Context-aware Speech Recognition in a Robot Navigation Scenario

Martin Hacker

Embedded Systems Initiative (ESI), Univ. of Erlangen-Nuremberg
Department Computer Science 8 (Artificial Intelligence)
Haberstr. 2, 91058 Erlangen, Germany
martin.hacker@cs.fau.de
http://www8.cs.fau.de

Abstract. Automatic speech recognition lacks the ability to integrate contextual knowledge into its optimization process – a resource that human speech perception makes extensive use of. We discuss shortcomings of current approaches to solve this problem, formalize the problem of context-aware speech recognition and understanding and introduce a robot navigation game that can be used to demonstrate and evaluate the impact of context on speech processing.

Keywords: speech recognition, speech understanding, context, robot navigation

1 Introduction

For the Artificial Intelligence community, Automatic Speech Recognition (ASR) has been a very challenging task for several decades. Although substantial progress has been made in both acoustic modeling and modeling of language use, systems are still clearly outperformed by humans when employed under real-life conditions. The main reason for this performance gap is that speech itself is highly ambiguous and susceptible to noise and external audio events. What is actually perceived by a human is already an interpretation of the ambiguous signal, highly depending on expectations what the speaker could say in the current situation and on associations of the listener emanating from current thoughts and perception of the environment.

The same holds for the interpretation of a correctly perceived utterance that is often ambiguous in spontaneous speech, but is not perceived as ambiguous as long as the listener can infer the correct meaning in the current situation.

The circumstances of this type as a whole that influence speech perception and understanding are commonly referred to as the (relevant) context of an utterance. The concept of context, however, is often either used in a vague way or simplified in an ad-hoc manner. The result is that the influence of context is not adequately integrated into state-of-the-art ASR systems, which is presumably one of the main reasons for the above mentioned performance gap.

In this article, we discuss the shortcomings of current approaches by working out the role of context in a robot navigation scenario. The remainder of the article is structured as follows: After reviewing approaches to model contextual influence in ASR, we formalize the problem of context-aware speech recognition and understanding. Then we introduce the application scenario and our data collection and evaluate the benefits of context-aware speech recognition using a simple context model. We conclude with a discussion of challenges of context modeling using selected examples and an outlook on open issues and future work.

2 Related Work

2.1 Definitions of Context

A frequently used definition of context was given by Dey [3]:

> Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Although this definition evokes associations with physical objects, it can be extended to cover also mental concepts such as dialogues. A situation is a concrete assignment of contextual variables, i. e. a set of attribute-value pairs that describe the relevant properties of an entity at a certain time.

Commonly, researchers distinguish between three types of context that are relevant for speech processing (see [7], e. g.):

- *Discourse context:* topic of the conversation, linguistic information from preceding utterances referred to by the current utterance (ellipses, anaphora, answering a question);
- *Physical context:* device states or physical sensor data including time and location characterizing the current environmental and in-domain objects;
- *Logical context:* interpreted sensor data accumulated over multiple sensors and/or over time (e. g. activity recognition), conclusions drawn from user utterances or expected impact of system actions.

2.2 Context in Dialogue

A traditional method to account for *discourse context* is to use separate language models for pre-defined discourse situations. This approach is straight-forward for simple applications which engage finite-state-based dialogue models. For example, in a system-initiated timetable information dialogue where the system step-by-step fills necessary slots for the database query, the vocabulary can be divided into question-relevant parts to decrease the perplexity of the language model.

The construction of such context-dependent language models can be automated if sufficient annotated training data is available. Among others, Bod [1] used discourse context to foster both speech recognition and understanding in a public transport information system. The corpus-based parsing model called data-oriented parsing (DOP) was made context-dependent by exploiting only training utterances made in the same discourse context as the current utterance. The method was shown to slightly increase system accuracy, but it requires syntactically and semantically annotated training corpora for the respective application domain. As discourse context, only the preceding system utterance was used. Thus, applying the method to applications providing more fine-grained sets of system utterances or integrating richer discourse context would result in sparse data problems due to over-fragmentation of the training corpus.

The general principle of using separate language models can easily be transferred to account for different environment situations derived from *physical context*. Everitt et al. [4] augmented speech recognition in a fitness room with physical sensor information about which gym object is currently being used by the speaker. The recognition grammar was split up into object-related grammars from which a single one is selected depending on the sensor information. The results are encouraging although the method proved to be susceptible to sensor errors. However, it remains unclear how the derivation of sub-grammars can be performed when more than one context variable is available.

Leong et al. [7] present a context-sensitive approach for controlling devices in an intelligent meeting room. As an intermediate step before deep parsing, ASR n-best lists are re-ranked by shallow analysis of user goals and their contextual coherence. Bayesian networks are used to estimate the probability of a user goal given the words in the utterance and contextual attributes. By using a Bayesian network structure, system designers can directly integrate knowledge on dependencies between contextual variables and linguistic entities to decrease the number of parameters to be learned. Their results show a significant improvement in disambiguating ASR output while the system is also enabled to process deictic references and elliptical utterances.

A potential shortcoming of the model used is that it is based on a bag-of-words language model. Although the approach is general enough for richer linguistic information to be included, it is unclear how deep linguistic structures might fit into the Bayesian network architecture, assuming that a deep understanding of such structures is necessary to model complex linguistic context phenomena. What is more, the approach requires to pre-define a set of all possible system actions. This is feasible for a room with a limited set of on-off devices as in the meeting room scenario. It would, however, not scale up to multiple parameterizable commands or even problem-solving dialogues. Neither is it feasible to contruct Bayesian networks that account for all possible user goals in such applications nor would it be possible to collect sufficient training data.

Summing up, all state-of-the art approaches either require training data to learn statistical language models or hand-crafted grammar solutions. Thus incorporating rich context for complex applications is not feasible, which underlines the need to integrate explicit world and domain knowledge about causal relations between entities in the environment and conversation. Logic seems to be an appropriate means to model this knowledge. Using logic, however, it is difficult to cope with uncertainty, ambiguity and vagueness which are unavoidable in real applications dealing with sensor data and spontaneous speech.

3 Formalizing the problem

3.1 Context

As mentioned in section 2.1, contextual information is commonly partitioned into discourse, physical and logical context. The distinction between physical and logical context, however, seems to be artificial since real physical sensor data also needs to be interpreted. The distinction between discourse-related and other context is based on the fact that dialogue participants can address entities on various levels:

- The discourse level, e. g. by directly or indirectly referring to a linguistic concept or a discourse element such as an utterance (e. g. "What did you say?").
- The in-domain level, i. e. talking about entities addressed by the application.
- The out-of-domain level. Entities outside the application and discourse domain are usually not referred to by utterances. However, they can have an influence on what is said and how it is said.

Instead, we propose a distinction according to the stage of speech production that is influenced by the contextual information, as this is more appropriate to modeling contextual influence:

- *Motivational context:* Information that influences *what* people want to express (e. g. the fact that the heating is running and the temperature is high would cause the user to instruct the system to switch off the heating).
- *Linguistic context:* Information that influences *how* people *phrase* what they want to express (e. g. discourse constituents already introduced enable the speaker to use anaphoric references).
- *Articulatory context:* Information that influences *how* people *articulate* the utterance (e. g. noise level or emotional state).
- *Acoustic context:* Simultaneous acoustic events and conditions that influence the speech signal as perceived by the system.

3.2 Speech Recognition

The problem of speech recognition is commonly defined as follows:

- Assume that the audio signal spans the time inverval from the beginning to the end of the utterance of interest (this is done by segmentation).
- From the audio signal, a sequence of feature vectors $X = x_0, \ldots, x_{n-1}$ has been extracted.

- Find the word chain $\hat{w} \in W^*$ that correctly transcribes the utterance:

$$\hat{w} = \underset{w \in W^*}{\text{argmax}} P(w|X) = \underset{w \in W^*}{\text{argmax}} \frac{P(X|w) \cdot P(w)}{P(X)} \tag{1}$$

- The denominator $P(X)$ is independent of w and thus can be ignored by the maximization process. Hence the best solution is the transcription that is the best trade-off between the acoustic model $P(X|w)$ and the language model $P(w)$.

3.3 Context-dependent Speech Recognition

Equation 1 describes which is the most probable sequence of words when nothing is known about the utterance but the acoustic observation X. However, if the context C of the utterance is known, we must reformulate the problem as follows:

$$\hat{w} = \underset{w \in W^*}{\text{argmax}} P(w|X, C) = \underset{w \in W^*}{\text{argmax}} \frac{P(X|w, C) \cdot P(w|C)}{P(X|C)} \tag{2}$$

$$= \underset{w \in W^*}{\text{argmax}} \frac{P(X|w, C) \cdot P(C|w) \cdot P(w)}{P(X, C)} \tag{3}$$

The *context-dependent acoustic model* $P(X|w, C)$ can be substituted by the traditional acoustic model $P(X|w)$ if we assume that the pronunciation of a given word chain is independent of any contextual factor. Apparently, this is not the case as, for instance, the emotional state of the speaker, the level of background noise and the distance from the listener have an influence on the audio signal received by the listener (articulatory and acoustic context). A viable solution to this problem would be to approximate $P(X|w, C)$ by a set of acoustic models $P_d(X|w)$ with d being disjunct classes of situations. In practice, this would mean that a set of different speech recognizers – e. g. for different emotional states – is available from which the one is chosen that best fits the current context. Instead of choosing one speech recognizer, multiple speech recognizers could be combined by linear combination to allow for arbitrary states between the prototypes.

The *context-dependent language model* $P(w|C)$ can be modeled in two ways. The one given in equation 2 is used by most approaches and uses different language models $P_d(w)$ for different classes d of situations. Again, it would be possible to engage a linear combination, but this seems to be quite unintuitive for language models, as they indicate *what* can be said and not *how* it can be said.

An alternative model is given in equation 3: We keep the general language model $P(w)$ and multiply it by the posterior probability $P(C|w)$. But how to estimate this distribution? We can easily find the values for C where $P(C|w)$ has a very small value or is even null. They correspond to situations that would not be considered as possible when an user utters w, particularly situations that are inconsistent with the meaning of the utterance. But it is difficult to estimate $P(C|w)$ for contexts that are considered as possible given w.

A simplifying solution would be to assign zero probabilities for contexts that are inconsistent and a uniform distribution for all other contexts. This would be equivalent to the commonly used approach to calculate the n-best lists with a context-independent speech recognizer in a first step and, in a second step, to remove n-best results that turn out to be inconsistent with the context after parsing.

3.4 Speech Understanding

Up to now, the task was to find the most probable sequence of words. This is sufficient for a dictation task. When the system, however, needs to understand the user utterance, it is rather interested in the user's goal than in its wording. Two or more word chains may result from the same user goal at the pragmatical level. To solve the problem of *speech understanding*, hence, we need to consider all possible user goals g and sum up the pobabilities of all possible wordings[1]:

$$\begin{aligned}\hat{g} &= \underset{g\in G}{\operatorname{argmax}} P(g|X,C) = \underset{g\in G}{\operatorname{argmax}} \sum_{w\in W^*} P(w|X,C)P(g|w,C) = \\ &= \underset{g\in G}{\operatorname{argmax}} \frac{1}{P(X|C)} \cdot \sum_{w\in W^*} P(X|w,C)\cdot P(w|C)\cdot P(g|w,C) = \qquad (4) \\ &= \underset{g\in G}{\operatorname{argmax}} \frac{P(g|C_M)}{P(X|C)} \cdot \sum_{w\in W^*} P(X|w,C)\cdot P(w|g,C_L) \qquad (5)\end{aligned}$$

Solving this formula for the combined speech recognition and understanding problem is computationally expensive as the goal must be computed for every word chain. To overcome this difficulty, the problem usually is divided into two sequential problems:

1. At first, the n best word chains are computed.
2. Then the goals are evaluated for the n best word chains. This means that the sum for all word chains in equation 4 is approximated by the sum for the n best word chains.

Now, as the problem becomes *computationally* manageable, the question is how the probability $P(g|w,C)$ can be *modeled.* Learning it from examples as done in [7][2] is even more data-intensive than learning $P(w|C)$ (which was discussed above) and therefore not feasible for rich context models and complex goals.

The sparse data problem can be alleviated (Equation 5) by transforming the context-dependent language model and reducing the corresponding contexts to the relevant classes as defined in section 3.1: For the user goal, only the motivational context C_M and for the wording, only the linguistic context C_L is

[1] Hereafter we assume that X and g are independent for a given w, i.e. every speech recognition hypothesis w includes the word chain including all annotations of features such as prosody that might depend on the current user goal.

[2] The authors transform the formula and learn $P(w,C|g)$, but omit the summation.

considered as relevant. The goal prior probability $P(g|C_M)$ can be modeled using knowledge about preconditions and effects of system actions as well as user motivations. The context- and goal-dependent language model can be approximated by $P(w)$ if g can be derived from w by parsing and by anaphora and ellipsis resolution using C_L and is set to 0 otherwise.

4 Experimental Work

4.1 Robertino-game: A Simulated Robot Navigation Task

Simulated robots are commonly used for AI research [6], as they are inexpensive and easy to maintain. To construct naturalistic conditions, sensor data must be simulated by noise and filters, and information must be hidden for the user to constrain her knowledge of the situation to that part she would be able to perceive in real environments. The robot can derive the current context from the simulated sensor data. In our first experiments, we assume that the user has full access to the context and the robot can reliably sense its direct environment. The data can though be used to perform future investigations with the robot using simulated sensors or further context such as maps that it constructs progressively.

Another difference engineers are faced with when simulating a robot is that continuous movements must be replaced by a series of steps. For instance, a rotation caused by a motor running for n frames with a certain power P would be replaced by n discrete rotation steps with an increment depending on P.

The scenario used for our experiments is a game where one human player navigates the robot through a fictious maze-like environment (see Fig. 1) to a pre-defined goal by only using speech commands. The environment consists of walls, bombs, substitute rockets and colored areas painted on the floor. The robot will explode when it runs into a bomb or a wall. It can perform the following behaviours:

- *Turning:* The user can roughly specify the degree of rotation with linguistic hedges (e. g. *slightly to the left*). Turning while the robot is driving will cause the robot drive a curve.
- *Driving:* The user can instruct the robot to drive in a certain direction relative to its orientation, without changing the orientation.
- *Shooting:* A rocket can be fired in the direction of the current orientation. This can be used to destroy bombs that are in the way. Three rockets are available after the game has been started. When a substitute rocket is passed by the robot, it is automatically loaded.

4.2 Data Collection

The data has been recorded in two sessions. The participants that played the game were mainly high school students and university staff members. In the first session, a close-distance speech microphone was used, while in the second

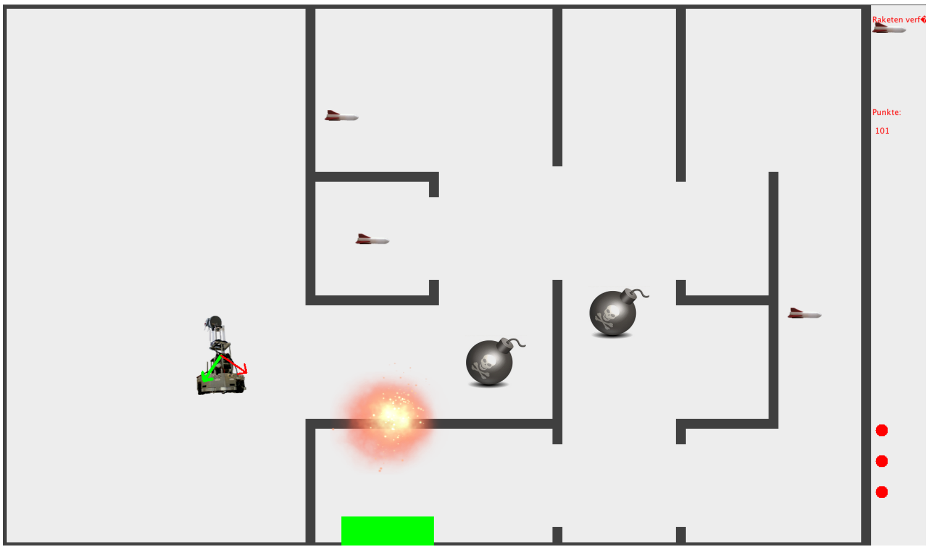

Fig. 1. Screenshot of the Robertino-game with the green arrow indicating the robot's translational direction and velocity and the red arrow indicating its orientation. The explosion was caused by a rocket hitting the wall.

session the integrated microphone of a MacBook Pro was used as a far-distance microphone. The participants attended the sessions in groups of 7–15 people, which caused background talking and laughing that makes speech recognition more challenging. In the second session, there was also continuing background noise from a construction site outside of the building.

The data includes the audio signal of each on-talk player utterance and its reference transcription, as well as all robot actions. The reference transcription was conducted by the presenter who attended the sessions. The utterances were also annotated with the presumed user goal. The robot actions are reactions to user utterances as they were understood by the system. Half of the games were conducted following the Wizard of Oz method. In this part of the data, the robot actions are reactions to user utterances as they were understood by the wizard, which is very close to the reference annotation.

4.3 Speech understanding

Table 1 shows a list of basic system actions that can be executed in every frame. These basic actions are highly implementation-dependent. For the user, however, it is irrelevant which robot actions will be executed in which frames. She would articulate a more abstract descripition of the intended robot behaviour that is independent from implementation details. Following common naming conventions, we will refer to these described behaviours as *user goals*.

Table 1. Basic system actions

Command	Parameter
TURN_STEP_LEFT	-
TURN_STEP_RIGHT	-
START_RUNNING	-
STOP_RUNNING	-
CHANGE_DIRECTION	degree
CHANGE_VELOCITY_RELATIVE	velocity_diff
CHANGE_VELOCITY_ABSOLUTE	velocity
SHOOT	-

A list of possible user goals is shown in Table 2, annotated by its meaning which is constructed by a mapping from utterances to sets of basic user actions. The mapping can be realized by a simple *keyword spotting* and *slot filling* approach, which is sufficient for most simple applications and therefore implemented in most commercial speech-based systems. However, when the application is more complex and the meaning of words and phrases depends on their syntactic function within the sentence, it is necessary to employ more advanced parsing and speech understanding techniques.

Table 2. Selection of basic user goals. The meaning representations contain functions that schedule basic system actions by mapping them onto frames[4].

Wording	Meaning
shoot	action(SHOOT, cur)
faster/slower	action(CHANGE_VELOCITY_RELATIVE(x), cur)
fast/slow	action(CHANGE_VELOCITY_ABSOLUTE(x), cur)
left	action(CHANGE_DIRECTION(-90), cur)
go left	action(CHANGE_DIRECTION(-90), cur), action(START_RUNNING, cur)
turn left	process(cur, cur + x, action(TURN_STEP_LEFT, t))
stop	action(STOP_RUNNING, cur), suspend(process(before(cur),after(cur),*), cur)
go left to the red area	action(CHANGE_DIRECTION(-90), cur), complex_action(action(START_RUNNING, cur), action(STOP_RUNNING, color_sensor(red))))
go back	process(cur, cur + cur, undo(action(*, cur + cur - t))
no	undo(last_action)

4.4 Evaluation of context awareness

A part of the recorded data containing 199 utterances was used to investigate the potential benefit of context-aware speech recognition. Each utterance was processed by a speech recognizer producing up to 5 best hypotheses[5]. We used a simple context model (see Table 3) to describe the game situation at the time when the utterance was processed by the system[6].

Table 3. A simple context model for a situation in the robot navigation domain

Type	Variable	Indication
system state	driving turning direction numRockets	The robot is moving. The robot is in the process of turning. the robot's current translational direction the number of available rockets

sensoric context	obstacleTowards(x)
	bomb(x)
	obstacleCC
		The sonors sense an nearby obstacle in dir. x.
		A bomb is in the line of fire in direction x.
		The robot approaches an obstacle.

discourse context	lastUserGoal	last goal uttered by the user

Following the terminology of section 3.1, the context model contains motivational context. To derive the possible user goals in a particular situation, we built a very simple motivational user model with the following rules:

Expected user goals:

- When an obstacle is coming closer, the user probably wants to *stop* the robot.
- When a bomb is in the line of fire in front of the robot and there are still rockets available, the user will probably tell the robot to *shoot.*
- When a bomb is in the line of fire in direction d and there are still rockets available, the user will probably tell the robot to *turn towards d.*

Implausible user goals:

- When an obstacle is in direction d, the user would not want the machine to *move towards d.*
- When the robot is neither driving nor turning and the last goal[7] was not *stop*, the user would not tell the machine to *stop.*

[5] There might be less than 5 hypotheses for short utterances when the speech recognizer cannot find more hypotheses with acoustic confidence above a certain threshold relative to the best hypothesis.

[6] In fact, the context is dynamic and can change during a user utterance. We used a fixed representation of the context immediately *after* the utterance was spoken. This was due to the fact that the users adapted to the system's latency by anticipating situations and starting to speak before the command was applicable.

- When the robot is driving and the last goal[7] was not *start moving*, the user would not tell the machine to *start moving*.
- When the robot is driving towards direction d and the last goal[7] was not *move towards d*, the user would not tell the machine to *move towards d*.

For almost 20% of the utterances, the underlying user goal is expected and can therefore be augmented by the system purely based on the context, i.e. without using a speech recognizer. For a few utterances (3%), the expressed user goal is implausible given the situation. The latter can be explained by the fact that user behavior is not always reasonable. In most of the present cases, the user was confused by the relative orientation of the robot and accidentally said *right* instead of *left* (and vice versa).

Table 4. Contextual coherence of hypotheses

	reference transcriptions	1-best hypothesis		5 best hypotheses	
		true	false	true	false
overall	199	123	76	293	559
expected	36 (18,1%)	26 (21,1%)	1 (1,3%)	58 (19,8%)	11 (2,0%)
implausible	6 (3,0%)	6 (4,9%)	5 (6,6%)	6 (2,0%)	30 (5,4%)

Table 4 shows how many hypotheses are considered as expected or implausible by the user model. The results show that context can play a distinctive role in deciding whether an (acoustically plausible) hypothesis is true or false. To evaluate the actual benefit of context-awareness for the ASR module, we applied a simple hypothesis re-ranking strategy (for re-ranking cf. [9] e.g.) based on the two features *expected* and *implausible*:

```
if n-best-list contains expected utterances
    choose the best-ranked one
elseif n-best-list contains utterances that are not implausible
    choose the best-ranked one
else
    reject utterance
```

When this strategy is applied, 9,2% of the false first hypotheses are corrected, and additional 5,3% are rejected. However, 3,3% of the correct first hypotheses are replaced by a false one, and additional 4,1% are rejected. Most false rejections and false corrections though correspond to utterances where the expressed user goal differs from the intended user goal. This includes the above mentioned user mistakes.

[7] Some rules were extended by a condition regarding the last user goal. This was necessary because of the latency of the system and the delayed reaction of the user: Sometimes the user repeated the command immediately after or at the same time as it was executed by the system.

The figures seem to suggest that contextual information only slightly increases ASR accuracy. We though think that the results are encouraging because both contextual model and motivational user model used for the evaluation were very simple. Only a few of the possible user goals were investigated with respect to their contextual coherence. We are convinced that a richer context model and an advanced user model that allows deep reasoning over multiple dialogue turns would have the potential to noticeably improve ASR performance.

Moreover, the classification and re-ranking algorithms we applied are very basic. Allowing for gradual values of contextual coherence and integrating features from other knowledge sources like acoustics and syntax would enable us to use powerful machine learning techniques for error correction (cf. [8]).

5 Challenges of Context Modeling for Speech Understanding

In the robot navigation scenario, the user is allowed to give instructions for future behavior depending on conditions, e. g. by saying

Example 1. move slowly left to the beginning of the red area and then turn right and follow the wall.

When such combinations of user goals are communicated, the system is faced with the following problems:

- *Persistence of user goals:* For some goals, it is unclear how long they are valid unless they are replaced by a new incompatible goal. In Example 1, it is unclear whether the system should continue going slowly when following the wall. The problem is also present when simple user goals are communicated in a series of utterances:

  Example 2. Move forward until you reach the red area.
  Stop! *(before the robot has reached the red area)*
  Continue moving.
  → *Should the system still stop at the red area as said before the stop command?*

- *Future context:* When calculating the contextual coherence of a future user goal, the system must take the anticipated future context into account rather than the current context. The future context depends on the effects of the behavior performed up to the time the goal of interest becomes active. This time, however, is unknown because it depends on some (sensory) conditions

[8] The frame immediately following after the understanding process is complete is denoted by *cur*. Frames can also be declared by conditions, which denotes the first future frame with the condition being fulfilled.

that cannot be evaluated before a future state is reached. Assuming that the user utterance can reliably be segmented into goal-specific parts, it would be possible to simplify the problem by evaluating parts not before they become active. However, when we consider why the user uttered a series of goals at once rather than waiting for the preceding actions to be completed, we must admit that this often happens because the user considers the future goals as relevant for the interpretation of the preceding goals. In other words, the announced future actions are part of the relevant context for the understanding of the preceding actions.
These considerations scratch the borderline between speech understanding and planning, two issues that are commonly treated as sequential but in fact are highly interactive.

– *Partial knowledge:* Speech input can be viewed as a sensor itself, with either the speaker giving information about the context directly (a) or the system deriving such information from the utterance (b).

  Example 3. (a) There's a wall to the left.
  (b) Follow the wall. *(→ there must be a wall)*

– *Subjectivity:* User and system can have different beliefs about contextual entities unless both have acknowledged the fact to be part of the common ground [2]. When the system uses context for speech understanding to find motivational evidence for an utterance, it must use the user's subjective context instead of its own. This means that the system needs to keep track of hypotheses about the user's beliefs.
– *Uncertainty:* The system's beliefs about contextual entities are partially derived from sensor data and therefore might be wrong. When wrong context is used for speech understanding, the correct interpretation of the user goal might be rejected due to missing contextual coherence. A major challenge is to design systems that are aware of potential misrecognitions and allow to update their context model in such situations. These are the possible reasons when the system cannot find enough acoustic and contextual evidence for any hypothesis:
  1. It was not possible to understand the utterance (e. g. out of domain or superimposed by other acoustic events).
  2. The system's beliefs about the (user) context are wrong and need to be updated using information from the acoustically evident, but contextually incoherent hypothesis.
  3. The user's beliefs about the context are wrong and differ from the system's beliefs about the user context. Hence it is up to the system to clarify the facts in a dialogue with the user.

6 Summary and Future Work

In this paper, we formalized the problem of context-aware speech understanding and introduced a speech-controlled robot navigation game. For this application, we demonstrated how speech recognition can benefit from contextual information and discussed challenges of context modeling using selected examples.

The extension of the application by a context model is still ongoing work. Future work will focus on building rich contextual and motivational models and integrating the contextual knowledge into speech recognition. In a second step, we will compare the context-aware system to human speech perception using the method described in [5].

References

1. Bod, R.: Context-sensitive spoken dialogue processing with the DOP model. Nat. Lang. Eng. 5, 309–323 (December 1999)
2. Clark, H.H., Brennan, S.E.: Grounding in communication. In: Resnick, L.B., Levine, J.M., Teasley, S.D. (eds.) Perspectives on socially shared cognition, pp. 127–149. American Psychological Association, Washington, DC (1991)
3. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. 5, 4–7 (January 2001)
4. Everitt, K.M., Harada, S., Bilmes, J., Landay, J.A.: Disambiguating speech commands using physical context. In: Proceedings of the 9th International Conference on Multimodal Interfaces. pp. 247–254. ICMI '07, ACM, New York, NY, USA (2007)
5. Hacker, M., Elsweiler, D., Ludwig, B.: Investigating human speech processing as a model for spoken dialogue systems: An experimental framework. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) Proceeding of the 19th European Conference on Artificial Intelligence (ECAI 2010). Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 1137–1138. IOS Press, Amsterdam, The Netherlands, The Netherlands (2010)
6. Hugues, L., Bredeche, N., Futurs, T.I.: Simbad: An autonomous robot simulation package for education and research. In: Proceedings of The Ninth International Conference on the Simulation of Adaptive Behavior (SAB'06). Roma, Italy - Springer's Lecture Notes in Computer Sciences / Artificial Intelligence series (LNCS/LNAI) n. pp. 831–842 (2006)
7. Leong, L.e.H., Kobayashi, S., Koshizuka, N., Sakamura, K.: CASIS: a context-aware speech interface system. In: Proceedings of the 10th International Conference on Intelligent User Interfaces. pp. 231–238. IUI '05, ACM, New York, NY, USA (2005)
8. Skantze, G., Edlund, J.: Early error detection on word level. In: Proceedings of ITRW on Robustness Issues in Conversational Interaction. Norvich, UK (2004)
9. Wai, C., Pieraccini, R., Meng, H.: A dynamic semantic model for re-scoring recognition hypotheses. Acoustics, Speech, and Signal Processing, IEEE International Conference on 1, 589–592 (2001)