in the sales process, particularly for complex products being sold to novice customers.
- *Technical support*: Responding to technical problems, such as diagnosing a problem with a device.

In commerce, clear communication is critical for acquiring, serving, and retaining customers. Companies must educate their potential customers about their products and services. They must also increase customer satisfaction and, therefore, customer retention, by developing a clear understanding of their customers' needs. Customers seek answers to their inquiries that are correct and timely. They are frustrated by fruitless searches through websites, long waits in call queues to speak with customer service representatives, and delays of several days for email responses.

Improving customer service and support is essential to many companies because the cost of failure is high: loss of customers and loss of revenue. The costs of providing service and support are high and the quality is low, even as customer expectations are greater than ever. Achieving consistent and accurate customer responses is challenging and response times are often too long. Effectiveness is, in many cases, further reduced as companies transition increasing levels of activity to Web-based self-service applications, which belong to the customer relationship management software sector.

Over the past decade, customer relationship management (CRM) has emerged as a major class of enterprise software. CRM consists of three major types of applications: sales force automation, marketing, and customer service and support. Sales force automation focuses on solutions for lead tracking, account and contact management, and partner relationship management. Marketing automation addresses campaign management and email marketing needs, as well as customer segmentation and analytics. Customer service applications provide solutions for call center systems, knowledge management, and e-service applications for Web collaboration, email automation, and live chat. It is to this third category of customer service systems that conversational agent technologies belong.

Companies struggle with the challenges of increasing the availability and quality of customer service while controlling their costs. Hiring trained personnel for call centers, live chat, and email response centers is expensive. The problem is exacerbated by the fact that service quality must be delivered at a level where customers are comfortable with the accuracy and responsiveness.

Companies typically employ multiple channels through which customers may contact them. These include expensive support channels such as phone and interactive voice response systems. Increasingly, they also include Web-based approaches because companies have tried to address increase demands for service while controlling the high cost of human-assisted support. E-service channels include live chat and email, as well as search and automated email response.

The tradeoff between cost and effectiveness in customer support presents companies with a dilemma. Although quality human-assisted support is the most effective, it is also the most expensive. Companies typically suffer from high turnover rates which, together with the costs of training, further diminish the appeal of human-assisted support. Moreover, high turnover rates increase the likelihood that customers will interact with inexperienced customer service representatives, who provide incorrect and inconsistent responses to questions.

Conversational agents offer a solution to the cost versus effectiveness tradeoff for customer service and support. By engaging in automated dialog to assist customers with their problems, conversational agents effectively address sales and support inquiries at a much lower cost than human-assisted support. Of course, conversational agents cannot enter into conversations about all subjects—because of the limitations of natural language technologies they can only operate in circumscribed domains—but they can nevertheless provide a cost-effective solution in applications where question-answering requirements are bounded. Fortunately, the applications noted above (customer service, help desk, website navigation, guided selling, and technical support) are often

characterized by subject matter areas restricted to specific products or services. Consequently, companies can meet their business objectives by deploying conversational agents that carry on dialogs about a particular set of products or services.

1.3 Technical Challenges

Conversational agents must satisfy two sets of requirements. First, they must provide sufficient language processing capabilities that they can engage in productive conversations with users. They must be able to understand users' questions and statements, employ effective dialog management techniques, and accurately respond at each "conversational turn." Second, they must operate effectively in the enterprise. They must be scalable and reliable, and they must integrate cleanly into existing business processes and enterprise infrastructure. We discuss each of these requirements in turn.

1.3.1 Natural Language Requirements

Accurate and efficient natural language processing is essential for an effective conversational agent. To respond appropriately to a user's utterance,[1] a conversational agent must (1) interpret the utterance, (2) determine the actions that should be taken in response to the utterance, and (3) perform the actions, which may include replying with text, presenting Web pages or other information, and performing system actions such as writing information to a database.

For example, if the user's utterance were:

(1) I would like to buy it now

the agent must first determine the literal meaning of the utterance: the user wants to purchase something, probably something mentioned earlier in the conversation. In addition, the agent must infer the goals that the user sought to accomplish by making an utterance with that meaning. Although the user's utterance is in the form of an assertion, it was probably intended to express a request to complete a purchase.

Once the agent has interpreted the statement, it must determine how to act. The appropriate actions depend on the current goal of the agent (e.g., selling products or handling complaints), the dialog history (the previous statements made by the agent and user), and information in databases accessible to the agent, such as data about particular customers or products. For example, if the agent has the goal of selling products, the previous discussion identified a particular consumer item for sale at the agent's website, and the product catalog shows the item to be in stock, the appropriate action might be to present an order form and ask the user to complete it. If instead the previous discussion hadn't clearly identified an item, the appropriate action might be to elicit a description of a specific item from the user. Similarly, if the item were unavailable, the appropriate action might be to offer the user a different choice.

Finally, the agent must respond with appropriate actions. The appropriate actions might include making a statement, presenting information in other modalities, such as product photographs, and taking other actions, such as logging information to a database. For example, if the appropriate action were to present an order form to the user and ask the user to complete it, the agent would need to retrieve or create a statement such as "Great! Please fill out the form below to complete your purchase," create or retrieve a suitable Web page, display the text and Web page on the user's browser, and log the information. Figure 1.1 depicts the data flow in a conversational agent system.

The three primary components in the processing of each utterance are shown in Figure 1.2. The first component in this architecture, the Interpreter, performs four types of analysis of the user's statement: syntactic, discourse, semantic, and pragmatic. Syntactic analysis consists of determining

[1] An *utterance* is a question, imperative, or statement issued by a user.

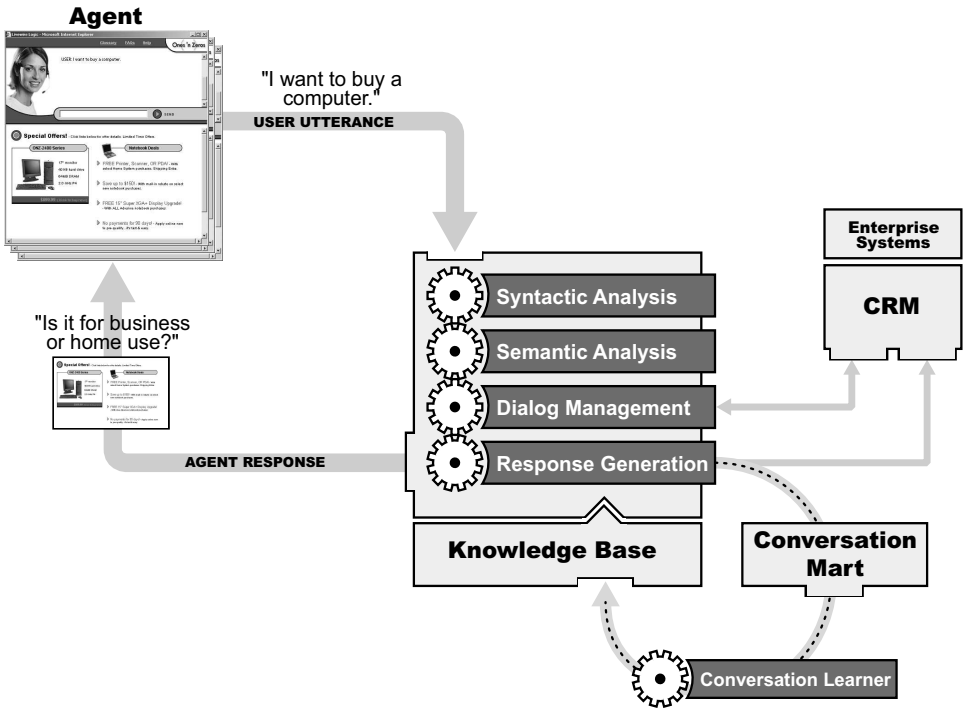

Figure 1.1: Data flow in a conversational agent

the grammatical relationships among the words in the user's statement. For example, in the sentence

(2) I would like a fast computer

syntactic analysis would produce a parse of the sentence showing that "would like" is the main verb, "I" is the subject, and "a fast computer" is the object. Although many conversational agents (including the earliest) rely on pattern matching without any syntactic analysis [Weizenbaum, 1966], this approach cannot scale. As the number of statements that the agent must distinguish among increases, the number of patterns required to distinguish among the statements grows rapidly in number and complexity.[2] Discourse analysis consists of determining the relationships among multiple sentences. An important component of discourse analysis is *reference resolution*, the task of determining the entity denoted by a referring expression, such as the "it" in, "I would like to buy it now." A related problem is interpretation of *ellipsis*, that is, material omitted from a statement but implicit in the conversational context. For example, "Wireless" means, "I would like the wireless network" in response to the question, "Are you interested in a standard or wireless network?", but the same utterance means "I want the wireless PDA" in response to the question, "What kind of PDA would you like?"

Semantic analysis consists of determining the meaning of the sentence. Typically, this consists of representing the statement in a canonical formalism that maps statements with similar meaning to a single representation and that facilitates the inferences that can be drawn from the representation. Approaches to semantic analysis include the following:

- Replace each noun and verb in a parse with a word sense that corresponds to a set of synonymous words, such as WordNet synsets [Fellbaum, 1999].

[2] In fact, conversational agents must address two forms of scalability: *domain scalability*, as discussed here, and *computational scalability*, which refers to the ability to handle large volumes of conversations and is discussed in Section 1.3.2.

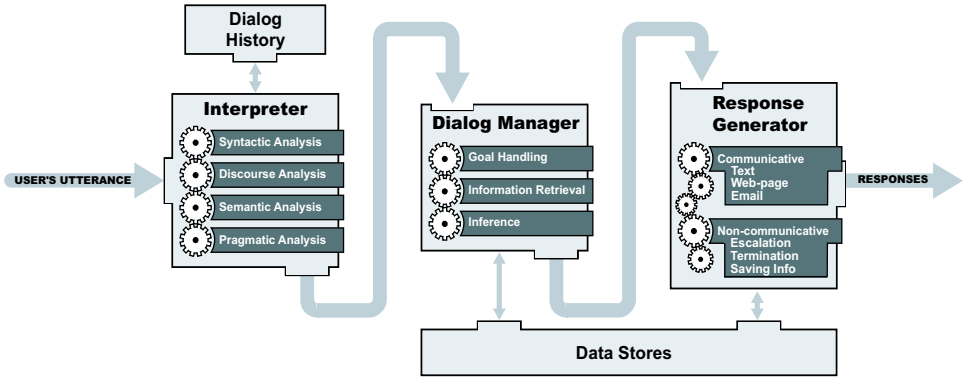

Figure 1.2: The primary natural language components of a conversational agent

- Represent the statement as a case frame [Fillmore, 1968], dependency tree [Harabagiu et al., 2000], or logical representation, such as first order predicate calculus.

Finally, the Interpreter must perform pragmatic analysis, determining the pragmatic effect of the utterance, that is, the speech (or communication) act [Searle, 1979] that the utterance performs. For example, "Can you show me the digital cameras on sale?" is in the form of a question, but its pragmatic effect is a request to display cameras on sale. "I would like to buy it now" is in the form of a declaration, but its pragmatic effect is also a request. Similarly, the pragmatic effect of "I don't have enough money," is a refusal in response to the question "Would you like to proceed to checkout?" but a request in response to "Is there anything you need from me?"

The interpretation of the user's statement is passed to a Dialog Manager, which is responsible for determining the actions to take in response to the statement. The appropriate actions depend on the interpretation of the user's statement and the dialog state of the agent, which represents the agent's current conversation goal. In the simplest conversational agents, there may be only a single dialog state, corresponding to the goal of answering the next question. In more complex agents, a user utterance may cause a transition from one dialog state to another. The new dialog state is, in general, a function of the current state, the user's statement, and information available about the user and the products and services under discussion. Determining a new dialog state may therefore require database queries and inference. For example, if the user's statement is, "What patch do I need for my operating system?" and the version of the user's operating system is stored in the user's profile, the next dialog state may reflect the goal of informing the user of the name of the patch. If the version of the operating system is unknown, the transition may be to a dialog state reflecting the goal of eliciting the operating system version.

The Dialog Manager is responsible for detecting and responding to changes in topic. For example, if a user's question can't be answered without additional information from the user, the dialog state must be revised to reflect the goal of eliciting the additional information. Similarly, if a user fails to understand a question and asks for a clarification, the dialog state must be changed to a state corresponding to the goal of providing the clarification. When the goal of obtaining additional information or clarification is completed, the Dialog Manager must gracefully return to the dialog state at which the interruption occurred.

The final component is the Response Generator. Responses fall into two categories: communications to the user, such as text, Web pages, email, or other communication modalities; and non-communication responses, such as updating user profiles (e.g., if the user's statement is a declaration of information that should be remembered, such as "My OS is Win-XP"), escalating from a conversational agent to a customer service representative (e.g., if the agent is unable to handle the conversation), and terminating the dialog when it is completed. The responses made by the agent

depend on the dialog state resulting from the user's statement (which represents the agent's current goals) and the information available to the agent through its dialog history, inference, or other data sources. For example, if the current dialog state corresponds to the goal of informing the user of the cost of an item for sale at a website and the price depends on whether the user is a repeat customer, the response might depend on the information in the dialog history concerning the status of the user and the result of queries to product catalogs concerning alternative prices.

Responses typically include references to existing content that has been created throughout the enterprise. Repurposing content is particularly important when the products and services that a response addresses change continually. Centralized authoring, validation, and maintenance of responses facilitate consistency and drastically reduce maintenance costs.

Enterprise applications of conversational agents impose several constraints not generally present in other forms of conversational agents. First, high accuracy and graceful degradation of performance are very important for customer satisfaction. Misunderstandings in which the agent responds as though the user had stated something other than what the user intended to say (false positives), can be very frustrating and difficult for the agent to recover from, particularly in dialog settings. Once the agent has started down the wrong conversational path, sophisticated dialog management techniques are necessary to detect and recover from the error. Uncertainty by the agent about the meaning of a statement (false negatives) can also be frustrating to the user if the agent repeatedly asks users to restate their questions. It is often preferable for the agent to present a set of candidate interpretations and ask the users to choose the interpretation they intended.

Second, it is essential that authoring be easy enough to be performed by non-technical personnel. Knowledge bases are typically authored by subject matter experts in marketing, sales, and customer care departments who have little or no technical training. They cannot be expected to create scripts or programs; they certainly cannot be expected to create or modify grammars consisting of thousands of productions (grammar rules). Authoring tools must therefore be usable by personnel who are non-technical but who can nonetheless provide examples of questions and answers. State-of-the-art authoring suites exploit machine-learning and other corpus-based and example-based techniques. They induce linguistic knowledge from examples, so authors are typically not even aware of the existence of the grammar. Hiding the details of linguistic knowledge and processing from authors is essential for conversational agents delivered in the enterprise.

1.3.2 Enterprise Delivery Requirements

In addition to the natural language capabilities outlined above, conversational agents can be introduced into the enterprise only if they meet the needs of a large organization. To do so, they must provide a "conversational QoS" that enables agents to enter into dialogs with thousands of customers on a large scale. They must be scalable, provide high throughput, and guarantee reliability. They must also offer levels of security commensurate with the conversational subject matter, integrate well with the existing enterprise infrastructure, provide a suite of content creation and maintenance tools that enable the enterprise to efficiently author and maintain the domain knowledge, and support a broad range of analytics with third-party business intelligence and reporting tools.

Scalability. Scalability is key to conversational agents. Because the typical enterprise that deploys a conversational agent does so to cope with extraordinarily high volumes of inbound contacts, conversational agents must scale well. To offer a viable solution to the contemporary enterprise, conversational agents must support on the order of tens of thousands of conversations each day. Careful capacity planning must be undertaken prior to deployment. Conversational agents must be architected to handle ongoing expanded roll-outs to address increased user capacity. Moreover, because volumes can increase to very high levels during crisis periods, conversational agents must support rapid expansions of conversations on short notice. Because volume is difficult to predict, conversational agents must be able to dynamically increase all resources needed to handle unexpected additional dialog demand.

Performance. Conversational agents must satisfy rigorous performance requirements, which are measured in two ways. First, agents must supply a conversational throughput that addresses the volumes seen in practice. Although the loads vary from one application to another, agents must be able to handle on the order of hundreds of utterances per minute, with peak rates in the thousands. Second, agents must also provide guarantees on the number of simultaneous conversations, as well as the number of simultaneous utterances, that they can support. In peak times, a large enterprise's conversational agent can receive a very large volume of questions from thousands of concurrent users, which must be processed as received in a timely manner to ensure adequate response times. As a rough guideline, agents must provide response times in a few milliseconds so that the total response time (including network latency) is within the range of one or two seconds.[3]

Reliability. For all serious enterprise deployments, conversational reliability and availability is critical. Conversational agents must be able to reliably address users' questions in the face of hardware and software failures. Failover mechanisms specific to conversational agents must be in place. For example, if a conversational agent server goes down, then ongoing and new conversations must be processed by remaining active servers, and conversational transcript logging must be continued uninterrupted. For some mission critical conversational applications, agents may need to be geographically distributed to ensure availability, and both conversational knowledge bases and transcript logs may need to be replicated.

Security. The security requirements of the enterprise as a whole, as well as those of the particular application for which a conversational agent is deployed, determine its security requirements. In general, conversational agents must provide at least the same level of security as the site on which it resides. However, because conversations can cover highly sensitive topics and reveal critical personal information, the security levels at which conversational agents must operate are sometimes higher than the environment they inhabit. Agents therefore must be able to conduct conversations over secure channels and support standard authentication and authorization mechanisms. Furthermore, conversational content creation tools (see below) must support secure editing and promotion of content.

Integration. Conversational agents must integrate cleanly with existing enterprise infrastructure. In the presentation layer, they must integrate with content management systems and personalization engines. Moreover, the agent's responses must be properly synchronized with other presentation elements, and if there is a visual manifestation of an agent in a deployment (e.g., as an avatar), all media must also be coordinated. In the application layer, they must easily integrate with all relevant business logic. Conversational agents must be able to access business rules that are used to implement escalation policies and other domain-specific business rules that affect dialog management strategies. For example, agents must be able to integrate with CRM systems to open trouble tickets and populate them with customer-specific information that provides details of complex technical support problems. In the data storage layer, conversational agents must be able to easily integrate with back-office data such as product catalogs, knowledge management systems, and databases housing information about customer profiles. Finally, conversational agents must provide comprehensive (and secure) administrative tools and services for day-to-day management of agent resources.

To facilitate analysis of the wealth of data provided by hundreds of thousands of conversations, agents must integrate well with third-party business intelligence and reporting systems. At runtime, this requirement means that transcripts must be logged efficiently to databases. At analysis time, it means that the data in "conversation marts" must be easily accessible for reporting on and for running exploratory analyses. Typically, the resulting information and its accompanying statistics provide valuable data that are used for two purposes: improving the behavior of the agent, and tracking users' interests and concerns.

[3] In well-engineered conversational agents, response times are nearly independent of the size of the subject matter covered by the agent.

1.4 Enabling Technologies

The key enabling technologies for Web-based conversational agents are empirical, corpus-based computational linguistics techniques, which permit development of agents by subject-matter experts who are not expert in computer technology, and techniques for robustly delivering conversations on a large scale.

1.4.1 Natural Language Processing Technologies

Natural language processing (NLP) is one of the oldest areas of Artificial Intelligence research, with significant research efforts dating back to the 1960s. However, progress in NLP research was relatively slow during its first decades because manual construction of NLP systems was time consuming, difficult, and error-prone. In the 1990s, however, three factors led to an acceleration of progress in NLP. The first was development of large corpora of tagged texts, such as the Brown Corpus, the Penn Treebank [LDC, 2003], and the British National Corpus [Bri, 2003]. The second factor was development of statistical, machine-learning, and other empirical techniques for extracting grammars, ontologies, and other information from tagged corpora. Competitions, such as MUC and TREC [Nat, 2003], in which alternative systems were compared head-to-head on common tasks, were a third driving force. The combination of these factors has led to rapid improvements in techniques for automating the construction of NLP systems.

The first stage in the interpretation of a user's statement, syntactic analysis, starts with tokenization of the user's statement, that is, division of the input in a series of distinct lexical entities. Tokenization can be surprisingly complex. One source of tokenization complexity is contraction ambiguity, which can require significant contextual information to resolve, e.g., "John's going to school" vs. "John's going to school makes him happy." Other sources of tokenization complexity include acronyms (e.g., "arm" can mean "adjustable rate mortgage" as well as a body part), technical expressions (e.g., "10BaseT" can be written with hyphens or spaces, as in "10 Base T"), multi-word phrases (e.g., "I like diet coke" vs. "when I diet coke is one thing I avoid"), and misspellings.

The greatest advances in automated construction of NLP components have been in syntactic analysis. There are two distinct steps in most implementations of syntactic analysis: part-of-speech (POS) tagging; and parsing. POS tagging consists of assigning to each token a part of speech indicating its grammatical function, such as singular noun or comparative adjective. There are a number of learning algorithms capable of learning highly accurate POS tagging rules from tagged corpora, including transformation-based and maximum entropy-based approaches [Brill, 1995; Ratnaparkhi, 1996].

Two distinct approaches to parsing are appropriate for conversational agents. Chunking, or robust parsing, consists of using finite-state methods to parse text into chunks, that is, constituent phrases with no post-head modifiers. There are very fast and accurate learning methods for chunk grammars [Cardie et al., 1999; Abney, 1995]. The disadvantage of chunking is that finite-state methods can't recognize structures with unlimited recursion, such as embedded clauses (e.g., "I thought that you said that I could tell you that . . . "). Context-free grammars can express unlimited recursion at the cost of significantly more complex and time-consuming parsing algorithms. A number of techniques have been developed for learning context-free grammars from tree banks [Sta, 2003]. The performance of the most accurate of these techniques, such as lexicalized probabilistic context-free grammars [Collins, 1997], can be quite high, but the parse time is often quite high as well. Web-based conversational agents may be required to handle a large number of user statements per second, so parsing time can become a significant factor in choosing between alternative approaches to parsing. Moreover, the majority of statements directed to conversational agents are short, without complex embedded structures.

The reference-resolution task of discourse analysis in general is the subject of active research [Ref, 2003], but a circumscribed collection of rules is sufficient to handle many of the most common cases. For example, recency is a good heuristic for the simplest cases of anaphora resolution, e.g., in the sentence (3), "one" is more likely to refer to "stereo" than to "computer."

(3) I want a computer and a stereo if one is on sale

Far fewer resources are currently available for semantic and pragmatic analysis than for syntactic analysis, but several ongoing projects provide useful materials. WordNet, a lexical database, has been used to provide lexical semantics for the words occurring in parsed sentences [Fellbaum, 1999]. In the simplest case, pairs of words can be treated as synonymous if they are members of a common WordNet synonym set.[4] FrameNet is a project that seeks to determine the conceptual structures, or frames, associated with words [Baker et al., 1998]. For example, the word "sell" is associated, in the context of commerce, with a seller, a buyer, a price, and a thing that is sold. The "sell" frame can be used to analyze the relationships among the entities in a sentence having "sell" as the main verb. FrameNet is based on a more generic case frame representation that organizes sentences around the main verb phrase, assigning other phrases to a small set of roles, such as agent, patient, and recipient [Filmore 67]. Most approaches to pragmatic analysis have relied on context to disambiguate among a small number of distinguishable communicative acts or have used ad hoc, manually constructed rules for communicative-act classification.

Figure 1.3 displays the steps in the interpretation of sentence (1) above. The first step is POS tagging, using the Penn Treebank POS tags. Next, the tagged text is parsed with a simple context-free grammar. The pronouns, "I" and "it," are replaced in the discourse analysis step, based on the rules that "I" refers to the user, in this case customer 0237, and that "it" refers to the most recently mentioned noun phrase. The resulting parse tree is converted to a case-frame representation in which "like" is the main verb, customer 0237 is the agent, and a case frame involving "buy" is the patient. In the pragmatic analysis step, the verb "like" with a case frame as its patient is interpreted as the illocutionary action "request" with the embedded case frame as the thing requested. The final result of the interpretation process is then passed to the Dialog Manager.

Three approaches have commonly been applied to dialog management. The first approach uses a finite-state automaton (FSA) as a dialog model. Transitions in the FSA correspond to individual statements, and states correspond to agent goals. The FSA approach is suitable for simple dialogs in which all possible sequences of dialog states can be anticipated. However, it is ill-suited for dialogs in which the order of statements is unpredictable, the user may supply several pieces of information in a single utterance, or mixed initiative dialogs, i.e., dialogs in which conversational control can move between interlocutors. Mixed initiative occurs when one person answers a question with another question, as when additional information is needed to answer a question. Consider the following dialog:

```
User:  Do I qualify for a loan?
Agent: What was your income last year?
User:  Does money I can get from my parents count as income?
Agent: Do you have to pay the money back?
User:  Yes.
Agent: Money you have to pay back is not income.
User:  Then my income was about $45000
Agent: Do you own anything that could be used as collateral?
```

The dialog starts with a question from the user about qualifying for a loan, but to answer the question the agent needs information about the user's income. The agent therefore changes the focus to the user's income. However, the user needs additional information about what qualifies as income

[4] For example, "tail" and "tag" belong to a WordNet synset that also includes "chase," "chase after," "trail," and "dog."

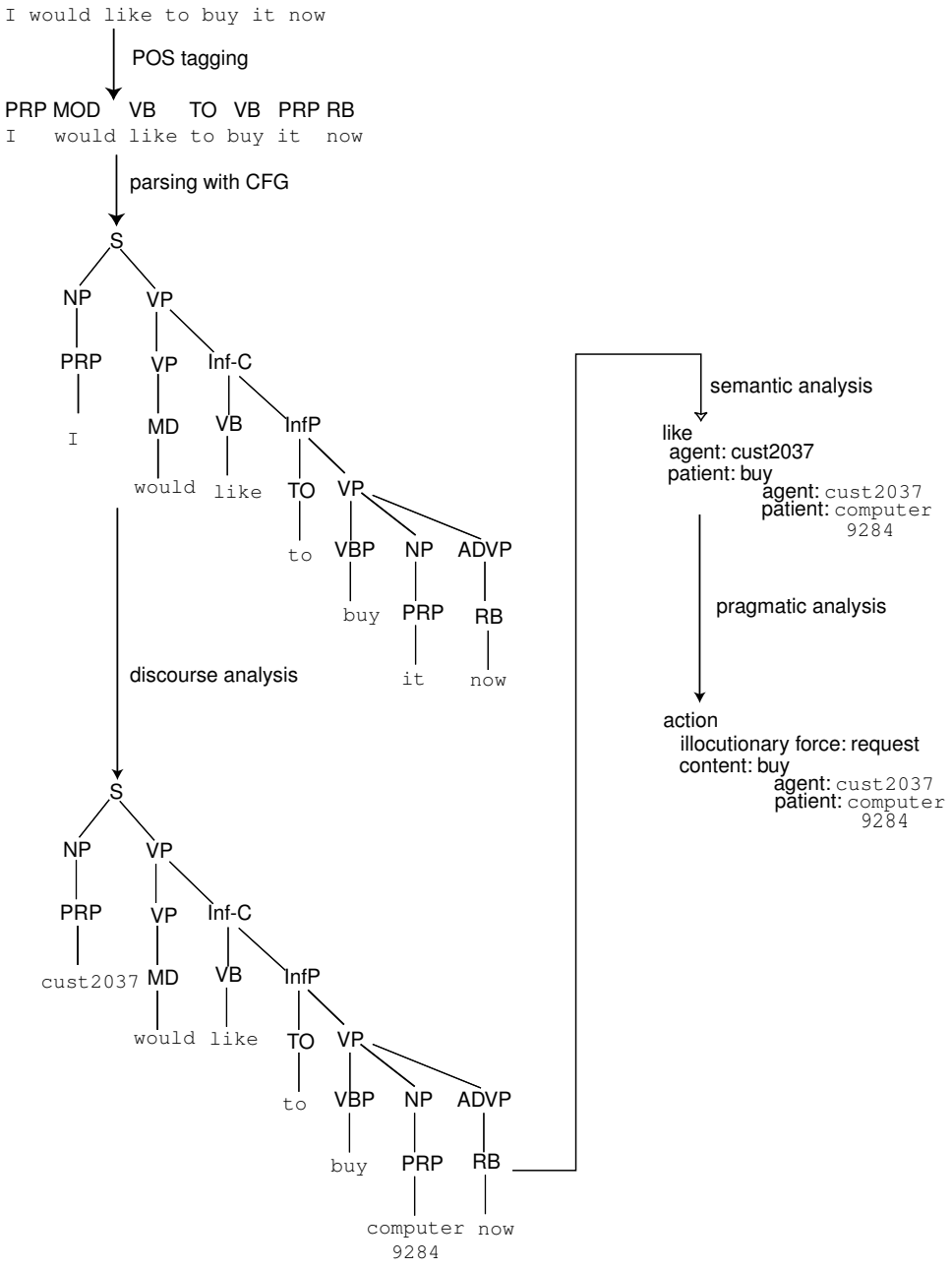

Figure 1.3: Steps in the processing of sentence (1)

to answer the agent's question, so the user takes the initiative again. Once again, the agent can only answer the question by asking an additional question about whether a transfer of money was income. After the user provides the information needed by the agent, the agent can answer the previous question by the user concerning income, allowing the user to answer the previous question about income. The agent then returns to the goal of eliciting the information needed to answer the original question.

A second approach to dialog management, suited for information elicitation systems, uses templates or frames with slots corresponding to the information to be elicited. This handles unpredictable statement order and compound statements more effectively than the FSA approach, but provides little support for mixed-initiative dialog.

The third approach uses a goal stack or an agenda mechanism to manage dialog goals. This approach can change topics by pushing a goal state corresponding to a new topic onto the stack, then popping the stack when the topic is concluded. The goal-stack approach is more complex to design than the FSA or template approaches, but is able to handle mixed-initiative dialogs.

Continuing the example of sentence (1) above, since the Dialog Manager has received a "request" communicative act from the Interpreter with content:

```
Buy
  Agent: cust0237
  Patient: computer9284
```

the Dialog Manager should change state, either by following a transition in an FSA corresponding to a request to buy or by pushing onto a goal stack a goal to complete a requested sale. If the patient of the buy request had been unspecified, the transition would have been to a dialog state corresponding to the goal to determine the thing that the user wishes to buy.

A change in dialog state by the Dialog Manager gives rise to call to the Response Generator to take one or more appropriate actions, including communications to the user and non-communication responses. Typically, only canned text is used, but sometimes template instantiation [Reiter, 1995] is used. In the current example, a dialog state corresponding to the goal completing a requested purchase of a computer might cause the Response Generator to instantiate a template with slots for the computer model and price. For example, the template

```
Great! <computer model> is on sale this week for just <price>!
```

might be instantiated as

```
Great! Power server 1000 is on sale this week for just $1,000.00!
```

Similarly, other communication modalities, such as Web pages and email messages, can be implemented as templates instantiated with context-specific data.

Over the course of a deployment, the accuracy of a well-engineered conversational agent improves. Both false positives and false negatives diminish over time as the agent learns from its mistakes. Learning begins before the go-live in "pre-training" sessions and continues after the agent is in high-volume use. Even after accuracy rates have climbed to very high levels, learning is nevertheless conducted on an ongoing basis to ensure that the agent's content knowledge is updated as the products and services offered by the company change.

Typically, three mechanisms have been put in place for quality improvement. First, transcripts of conversations are logged for offline analysis. This "conversation mining" is performed automatically and augmented with a subject matter expert's input. Second, enterprise-class conversational agent systems include authoring suites that support semi-automated assessment of the agent's performance. These suites exploit linguistic knowledge to summarize a very large number of questions posed by users since the most recent review period (i.e., frequently on the order of several thousand conversations) into a form that is amenable to human inspection. Third, the conversational agent

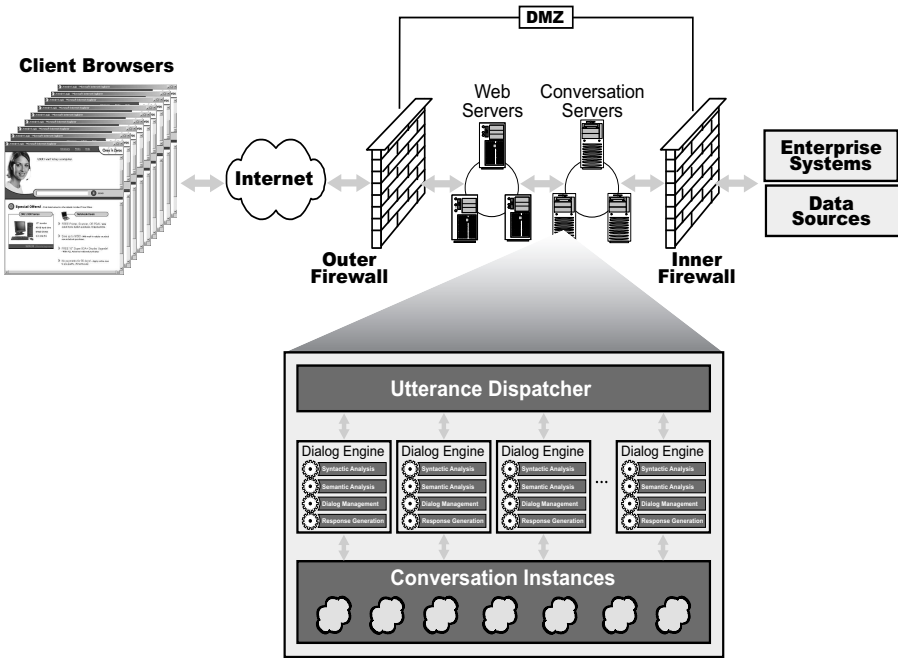

Figure 1.4: An enterprise deployment scheme for conversational agents

performs a continuous self-assessment to evaluate the quality of its behavior. For example, well-engineered conversational agents generate confidence ratings for each response which they then use both to improve their performance and to shape the presentation of the summarized logs for review.

1.4.2 Enterprise Integration Technologies

Conversational agents satisfy the scalability, performance, reliability, security, and integration requirements by employing the deployment scheme depicted in Figure 1.4. They should be deployed in an n-tier architecture in which clustered conversational components are housed in application-appropriate security zones. When a user's utterance is submitted via a browser, it is transported using HTTP or HTTPS. Upon reaching the enterprise's outermost firewall, the utterance is sent to the appropriate Web server, either directly, or using a dedicated hardware load balancer. When a Web server receives the utterance, it is submitted to a conversation server for processing. In large deployments, submission to conversation servers must themselves be load balanced.

When the conversation server receives the utterance, it determines whether a new conversation is being initiated, or whether the utterance belongs to an ongoing conversation. For new conversations, the conversation server creates a conversation instance. For ongoing conversations, it retrieves the corresponding conversation instance, which contains the state of the conversation, including the dialog history.[5] Next, the conversation server selects an available dialog engine and passes the utterance and conversation instance to it for interpretation, dialog management, and response generation. In some cases, the conversation server will invoke business logic and access external data to select the appropriate response or take the appropriate action. Some business rules and data sources will be housed behind a second firewall for further protection. For example, a conversation server may use the CRM system to inspect the user's profile or to open a trouble ticket and populate

[5]For deployments where conversations need to be persisted across sessions (*durable conversations*), the conversation server retrieves the relevant *dormant* conversation by indexing on the user's identification and then re-initiating it.

it with data from the current conversation. In the course of creating a response, the conversation agent may invoke a third-party content management system and personalization engines to retrieve (or generate) the appropriate response content.

Once language processing is complete, the conversation instance is updated and relevant data is logged into the conversation mart, which is used by the enterprise for analytics, report generation, and continued improvement of the agent's performance. The response is then passed back to the conversation server and relayed to the Web server, where an updated view of the agent presentation is created with the new response. Finally, the resulting HTML is transmitted back to the user's browser.

Scalability. This deployment scheme achieves the scalability objectives in three ways. First, each conversation server contains a pool of dialog engines. The number of dialog engines per server can be scaled according to the capabilities of the deployment hardware. Second, conversation servers themselves can be clustered, thereby enabling requests from the Web servers to be distributed across the cluster. Conversation instances can be assigned to any available conversation server. Third, storage of the knowledge base and conversation mart utilize industry-standard database scaling techniques to ensure that there is adequate capacity for requests and updates.

Performance. Conversational agents satisfy the performance requirements by providing a pool of dialog engines for each conversation server and clustering conversation servers as needed. Guarantees on throughputs are achieved by ensuring that adequate capacity is deployed within each conversation server and its dialog engine pool. Guarantees on the number of simultaneous conversations that can be held are achieved with the same mechanisms: if a large number of utterances are submitted simultaneously, they are allocated across conversation servers and dialog engines. Well-engineered conversational agents are deployed on standard enterprise-class servers. Typical deployments designed to comfortably handle up to hundreds of thousands of questions per hour consist of one to four dual-processor servers.

Reliability. A given enterprise can satisfy the reliability and availability requirements by properly replicating conversation resources across a sufficient number of conversation servers, Web servers, and databases, as well as by taking advantage of the fault tolerance mechanisms employed by enterprise servers. Because maintaining conversation contexts, including dialog histories, is critical for interpreting utterances, in some deployments it is particularly important that dialog engines be able to access the relevant context information, but nevertheless be decoupled from it for purposes of reliability. This requirement is achieved by disassociating conversation instances from individual dialog engines.

Security. The deployment framework achieves the security requirements through four mechanisms. First, conversational traffic over the Internet can be secured via HTTPS. Second, conversation servers should be deployed within a DMZ to provide access by Web servers but to limit access from external systems. Depending on the level of security required, conversation servers are sometimes placed behind internal firewall to increase security. Third, using industry standard authentication and authorization mechanisms, information in the knowledge base, as well as data in the conversation mart, can be secured from unauthorized access within the organization. For example, the content associated with particular knowledge base entries should be modified only by designated subject matter experts within a specific business unit. Finally, for some conversational applications, end users may need to be authenticated so that only content associated with particular roles is communicated with them.

Integration. Conversational agents in the framework integrate cleanly with the existing IT infrastructure by exposing agent integration APIs and accessing and utilizing APIs provided by other enterprise software. They typically integrate with J2EE- and .NET-based Web services to invoke enterprise-specific business logic, content management systems, personalization engines, knowledge management applications, and CRM modules for customer segmentation and contact center management. In smaller environments it is also useful for conversational agents to access third-party databases (housing, for example, product catalogs and customer records) via mechanisms such as

JDBC and ODBC.

In summary, well-engineered conversational agents utilizing the deployment scheme described above satisfy the high-volume conversation demands experienced in the enterprise. By housing dialog engines in a secure distributed architecture, the enterprise can deliver a high throughput of simultaneous conversations reliably, integrate effortlessly with the existing environment, and scale as needed.

1.5 Conclusion

With advances in computational linguistics, well-engineered conversational agents have begun to play an increasingly important role in the enterprise. By taking advantage of highly effective parsing, semantic analysis, and dialog management technologies, conversational agents clearly communicate with users to provide timely information that helps them solve their problems. While a given agent cannot hold conversations about arbitrary subjects, it can nevertheless engage in productive dialogs about a specific company's products and services. With large-scale deployments that deliver high volumes of simultaneous conversations, an enterprise can employ conversational agents to create a cost-effective solution to its increasing demands for customer service, guided selling, website navigation, and technical support. Unlike the monolithic CRM systems of the 1990s, which were very expensive to implement and whose tangible benefits were questionable, self-service solutions such as conversational agents are predicted by analysts to become increasingly common over the next few years. Because well-engineered conversational agents operating in high volume environments offer a strong return on investment and a low total cost of ownership, we can expect to see them deployed in increasing numbers. They are currently in use in large-scale applications by many Global 2000 companies. Some employ external-facing agents on retail sites for consumer products, while others utilize internal-facing agents to assist customer service representatives with support problems.

To be effective, conversational agents must satisfy the linguistic and enterprise architecture requirements outlined above. Without a robust language processing facility, agents cannot achieve accuracy rates necessary to meet the business objectives of an organization. Conversational agents that are not scalable, secure, reliable, and interoperable with the IT infrastructure cannot be used in large deployments.

In addition to these two fundamental requirements, there are three additional practical considerations for deploying conversational agents. First, content reuse is critical. Because of the significant investment in the content that resides in knowledge management systems and on websites, it is essential for conversational agents to have the ability to leverage content that has already been authored. For example, conversational agents for HR applications must be able to provide access to relevant personnel policies and benefits information. Second, all authoring activities must be simple enough to be performed by non-technical personnel. Some early conversational agents required authors to perform scripting or programming by authors. These requirements are infeasible for the technically untrained personnel typical of the divisions in which agents are usually deployed, such as customer care and product management. Finally, to ensure a low level of maintenance effort, conversational agents must provide advanced learning tools that automatically induce correct dialog behaviors. Without a sophisticated learning facility, maintenance must be provided by individuals with technical skills or by professional service organizations, both of which are prohibitively expensive for large-scale deployments.

With advances in the state-of-the-art of their foundational technologies, as well as changes in functionality requirements within the enterprise, conversational agents are becoming increasingly central to a broad range of applications. As parsing, semantic analysis, and dialog management

capabilities continue to improve, we are seeing corresponding increases in both the accuracy and fluidity of conversations. We are also seeing a gradual movement towards multilingual deployments. With globalization activities and increased internationalization efforts, companies have begun to explore multilingual content delivery. Over time, it is expected that conversational agents will provide conversations in multiple languages for language-specific website deployments. As text-mining and question-answering capabilities improve, we will see an expansion of agents' conversational abilities to include an increasingly broad range of "source" materials. Coupled with advances in machine learning, these developments are further reducing the level of human involvement required in authoring and maintenance. Finally, as speech recognition capabilities improve, we will begin to see a convergence of text-based conversational agents with voice-driven help systems and IVR. While today's speech-based conversational agents must cope with much smaller grammars and limited vocabularies—conversations with speech-based agents are much more restricted than those with text-based agents—tomorrow's speech-based agents will bring the same degree of linguistic proficiency that we see in today's text-based agents. In short, because conversational agents provide significant value, they are becoming an integral component of business processes throughout the enterprise.

References

Steven Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1995.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 86–90, San Francisco, California, 1998. Morgan Kaufmann Publishers.

Eric Brill. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

British national corpus, 2003. `http://www.natcorp.ox.ac.uk/`.

Claire Cardie, Scott Mardis, and David Pierce. Combining error-driven pruning and classification for partial parsing. In *Proceedings of the 16th International Conference on Machine Learning*, pages 87–96. Morgan Kaufmann, San Francisco, CA, 1999.

Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, 1997.

Christiane Fellbaum, editor. *Wordnet: An Electronic Lexical Database*. The MIT Press, 1999.

Charles Fillmore. The case for case. In *Universals in Linguistic Theory*, pages 1–90. Holt, Rinehart & Winston, New York, 1968.

Sanda Harabagiu, Marius Pasca, and Steven Maiorano. Experiments with open-domain textual question answering. In *Proceedings of COLING-2000*, Saarbrüken Germany, August 2000.

LDC catalog, 2003. `http://www.ldc.upenn.edu/Catalog/`, University of Pennsylvania.

Text retrieval competition, 2003. National Institute of Standards and Technology, `http://trec.nist.gov/`.

Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics, Somerset, New Jersey, 1996.

Proceedings of the 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization, Venice, Italy, June 23–24 2003.

Ehud Reiter. NLG vs. templates. In *Proceedings of the Fifth European Workshop on Natural-Language Generation*, Leiden, The Netherlands, 1995.

John Searle. *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge University Press, 1979.

Statistical natural language processing and corpus-based computational linguistics: An annotated list of resources, 2003. Stanford University, `http://www-nlp.stanford.edu/links/statnlp.html`.

Joseph Weizenbaum. ELIZA - a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery*, 9(1): 36–45, 1966.