Chapitre 1

Natural Language and Dialogue Processing

1.1 Introduction

Natural Language Processing (NLP) traditionally involves the manipulation of written text and abstract meanings, aiming at relating one to the other. It is of major interest in multimodal interfaces since written inputs and outputs are important means of communication but also because speech-based interaction is often made via textual transcriptions of spoken inputs and outputs. Whatever the information supports, the interaction is based on the assumption that both participants do understand each other and produce meaningful responses. Speech recognition and speech synthesis are therefore mandatory but not sufficient to build a complete voice-enabled interface. Humans and machines have to manipulate meanings (either for understanding or producing speech and text) and to plan the interaction. In this contribution, NLP will be considered under three points of view. First, Natural and Spoken Language Understanding (NLU or SLU) aiming at extracting meanings from text or spoken inputs. Second Natural Language Generation (NLG) which involves the transcription of a sequence of meanings into a written text. Finally, Dialogue Processing (DP) that analyses and models the interaction from the system's side and feeds NLG systems according to extracted meanings from users' inputs by NLU.

1.2 Natural Language Understanding

It is the job of a Natural Language Understanding (NLU) system to extract meanings of text inputs. In the case of spoken language understanding, previous processing systems (such as Automatic Speech Recognition : ASR) are error-prone and can add semantic noise such as hesitations, stop words etc. This has to be taken into account. To do so, some NLU systems are closely coupled to the ASR system, using some of its internal results (Nbest lists, lattices or confidence scores). Others maintain several hypotheses so as to propagate uncertainty until it can be disambiguated by the context.

Assuming that the input is a correct word sequence, most of NLU systems

can be decomposed in three steps : syntactic parsing, semantic parsing and contextual interpretation. In the following a brief overview of each step is given. For further details about the basic ideas and methods of NLU, readers are invited to refer to [5].

1.2.1 Syntactic Parsing

Before trying to extract any meaning out of a sentence, the syntactic structure of this sentence is generally analyzed : the function of each word (part of speech), the way words are related to each other, how they are grouped into phrases and how they can modify each other. It helps resolving some ambiguities as homographs (homophones) having different possible functions. For instance, the word "fly" can be a noun (the insect) or a verb and the word "flies" can stand for the plural form of the noun or an inflexion of the verb as shown on Fig. 1.1. Most syntactic representations of language are based on the notion of

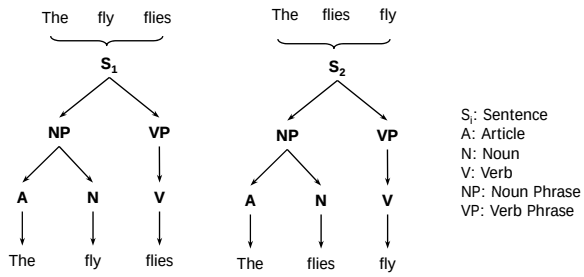

FIG. 1.1 – Syntactic Parsing

Context-Free Grammars (CFG) [14]. Sentences are then split in a hierarchical structure (Fig. 1.1). Most of early syntactic parsing algorithms, aiming at creating this *parse tree*, were developed with the goal of analysing programming language rather than natural language [4]. Two main techniques for describing grammars and implementing parsers are mainly used : *context-free rewrite rules* and *transition networks* [87].

For instance, a grammar capturing the syntactic structure of the first sentence in Fig. 1.1 can be expressed by a set of rewrite rules as follows :

1. S → NP VP
2. NP → A N
3. VP → V
4. A → The
5. N → fly
6. V → flies

The BNF (Backus-Nauer Form)[35] notations are often used to express those rules. First three rules are called *non-terminal rules* because they can still be

decomposed and others are called *terminal rules* (or terminal symbols). Rewrite rules are very similar to those used in logical programming (like PROLOG). This is why logical programming has been widely used for implementing syntactic parsers [23].

The above grammar can also be put into the form of the following State-Transition Network (STN) : Natural languages often involve restrictions in the

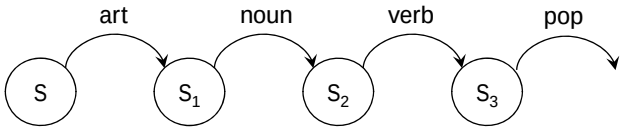

FIG. 1.2 – Transition Network Grammar

combinations of words and phrases. Many forms of agreements exists such as number agreement, subject-verb agreement, gender agreement, etc. For instance, the noun phrase "a cats" is not correct in English, as it doesn't satisfy the number agreement restriction. The previously described formalisms can be extended to handle these restrictions. Constituents are then associated to features (like number, gender, etc) and this leads to *augmented grammars*. There are Augmented CFGs and Augmented Transition Networks (ATN).

CFGs and ATNs have been the topic of extensive research. Yet, they are still limited because often handcrafted and deterministic. Coping with ambiguities is made uneasy when several interpretations are possible. With the emergence of data-driven and statistical techniques came other solutions to the parsing problem [31]. For instance, the part-of-speech tagging problem can be expressed as the problem of selecting the most likely sequence of syntactic categories $(C_1, ..., C_T)$ for the words $(w_1, ..., w_T)$ in a sentence, that is the sequences $(C_1, ..., C_T)$ that maximizes the probability :

$$P(C_1, C_2, ..., C_T | w_1, w_2, ..., w_T)$$

Using Bayes' rule :

$$P(C_1, C_2, ..., C_T | w_1, w_2, ..., w_T) = \frac{P(w_1, w_2, ..., w_T | C_1, C_2, ..., C_T) P(C_1, C_2, ..., C_T)}{P(w_1, w_2, ..., w_t)}$$

The solution to the syntactic parsing problem is :

$$\underset{C_1, C_2, ..., C_T}{\operatorname{argmax}} \; P(w_1, w_2, ..., w_T | C_1, C_2, ..., C_T) P(C_1, C_2, ..., C_T),$$

since the the denominator remains constant. This is generally not computable because probabilities involved in this expression are hardly computable and even difficult to estimate from data in the case of long sentences.

Two assumptions are then generally made. First, let's assume that $P(C_1, ..., C_T)$ can be approximated by a n-gram model. That is assuming that the probability of occurrence of category C_i only depends on the $n-1$ previous categories :

$$P(C_i | C_{i-1}, ..., C_0) = P(C_i | C_{i-1}, ..., C_{i-(n-1)})$$

A bigram model is often adopted and only $P(C_i|C_{i-1})$ is estimated, which leads to :

$$P(C_1, C_2, ..., C_T) = P(C_1) \prod_{i=2}^{T} P(C_i|C_{i-1})$$

Second, let's assume that a word appears in a category independently of the word in the preceding and the succeeding categories :

$$P(w_1, w_2, ..., w_T|C_1, C_2, ..., C_T) = \prod_{i=1}^{T} P(w_i|C_i)$$

In this case, each $P(w_i|C_i)$ can be compared to the emission probability of the ASR problem while each $P(C_i|C_{i-1})$ can be compared to a transition probability and Hidden Markov Models (HMM) can therefore serve as a probabilistic model. The part-of-speech tagging problem can be translated into the search of the sequence $(C_1, ..., C_T)$ that maximizes :

$$P(w_1|C_1)P(C_1) \prod_{i=2}^{T} P(w_i|C_i)P(C_i|C_{i-1})$$

All the probabilities in the above equation can be estimated using a manually annotated data corpus (even if unaligned, thanks to the EM algorithm) and the problem can then be solved with the same tools used for solving the ASR problem (a Viterbi algorithm, for instance).

When dealing with spoken language, syntactic parsing is often inefficient because of speech recognition errors, hesitations etc [70]. This is why semantic-only-based methods have been investigated for spoken language understanding.

1.2.2 Semantic Parsing

The role of a semantic parser is to extract the context-independent meaning of a written sentence. For instance, the noun phrase "*video cassette recorder*" has a single meaning and refers to the device that records and plays back video tapes, whatever the context. In the same way, albeit the word "*flies*" is ambiguous, once it has been correctly identified as a verb by the syntactic parser it doesn't need any contextual interpretation to reveal its meaning (yet, syntactic parsing was required).

However, in a large number of cases, a single word can endorse several meanings that cannot be disambiguated by a syntactic parsing. The other way around, a same meaning can also have several realisations (synonyms). Often, some meanings of a specific word can be eliminated at the sentence level because they do not fit the direct context of the word. For instance, in the sentence "*John wears glasses*", the word "*glasses*" means spectacles and not receptacles containing a liquid, because John cannot "wear" receptacles. Remaining ambiguities are context-dependent and will be considered in the next sub-section.

Given this, it appears that semantic interpretation ressemble to a classification process aiming at categorizing words or groups of words in classes regrouping synonyms. The set of classes in a particular representation of the world (or at least of a domain) is called its *taxonomy* while the relationships between those classes is the *ontology*. Such classifications have been of interest for a very

long time and arise in Aristotle's writings in which he suggested the following major classes : substance, quantity, quality, relation, place, time, position, state, action, affection. Some information have to be enclosed in the semantic representation. For example, the verb is often described as the word in a sentence that expresses the action, while the subject is described as the actor. Other roles can be defined and are formally called *thematic roles*. Semantic representations should therefore enclose information about thematic roles.

Utterances are not always used to make simple assertions about the world. The sentence "*Can you pass the salt ?*" is not a question about the ability of someone to pass the salt but rather a request to actually pass the salt. Some utterances have therefore the purpose of giving rise to reactions or even of changing the state of the world. They are the observable performance of communicative actions. Several actions can be associated to utterances like assertion, request, warning, suggestion, informing, confirmation etc. This is known as the *Speech Act* theory and has been widely studied in the field of philosophy [7]. When occuring in a dialogue, they are often refered as *Dialog Acts*. The Speech Act theory is an attempt to connect language to goals. In the semantic representation of a utterance speech acts should therefore be associated (often one speech act per phrase). Computational implementations of this theory have early been developed [16]. Philosophy provided lots of other contributions to natural language analysis and particularly in semantics (like the lambda calculus, for example) [49].

From this, Allen proposes the logical form [5] framework to map sentences to context independent semantic representations. Although it is not exploited in every system, lots of currently used semantic representation formalisms are related to logical forms in one way or another. Allen also proposes some methods to automatically transfer from a syntactic representation to the corresponding logical form.

Statistical data driven methods such as Hidden Markov Models (HMM) previously used in speech recognition and syntactic parsing have also been applied to semantic parsing of spoken utterances [51]. Semantic parsing can indeed be expressed as the problem of selecting the most likely sequence of concepts $(C_1, ..., C_t)$ for the words $(w_1, ..., w_T)$ in a sentence. This problem can be solved considering that the concepts are hidden states of a stochastic process while the words are observations. Notice that this technique bypasses the syntactic parsing and can be adapted to spoken language understanding. More recently, using the same idea, other generative statistical models as dynamic Bayesian networks were used to infer semantic representations [27]. The advantage of such models is that they do not need to be trained on manually aligned data which is time-consuming to obtain.

One disadvantage of generative methods is that they are not trained discriminatively like supervised methods. They also make some independence assumptions over the features. This results generally in lower performances in classification tasks. Discriminative methods such as Support Vector Machines (SVM) have therefore been applied to the problem of semantic parsing [61]. But these last methods require accurate labeled data which are generally not available. Consequently, combination of discriminative and generative models trained on unaligned data has recently been introduced [64, 46].

Other statistical methods such as decision trees and CART (Classification and Regression Trees) are also often part of NLU systems for semantic parsing

[32]. They extract a so-called semantic- or parse-tree from a sentence and aim at finding the most probable tree that fits the sentence.

1.2.3 Contextual Interpretation

Contextual interpretation takes advantage of information at the discourse level to refine the semantic interpretation and to remove remaining ambiguities. Here, the term discourse defines any form of multi-sentence or multi-utterance language. Three main ambiguities can be resolved at the discourse level :

Anaphors are parts of a sentence that typically replace a noun phrase (e.g. the use of the words "*this*", "*those*",...). Several types of anaphors can be cited such as intrasentence (the reference is in the same sentence), intersentence (the reference is in a previous sentence) and surface anaphors (the reference is evoked but not explicitly referred in any sentence of the discourse).

Pronouns are particular cases of anaphors and typically refer to noun phrases. Lots of specific studies have addressed this kind of anaphora.

Ellipses involve the use of clauses (parts of discourse that can be considered as stand-alone) that are not syntactically complete sentences and often refer to actions or events. For instance, the second clause (B) in the following discourse segment is an ellipsis :
- A. John went to Paris.
- B. I did too.

Although there exist a wide range of techniques for resolving anaphors, they are almost all based on the concept of *Discourse Entity* (DE) [34]. A DE can be considered as a possible antecedent for an unresolved ambiguity in the local context and it is typically a possible antecedent for a pronoun. The system maintains a DE list which is a set of constants referring to objects that have been evoked in previous sentences and can subsequently be referred implicitly. A DE is classically generated for each noun phrase.

Simple algorithms for anaphora resolutions based on DE history exist (the last DE is the most likely to be referred to). Yet, the most popular methods are based on a computational model of discourse focus [72] that evolved into the centering model [83]. These theories rely on the idea that the discourse is organized around an object (the center), that the discourse is about, and that the center of a sentence is often pronominalised. The role of the system is then to identify the current center of the discourse and to track center moves.

1.3 Natural Language Generation

Natural Language Generation (NLG) aims at producing understandable text from non-linguistic representations of information (concepts). Research in this field started in the 1970's, that is later than NLU or ASR. Thus, reference books are quite recent [66]. Applications of NLG are automatic documentation of programming language [38], summarization of e-mails, information about weather forecast [25], and spoken dialogue systems [63, 28, 30]. Many NLG techniques exist and particularly when the generated text is intended to be used for speech synthesis (concept-to-speech synthesis).

In most of industrial spoken dialogue systems, the NLG sub-system is often very simple and can be one of the following :

Pre-recorded prompts : sometimes real human voice is still preferred to TTS systems because it is more natural. This is only possible if the set of concepts that have to be managed is small enough. In this case, a simple table mapping concepts to corresponding audio records is built. This technique has a lot of drawbacks as the result is static and recording human speech is often expensive.

Human authoring : the idea is approximately identical except that the table contains written text for each possible concept sequence. The text is subsequently synthesized by a TTS system. This solution is more flexible as written text is easier to produce and modify but still needs human expertise. This technique is still extensively used.

Although using one of those techniques is feasible and widely done in practice, it is not suitable for certain types of dialogues like tutoring, problem-solving or troubleshooting for example. In such a case, the system utterances should be produced in a context-sensitive fashion, for instance by pronominalising anaphoric references, and by using more or less sophisticated phrasing or linguistic style [80, 30] depending on the state of the dialogue, the expertise of the user, etc. Therefore, more complex NLG methods are used and the process is commonly split into three phases [66] : document planning, microplanning and surface realisation. This first phase is considered as language independent while the two following are language dependent.

1.3.1 Document Planning

NLG is a process transforming high-level communicative goals into a sequence of communicative acts (Speech Acts), which accomplish the initial communicative goals. The job of the document-planning phase (or text-planning phase) is to brake high-level communicative goals into structured representations of atomic communicative goals that can be attained with a single speech act (in language, by uttering a single clause). Document planning results in an first overview of the document structure to be produced. It also creates an inventory of the information contained in the futur document. It is very task-dependent and techniques used for document planning are closely related to expert systems techniques.

1.3.2 Microplanning

During the microplanning (or sentence planning) phase, abstract linguistic resources are chosen to achieve the communicative goals (lexicalisation). For instance, a particular verb is selected for expressing an action. An abstract syntactic structure for the document is then built (aggregation). Microplanning is also the phase were the number of generated clauses is decided. So as to produce language with improved naturalness, a last process focuses on referring expressions generation (e.g. pronoun, anaphora). Indeed, if several clauses of the generated document refer to the same discourse entity, the entity can be pronominalized after the first reference. The result is a set of phrase prototypes (or *proto-phrases*)

Microplanning can be achieved using several methods. Some of them are based on *reversed parsing*, that is semantic grammars used for parsing in NLU are used in a generative manner to produce document prototypes [71]. Most of probabilistic grammar models are generative models that can be used in this purpose. The idea seems attractive but the development of semantic grammars suitable for both NLG and NLU appeared to be very tricky and in general, same meanings will always lead to the same generated proto-phrases (which is unnatural). Other techniques are based on grammar specifically designed for NLG purposes. In spoken dialogue systems, template-based techniques are widely used in practice [47] but learning techniques such as boosting or even reinforcement learning have been recently successfully applied to sentence planning [85, 30].

1.3.3 Surface Realisation

During surface realisation, the abstract structure (proto-phrases) built during microplanning are transformed into surface linguistic utterances by adding function words (such as auxiliaries and determiners), inflecting words, building number agreements, determining word order, etc. The surface realization process even more than the previous one is strongly language dependent and uses resource specific to the target language. This phase is not a planning phase in that it only executes decisions made previously, by using grammatical information about the target language.

1.4 Dialogue Processing

Human-human dialogues are mostly multimodal mixing speech, gesture, drawings but also semantic and pragmatic knowledge. When a person engages a conversation with another, information coming from all senses are combined to background knowledge to understand the other interlocutor. Although we focus here on speech-based dialogues, it is still without loss of generality. We also focus on goal-directed dialogues, that is dialogues in which both participants collaborate to achieve a goal. Social dialogue is not in the scope of this contribution. A spoken dialog system is not only the combination of speech and natural language processing systems. The interaction has to be managed in some way. The role of the dialogue management component in a man-machine interface is to organize the interaction in terms of sequencing. A man-machine dialog is here considered as a turn-taking process in which information is transfered from one participant to the other through (possibly noisy) channels (ASR, NLU, NLG and TTS). One partcipant is the user, the other being the Dialog Manager (DM, see Fig. 1.3). The DM has thus to model the dialogue progress over time but also its structure and the discourse structure.

1.4.1 Discourse modeling

Human-human dialogue have been the topic of intensive investigations since the beginning of artificial intelligence research. One aim of these investigations was the development of artificial spoken dialogue systems. Discourse modelling aimed at developping a theory of dialogue, including, at least, a theory of cooperative task-oriented dialogue, in which the participants are communicating

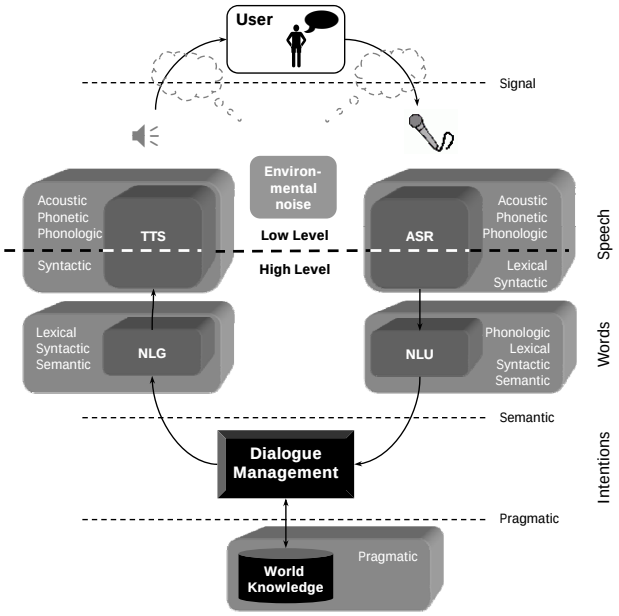

FIG. 1.3 – Spoken Dialogue System

in the aim of achieving some goal-directed task. Although there is no common agreement on the fact that human-human dialogues should serve as a model for human-machine dialogues (users' behaviour being often adapted when talking to machines [33, **?**]), several approaches to modelling human-machine dialogues coming from studies about human-human dialogues can be cited such as dialogue grammar models, plan-based models, conversational game models and joint action models.

Dialogue grammars are an approach with a long history. This approach considers that regularities exist in a dialogue sequencing and that some Speech Acts (or Dialogue Acts) can only follow some others [65]. For example, a question is generally followed by an answer. In the vein of the NLU process, a set of rules imposes sequential and hierarchical constraints on acceptable dialogues, just as syntactic grammar rules state constraints on grammatically acceptable utterances. The major drawback of this approach (and what makes it so simple) is that it considers that the relevant history for the current dialogue state is only the previous sentence (the previous Speech Act or Dialogue Act). For this reason only few systems were successfully based on this model [17].

Plan-based models are founded on the observation that humans do not just perform actions randomly, but rather they plan their actions to achieve various goals, and in the case of communicative actions (Speech Acts), those goals in-

clude changes to the mental state of listeners. Plan-based models of dialogue assume that the speaker's Speech Acts are part of a plan, and the listener's job is to uncover and respond appropriately to the underlying plan, rather than just to the previous utterance [12]. In other words, when inferring goals from a sentence, the system should take the whole dialogue into account and interpret the sentence in the context of the plan. Those models are more powerful than dialogue grammars but goal inference and decision making in the context of plan-based dialogue are sometimes very complex [11]. Nevertheless, plan-based dialogue models are used in SDSs [22].

The *Conversational Game Theory* (CGT) [60] is an attempt to combine the ideas from both plan-based models and dialogue grammars in a same framework. Dialogues are then considered as exchanges called *games*. Each game is composed with a sequence of *moves* that are valid according to a set of rules (similar to grammars) and the overall game has a goal planned by participants (similar to plan-based models). Participants share knowledge (beliefs and goals) during the dialogue and games can be nested (sub-dialogues are possible) to achieve sub-goals. Moves are most often assimilated to Speech or Dialogue Acts [1]. The CGT defines quite formally the moves allowed for each of the participants at a given state in the game (according to the rules and the goal) and simulation of dialogues is made possible as well [60]. Computational versions of CGT have been succesfully applied to dialogue systems [42].

The previous approaches considered a dialogue as a product of the interaction of a plan generator (the user) and a plan recognizer (the dialogue manager) working in harmony, but it does not explain why participants ask clarification questions, why they confirm etc. This process used by dialogue participants to ensure that they share the background knowledge necessary for the understanding of what will be said later in the dialogue is called *grounging* [15]. According to this, another dialogue model has emerged in which dialogue is regarded as a *joint activity* shared by participants. Engaging in a dialogue requires the interlocutors to have at least a joint agreement to understand each other, and this motivates the clarifications and confirmations so frequent in dialogues. Albeit this family of dialogue models is of a great interest, its complexity makes it diffiicult to implement [20].

1.4.2 Dialogue Management

Any of the dialogue models described earlier can be used in SDSs in order to interpret semantics or build internal state, yet the dialogue management can be of several kinds. A dialogue manager implements an interaction strategy aiming at organizing the sequencing of systems dialogue acts so as to achieve the common goal of the user and the system. In the following different dialogue management methods found in the literature are further described. Although not exhaustive, this list is representative of what exists in most current spoken dialogue systems.

Theorem Proving

The *theorem proving* approach to dialogue management has been developed in [74] in the framework of a problem-solving (or troubleshooting) SDS for equipment fixing. The idea underlying this method is that the system tries

to demonstrate that the problem is solved (theorem). As in mathematical demonstration, there are several steps to follow. At each step, the system can use *axioms* (things known for sure) or deductions (obtained by logic inference) to move on. If in a given dialogue state, the *World Knowledge* included in the system does not contain an axiom allowing the dialogue manager to go on or if the dialogue manager cannot infer it from other axioms, the axiom is considered as missing (*missing axiom theory*). The system thus prompts the user for more information. If the user is not able to provide information, a new theorem has to be solved : "*the user is able to provide relevant information*". A tutoring sub-dialogue then starts. This technique has been implemented in Prolog as this rule-based method is suitable for Logic Programming. The major drawback of theorem proving management is that the strategy is fixed as the demonstration steps are written in advance. Also this method is strongly task-dependent and rules can hardly be reused accross domains.

Finite State Machine

In the case of *finite-state* methods, the dialogue is represented like a state-transition network where transitions between dialogue states specify all legal paths through the network. In each state an action is chosen according to the dialogue strategy. Actions can be greeting the user, request information, provide information, ask for confirmation, closing the dialogue etc. The result of the action leads to a new transition between states. The construction of a dialogue state is a tricky problem. Usually, dialogue states are built according to the information that as been exchanged between the user and the system. In this case, the state is said *informational*. For example, if the aim of a dialogue system is to retrieve the first and familly names of a person, 4 dialogue states can be built :

- both names are unknown,
- only the firstname is known,
- only the familly name is known,
- both names are known.

The main drawback of these methods is that all possible dialogues, that is all paths in the state space, have to be known and described by the designer. The structure is mostly static resulting in inflexible and often system-led behaviours. Nevertheless, state-transition methods have been widely used in dialogue systems but above all in various toolkits and authoring environments [48] for several reasons. It is an easy manner for modelling dialogues that concern well-structured tasks that can be mapped directly on to a dialogue structure. Finite-state methods are easier to understand and more intuitive for the designer as it provides a visual, global and ergonomic representation. Scripting languages as VoiceXML [77] can be very easily used for representing state-transition networks. As discussed in [48], lots of applications like form-filling, database querying or directory interrogation are more successfully processed this way. Those applications are the most popular. Eventually, nested sub-dialogues may help to gain in flexibility in this kind of methods.

Form Filling

Form filling methods, also called *frame-driven*, *frame-based* or *slot-based* methods, are suitable when the goal of the application consists in a one-way transfer of information from the user to the SDS. The information has to be represented as a set of attribute-value pairs. The attribute-value structure can be seen as a form and the user should provide values for each field (attribute, slot, frame) of the form. Each empty field is then eligible for a prompt from the system to the user. The dialogue strategy aims at filling completely the form and to retrieve values for all the fields. Each field is given a priority defining the sequence in which the user is prompted [24]. Each of the user's utterance has then to be processed in order to provide an attribute-value representation of its meaning.

Self-Organized Management

Unlike above-cited techniques, the *self-organized* management does not require all the dialogue paths to be specified in advance. Each action of the system and reaction of the user contributes to build a new configuration to which is associated a particular behaviour. There is no need to know how the configuration occurred. This is generally referred to as event-driven dialogue management. One famous attempts to use this kind of dialogue management method was the Philips Speechmania software based on the HDDL language (Harald's Dialogue Description Language, from the first name of its creator) [6]. Other attempts were made to use event-driven dialogue management [8] but the complexity of development of such systems overcame their potentialities.

Markov Decision Processes and Statistical Optimization

Because rapid development and reusability of previously designed systems are almost impossible with above-cited management techniques but also because objective assessment of performance are very difficult, machine learning methods for automatic search of optimal interaction policies have been developed during the last decade. This development is also due to the fact that a dialogue strategy has to take into account varying factors which are difficult to model like performances of the subsystems (like ASR, NLU, NLG, TTS etc.), the nature of the task (e.g. form filling [56], tutoring [26], robot control, or database querying [52]), and the user's behaviour (e.g. cooperativeness, expertise [53])

The main trend of research concerns reinforcement learning [76] methods in which a spoken dialogue is modelled as a *Markov Decision Process* (MDP) [40, 73, 53, 21] or a *Partially Observable MDP* [86, 88]. One concern for such approaches is the development of appropriate dialogue corpora for training and testing. However, the small amount of data generally available for learning and testing dialogue strategies does not contain enough information to explore the whole space of dialogue states (and of strategies). Therefore dialogue simulation is most often required to expand the existing dataset and man-machine spoken dialogue stochastic modelling and simulation has become a research field in its own right. Particularly, user simulation is a important trend of research [73, 56, 41, 45, 54, 68] but also NLU [55] and ASR [57, 67] simulation.

1.4.3 Degrees of Initiative

A major issue of dialogue strategy design is the degree of initiative let to the user in each dialogue state. Roughly speaking, the dialogue management will be easier if the control of the dialogue flow is complietly let to the system while a user will probably be more satisfied if he/she can be over-informative and take initiatives. Three degrees of initiative can be distinguished :

System-Led : the system controls completely the dialogue flow and asks sequences of precise questions to the user. The user is then supposed to answer those questions and provide only the information he/she has been asked for.

User-Led : the initiativ is complietly let to the user who asks for information to the system. The system is supposed to interpret correctly the user's query and to answer those precise questions without asking for more details.

Mixed Initiative : both participants (the user and the system) share the control to cooperate in order to achieve the conversation goal. The user can be over-informative and provide information he has not yet been asked for. He/she can ask the system to perform particular actions as well. The system can take the control at certain dialogue states so as not to deviate from the correct path leading to goal achievement. The dialogue manager can also decide to take the control because the performance of the previous systems in the chain (ASR, NLU) is likely to get poorer.

Instinctively, mixed-initiative systems are thougth to potentially perform better from the user's point of view. However, some research have shown that users sometimes prefer system-led SDSs because the goal achievement rate is greater [59]. Other studies exemplified that, besides the fact that system-led gives better performance with inexperienced users, the mixed-initiative version of the same system does not outperform the first (objectively neither subjectively) with more experienced users [81]. Indeed, evidence is made that human users adapt their behaviour because they know they are interacting with a machine. They usually accept some constraints in order to obtain a better goal completion rate.

1.4.4 Evaluation

Previous sections underlined fthe main issues in dialogue management design like the choices of a dialogue model, the degree of initiative etc. In the litterature, several studies lead to controversial conclusions about each of these issues. Moreover, objective evaluation criteria are used in machine learning methods for automatic optimization of dialogue strategies (see section 1.4.2). Consequently, SDS assessment methods became the topic of a large field of current researches.

Despite the amount of researches dedicated to the problem of SDS performance evaluation, there is no clear, objective and commonly accepted method to tackle this issue. Indeed, performance evaluation of such high-level communicative systems strongly relies on the opinion of end-users and is therefore strongly subjective and most probably task-dependent. Studies on subjective evaluation of SDSs through user satisfaction surveys have often (and early) been conducted [58]. The reproductivity of experiments and the independence between esperiments is hardly achieved. This is why those experiments have non-trivial interpretation. For instance, in [19] the authors demonstrate as a side-conclusion

that the users' appreciation of different strategies depends on the order in which SDSs implementing those strategies were presented for evaluation. Nevertheless, several attempts have been made to determine the overall system performance thanks to objective measures made on the sub-components, such as ASR performance. One one of the first tries can be found in [29]. Other objective measures taking into account the whole system behaviour (like the average number of turns per transaction, the task success rate etc.) have been exercised in the aim of evaluating different versions (strategies) of the same SDS [18] with one of the first application in the SUNDIAL project (and after within the EAGLES framework) [2]. More complex paradigms have been developed afterwards as described hereafter.

PARADISE

A popular framework for SDS evaluation is PARADISE (PARAdigm for DIalogue Systems Evaluation) [84]. This paradigm attempts to explain users' satisfaction as a linear combination of objective measures. For the purpose of evaluation, thec task is described as an attribute-value matrix. The user's satisfaction is then expressed as the combination of a task completion measure (κ) and a dialogue cost expressed computed as a weighted sum of objective measures (c_i). The overall system performance is then approximated by :

$$P(U) = \alpha N(\kappa) + \sum_i w_i N(c_i)$$

where N is a Z-score normalization function that normalises the results to have mean 0 and standard deviation 1. This way, each weight (α and w_i) expresses the relative contribution of each term of the sum to the performance of the system. The task completion measure κ is the Kappa coefficient [13] that is obtained from a confusion matrix M summarizing how well the transfer of information performed between the user and the system. M is a square matrix of dimension n (number of values in the attribute-value matrix) where each element m_{ij} is the number of dialogues in which the value i was interpreted while value j was meant. The κ coefficient is then defined as :

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the proportion of correct interpretations (sum of the diagonal elements of M (m_{ii}) on the total number of dialogues) and $P(E)$ is the proportion of correct interpretations occurring by chance. One can see that $\kappa = 1$ when the system performs perfect interpretation ($P(A) = 1$) and $\kappa = 0$ when the only correct interpretations were obtained by chance ($P(A) = P(E)$). So as to compute optimal weights α and w_i, test users are asked to answer a satisfaction survey after having used the system while costs c_i are measured during the interaction. The questionnaire includes around 9 statements to be rated on a five-point Likert scale and the overall satisfaction is computed as the mean value of collected ratings. A Multivariate Linear Regression (MLR) is then applied with the result of the survey as the dependent variable and the weights as independent variables. Criticisms can be made about assumptions and methods involved in the PARADISE framework. First, the assumption of independency between the different costs c_i made when building an additive

evaluation function has never been proven (it is actually false as the number of turns and the time duration of a dialogue session are strongly correlated [37]). The κ coefficient as a measure of task success rate can also be discussed, as it is often very difficult to compute when a large number of values are possible for a given attribute. In [10] this is examplified by the application of PARADISE to the PADIS system (Philips Automatic Directory Information System). The satisfaction questionnaire as also been the subject of criticisms. While [75] proposes to add a single statement rating the overall performance of the system on a 10-point scale, [37] recommends to rebuild the whole questionnaire, taking psychometric factors into account. Finally, the attribute-value matrix representation of the task has proven to be hard to extend to multimodal systems and thus, seems not to be optimal for system comparisons. Some attempts to modify PARADISE have been proposed to evaluate multimodal interfaces [9]. Besides the abovementioned criticisms, PARADISE has the advantage of being reproducible and involve automatic computations. It has thus been applied on a wide range of systems. It was adopted as the evaluation framework for the DARPA Communicator project and applied to the official 2000 and 2001 evaluation experiments [82]. However, experiments on different SDSs reached different conclusions. PARADISE developers themselves found contradicting results and reported that time duration was weakly correlated with user satisfaction in [79] while [78] reports that dialogue duration, task success and ASR performance were good predictors of user's satisfaction. On another hand, [62] reports a negative correlation between users' satisfaction and dialogue duration because users hung up when unsatisfied which reduces the dialogue length. Finally [36] surprisingly reports that ASR performance is not a so good predictor of user's satisfaction.

Analytical Evaluation

The principal drawback of PARADISE is the need of data collection. Indeed, subjective evaluation means that the system should be released to be evaluated and that a large number of test users have to be found. It is a time-consuming and expansive process that has to be done again for each prototype release. Although the usual process of SDS design obeys to the classical prototyping cycle composed of successive pure design and user evaluation cycles, there should be as little user evaluations as possible. This is why attempts to analyse strategies by mathematical means have been developed. In [43], the authors propose some ways to diagnose the future performance of the system during the design process. Other mathematical models of dialogue have been proposed [50] and closed forms of dialogue metrics (like the number of dialogue turns) have been proposed. Nevertheless, too few implementations were made to prove their reliability. Moreover, lots of simplifying assumptions have to be made for analytical evaluation and it is thus difficult to extend those methods to complex dialogue configurations and task-independency is hardly reachable.

Computer-Based Simulation

Because of inherent difficulties of data collection, some efforts have been done in the field of dialogue simulation for performance assessment. The purpose of using dialogue simulation for SDS evaluation is mainly to enlarge the

set of available data and to predict the behaviour of the SDS in unseen situations. Among simulation methods presented in the literature, one can distinguish between state-transition methods like proposed in [73] and methods based on modular simulation environments as described in [56, 41, 45, 54]. The first type of methods is more task-dependent as well as the hybrid method proposed in [69]. One can also distinguish methods according to the level of abstraction at which the simulation takes place. While [45, 44] models the dialog at the acoustic level, most of other methods [73, 41, 69, 54, 56] remain at the intention level, arguing that simulation of other levels can be infered from intentions.

1.5 Conclusion

In this chapter, we have described processing systems that are usually hidden to the user although essential for building speech- or text-based interfaces. All of these systems are still being the topic of intensive research and there exists room for improvement in performance. Especially, data-driven methods for optimizing end-to-end systems from speech recognition to text-to-speech synthesis are being investigated [39] albeit data collection and annotation is still a major problem. What is more, language processing is still limited to domain-dependent applications (such as troubleshooting, database access etc) and cross-domain or even cross-language methods are still far from being available. Also, transfer of academic research into the industrial world is still rare [3]. The search for efficiency often leads to hand-crafted and system-directed management strategies which are easier to understand and control.

Bibliographie

[1] *Modelling Cognition*, chapter Why to Speak, What to Say, and How to Say It, pages 249–267. Wiley, p. morris edition, 1987.

[2] N. F. A. Simpson. Black box and glass box evaluation of the sundial system. In *Proceedings of the 3rd European Conference on Speech Communication and Technology (Eurospeech'93)*, pages 1423–1426, Berlin (Germany), 1993.

[3] K. Acomb, J. Bloom, K. Dayanidhi, P. Hunter, P. Krogh, E. Levin, and R. Pieraccini. Technical support dialog systems, issues, problems, and solutions. In *Proeedings of the HLT 2007 Workshop on "Bridging the Gap, Academic and Industrial Research in Dialog Technology,"*, Rochester, NY (USA), April 2007.

[4] A. Aho and J. Ullman. *The Theory of Parsing, Translation, and Compiling.* Prentice-Hall, 1972.

[5] J. Allen. *Natural Language Understanding.* Benjamin Cummings, second edition, 1994.

[6] H. Aust and O. Schroer. An overview of the philips dialog system. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia (USA), February 1998.

[7] J. Austin. *How to Do Things with Words.* Harvard University Press, Cambridge, MA, 1962.

[8] A. Baekgaard. Dialogue management in a generic dialogue systems. In *TWLT-11 : Dialogue Management in Natural Language Systems*, pages 123–132, Netherlands, 1996.

[9] N. Beringer, U. Kartal, K. Louka, F. Schiel, and U. Türk. Promise - a procedure for multimodal interactive system evaluation. In *Proceedings of the Workshop on Multimodal Resources and Multimodal Systems Evaluation*, Las Palmas, Gran Canaria (Spain), 2002.

[10] G. Bouwman and J. Hulstijn. Dialogue strategy redesign with reliability measures. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pages 191–198, 1998.

[11] T. Bylander. Complexity results for planning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 274–279, 1991.

[12] S. Carberry. *ACL-MIT Press Series in Natural Language Processing*, chapter Plan Recognition in Natural Language Dialogue. Bradford Books, MIT Press, 1990.

[13] J. Carletta. Assessing agreement on classification tasks : the kappa statistic. *Computational Linguistics*, 22(2) :249–254, 1996.

[14] N. Chomsky. Three models for description of languages. *Transaction on Information Theory*, pages 113–124, 1956.

[15] H. Clarck and E. Schaefer. Contributing to discourse. *Cognitive Science*, 13 :259–294, 1989.

[16] P. Cohen and C. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3 :117–212, 1979.

[17] N. Dahlbäck and A. Jönsson. An empirically based computationally tractable dialogue model. In *Proceedings of the 14th Annual Conference of the Cognitive Science Society (COGSCI'92)*, Bloomington, Indiana (USA), July 1992.

[18] M. Danieli and E. Gerbino. Metrics for evaluating dialogue strategies in a spoken language system. In *Working Notes of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 34–39, Stanford, CA (USA), March 1995.

[19] L. Devillers and H. Bonneau-Maynard. Evaluation of dialog strategies for a tourist information retrieval system. In *Proceedings of the 5th International Conference on Speech and Language Processing (ICSLP'98)*, pages 1187–1190, Sydney (Australia), December 1998.

[20] P. Edmonds. A computational model of collaboration on reference in direction-giving dialogues. Master's thesis, Computer Systems Research Institute, Department of Computer Science, University of Toronto, October 1993.

[21] M. Frampton and O. Lemon. Learning more effective dialogue strategies using limited dialogue move features. *Proceedings of ACM*, 2006.

[22] R. Freedman. Plan-based dialogue management in a physics tutor. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 52–59, Seattle, WA (USA), 2000.

[23] G. Gazdar and C. Mellish. *Natural Language Programming in PROLOG*. Addison-Wesley, Reading, MA, 1989.

[24] D. Goddeau, H. Meng, J. Polifroni, S. Sene, and S. Busayapongchai. A form-based dialogue manager for spoken language applications. In *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP'96)*, pages 701–704, Philadelphia, PA (USA), 1996.

[25] E. Goldberg, N. Driedger, and R. Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert : Intelligent Systems and Their Applications*, 9(2) :45–53, April 1994.

[26] A. Graesser, K. VanLehn, C. Rosé, P. Jordan, and D. Harter. Intelligent tutoring systems with conversational dialogue. *AI Magazine*, 22(4) :39–52, 2001.

[27] Y. He and S. Young. Spoken language understanding using the hidden vector state model. *Speech Communication*, 48(3-4) :262–275, 2006.

[28] R. Higashinaka, M. Walker, and R. Prasad. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. *Journal of ACM Transactions on Speech and Language Processing*, 2007. in Press.

[29] L. Hirschman, D. Dahl, D. McKay, L. Norton, and M. Linebarger. Beyond class a : A proposal for automatic evaluation of discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 109–113, 1990.

[30] S. Janarthanam and O. Lemon. Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems. In *Proceedings of ENLG*, 2009.

[31] F. Jelinek. *Readings in Speech Recognition*, chapter Self-Organized Language Modeling for Speech Recognition, pages 450–506. Morgan Kaufmann, 1990.

[32] F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Ratnaparkhi, and S. Roukos. Decision tree parsing using a hidden derivation model. In *HLT '94 : Proceedings of the workshop on Human Language Technology*, pages 272–277, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[33] A. Jönsson and N. Dahlbäck. Talking to a computer is not like talking to your best friend. In *Proceedings of The 1st Scandinivian Conference on Artificial Intelligence*, Tromso (Nroway), March 1988.

[34] L. Kartunnen. *Syntax and Semantics 7*, chapter Discourse Referents, pages 363–385. Academic Press, 1976.

[35] D. E. Knuth. Backus normal form vs. backus naur form. *Communications of the ACM*, 7(12) :735–736, December 1964.

[36] L. Larsen. Combining objective and subjective data in evaluation of spoken dialogues. In *Proceedings of the ESCA Workshop on Interactive Dialogue in Multi-Modal Systems (IDS'99)*, pages 89–92, Kloster Irsee (Germany), 1999.

[37] L. Larsen. Issues in the evaluation of spoken dialogue systems using objective and subjective measures. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'03)*, St. Thomas (U.S. Virgin Islands), 2003.

[38] B. Lavoie, O. Rambow, and E. Reiter. Customizable descriptions of object-oriented models. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC, 1997.

[39] O. Lemon and O. Pietquin. Machine learning for spoken dialogue systems. In *Proceedings of the European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, Anvers (Belgium), August 2007.

[40] E. Levin, R. Pieraccini, and W. Eckert. Learning dialogue strategies within the markov decision process framework. In *Proceedings of the International Workshop on Automatic Speech Recognition and Understanding (ASRU'97)*, December 1997.

[41] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1) :11–23, 2000.

[42] I. Lewin. A formal model of conversational games theory. In *Proceedings of the 4th Workshop on the Semantics and Pragmatics of Dialogues, GOTALOG'00*, Gothenburg, 2000.

[43] D. Louloudis, K. Georgila, A. Tsopanoglou, N. Fakotakis, and G. Kokkinakis. Efficient strategy and language modeling in human-machine dialogues. In *Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI'01)*, volume 13, pages 229–234, Orlando, Fl (USA), 2001.

[44] R. López-Cózar, Z. Callejas, and M. F. McTear. Testing the performance of spoken dialogue systems by means of an artificially simulated user. *Artificial Intelligence Review*, 26(4) :291–323, 2006.

[45] R. López-Cózar, A. de la Torre, J. Segura, and A. Rubio. Assesment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40(3) :387–407, May 2003.

[46] F. Mairesse, M. Gašić, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, and S. Young. Spoken language understanding from unaligned data using discriminative classification models. In *Proceedings of ICASSP*, 2009.

[47] S. McRoy, S. Channarukul, and S. Ali. Text realization for dialog. In *Proceedings of the International Conference on Intelligent Technologies*, Bangkok (Thailand), 2000.

[48] M. McTear. Modelling spoken dialogues with state transition diagrams : Experiences with the cslu toolkit. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, Sydney (Australia), 1998.

[49] R. Montague. *Formal Philosophy*. Yale University, New Haven, 1974.

[50] Y. Niim and T. Nishimoto. Mathematical analysis of dialogue control strategies. In *Proceedings of the 6th European Conference on Speech Technology, (EuroSpeech'99)*, Budapest, September 1999.

[51] R. Pieraccini and E. Levin. Stochastic representation of semantic structure for speech understanding. *Speech Communication*, 11 :238–288, 1992.

[52] O. Pietquin. Machine learning for spoken dialogue management : an experiment with speech-based database querying. In J. Euzenat and J. Domingue, editors, *Artificial Intelligence : Methodology, Systems and Applications*, volume 4183 of *Lecture Notes in Artificial Intelligence*, pages 172–180. Springer Verlag.

[53] O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. SIMILAR Collection. Presses Universitaires de Louvain, 2004. ISBN : 2-930344-63-6.

[54] O. Pietquin. A probabilistic description of man-machine spoken communication. In *Proceedings of the International Conference on Multimedia and Expo (ICME'05)*, Amsterdam (Netherlands), July 2005.

[55] O. Pietquin and T. Dutoit. Dynamic bayesian networks for nlu simulation with application to dialog optimal strategy learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume I, pages 49–52, Toulouse (France), may 2006.

[56] O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2) :589–599, March 2006.

[57] O. Pietquin and S. Renals. Asr system modeling for automatic evaluation and optimization of dialogue systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, volume I, pages 45–48, Orlando, (USA, FL), may 2002.

[58] J. Polifroni, L. Hirschman, S. Seneff, and V. Zue. Experiments in evaluating interactive spoken language systems. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 28–33, Harriman, NY (USA), Ferbuary 1992.

[59] J. Potjer, A. Russel, L. Boves, and E. den Os. Subjective and objective evaluation of two types of dialogues in a call assistance service. In *Proceedings of the IEEE Third Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA 96)*, pages 89–92., 1996.

[60] R. Power. The organization of purposeful dialogues. *Linguistics*, 17 :107–152, 1979.

[61] S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of HLT-NAACL*, 2004.

[62] M. Rahim, G. D. Fabbrizio, M. W. C. Kamm, A. Pokrovsky, P. Ruscitti, E. Levin, S. Lee, A. Syrdal, and K. Schlosser. Voice-if : a mixed-initiative spoken dialogue system for at& t conference services. In *Proceedings of the 7th European Conference on Speech Processing (Eurospeech'01)*, Aalborg (Danmark), 2001.

[63] O. Rambow, S. Bangalore, and M. Walker. Natural language generation in dialog systems. In *Proceedings of the 1st International Conference on Human Language Technology Research (HLT'01)*, San Diego, USA, 2001.

[64] C. Raymond and G. Riccardi. Generative and discriminative algorithms for spoken language understanding. In *Proceedings of Interspeech*, Anvers (Belgium), August 2007.

[65] R. Reichman. *Plain-Speaking : A Theory and Grammar of Spontaneous Discourse*. PhD thesis, Department of Computer Science, Harvard University, Cambridge, Massashussetts, 1981.

[66] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, 2000.

[67] J. Schatzmann, B. Thomson, and S. Young. Error simulation for training statistical dialogue systems. In *Proceedings of the International Workshop on Automatic Speech Recognition and Understanding (ASRU'07)*, Kyoto (Japan).

[68] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2) :97–126, June 2006.

[69] K. Scheffler and S. Young. Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proceedings of NAACL Workshop on Adaptation in Dialogue Systems*, 2001.

[70] S. Seneff. Tina : A natural language system for spoken language applications. *Computaional Linguistics*, 18(1) :61–86, 1992.

[71] S. Shieber, G. van Noord, F. Pereira, and R. Moore. Semantic head-driven generation. *Computational Linguistics*, 16 :30–42, 1990.

[72] C. Sidner. *Computational Models of Discourse*, chapter Focusing in the Comprehension of Definite Anaphora, pages 267–330. MIT Press, Cambridge, Mass, 1983.

[73] S. Singh, M. Kearns, D. Litman, and M. Walker. Reinforcement learning for spoken dialogue systems. In *Proceedings of the Neural Information Processing Society Meeting (NIPS'99)*, Vancouver (Canada), 1999.

[74] R. Smith and R. Hipp. *Spoken Natural Language Dialog Systems : a Practical Approach*. Oxford University Press, New York, 1994.

[75] P. Sneele and J. Waals. Evaluation of a speech-driven telephone information service using the paradise framework : a closer look at subjective measures. In *Proceedings of the 8th European Conference on Speech Technology, (EuroSpeech'03)*, pages 1949–1952, Geneva (Switzerland, September 2003.

[76] R. Sutton and A. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 1998. ISBN : 0-262-19398-1.

[77] W3C. *VoiceXML 3.0 Specifications*, December 2008. http ://www.w3.org/TR/voicexml30/.

[78] M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard, and S. Whittaker. Darpa communicator dialog travel planning systems : The june 2000 data collection. In *Proceedings of the 7th European Conference on Speech Technology, (Eurospeech'01)*, Aalborg (Danmark), 2001.

[79] M. Walker, J. Boland, and C. Kamm. The utility of elapsed time as a usability metric for spoken dialogue systems. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'99)*, Keystone, CO (USA), 1999.

[80] M. Walker, J. Cahn, and S. Whittaker. Linguistic style improvisation for lifelike computer characters. In *Proceedings of the AAAI Workshop AI, Alife and Entertainment*, Portland, 1996.

[81] M. Walker, D. Hindle, J. Fromer, G. D. Fabbrizio, and C. Mestel. Evaluating competing agent strategies for a voice email agent. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech'97)*, Rhodes (Greece), 1997.

[82] M. Walker, L. Hirschman, and J. Aberdeen. Evaluation for darpa communicator spoken dialogue systems. In *Proceedings of the Language Resources and Evaluation Conference (LREC'00)*, 2000.

[83] M. Walker, A. Joshi, and E. Prince, editors. *Centering Theory in Discourse*. Oxford University Press, 1998.

[84] M. Walker, D. Litman, C. Kamm, and A. Abella. Paradise : A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 271–280, Madrid (Spain), 1997.

[85] M. Walker, O. Rambow, and M. Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language Special Issue on Spoken Language Generation*, 16(3-4) :409–433, 2002.

[86] P. P. J. Williams and S. Young. Partially observable markov decision processes with continuous observations for dialogue management. In *Proceedings of the SigDial Workshop (SigDial'06)*, 2005.

[87] W. Woods. Transition network grammars for natural language analysis. *Communications of the ACM*, 13 :591–606, 1970.

[88] S. Young. Using pomdps for dialog management. In *Proceedings of the 1st IEEE/ACL Workshop on Spoken Language Technologies (SLT'06)*, 2006.