Making grounding decisions: Data-driven estimation of dialogue costs and confidence thresholds

Gabriel Skantze
Centre for Speech Technology, KTH
gabriel@speech.kth.se

Abstract

This paper presents a data-driven decision-theoretic approach to making grounding decisions in spoken dialogue systems, i.e., to decide which recognition hypotheses to consider as correct and which grounding action to take. Based on task analysis of the dialogue domain, cost functions are derived, which take dialogue efficiency, consequence of task failure and information gain into account. Dialogue data is then used to estimate speech recognition confidence thresholds that are dependent on the dialogue context.

1 Making grounding decisions

One of the greatest challenges when building spoken dialogue systems is to deal with *uncertainty*, which stems partly from the ambiguity of language itself and partly from the error-prone speech recognition process. Speakers handle uncertainty in dialogue by exchanging positive and negative evidence of understanding, a process commonly referred to as *grounding* (Clark, 1996). In order for a speaker to evaluate the addressee's understanding of the concepts involved, two kinds of evidence are especially useful: display of understanding (DISPLAY) and clarification requests (CLARIFY). The addressee can also choose not to provide any direct evidence, but to simply ACCEPT or REJECT the hypothesis. These options are exemplified in the following alternative system responses in a human-computer dialogue, where the system is uncertain about *the red building*:

(1) U: I can see a red building.
S (ACCEPT): *Ok, can you see a tree in front of you?*
S (DISPLAY): *Ok, a red building, can you see a tree in front of you?*
S (CLARIFY): *A red building?*
S (REJECT): *What did you say?*

In many dialogue systems, CLARIFY and DISPLAY are referred to as *explicit* and *implicit* verification.

The problem addressed in this paper can be described as follows: Given these different grounding options, how should a dialogue system choose what kind of evidence to give and which hypotheses to accept and reject? We will refer to this as the *grounding decision problem*. There are at least three important factors that speakers may take into account when making this decision:

1. Level of uncertainty
2. Task-related costs and utility
3. Cost of grounding actions

First, the more uncertain listeners are, the more evidence they provide. Second, as less evidence is given, the risk that a misunderstanding occurs will increase – thereby jeopardizing the task the speakers may be involved in. However, the cost of such a misunderstanding depends on the task at hand. Third, it would not be efficient to always display understanding or clarify everything that is said. Sometimes it may be more efficient to risk a misunderstanding and take the consequences.

A common approach to grounding decisions is to compare the speech recognition confidence score against a set of hand-crafted thresholds, and choose ACCEPT when the confidence is high, DISPLAY for middle-high scores, CLARIFY for middle-low scores and REJECT for low scores (see for example Bouwman et al., 1999). However, in this simple account, only Factor 1 above (level of uncertainty) is considered, and the thresholds used are typically only based on intuition.

In order to take Factor 2 (task-related costs and utility) into account, Bohus & Rudnicky (2001) uses a data-driven technique to derive actual costs from dialogue data, which showed that false acceptances were more costly than false rejections. Another aspect is that task costs are dynamic and often depend on the current state of the dialogue. To incorporate this aspect, Bohus & Rudnicky (2005) presents a method where binary logistic regression is used to determine the costs (in terms of task success) of various types of understanding errors involved in the rejection trade-off. Different regressions may then be calculated in different dialogue states, resulting in dynamic thresholds. However, these methods do not consider other grounding actions than ACCEPT

and REJECT. To do this, Factor 3 above (cost of grounding actions) must also be considered.

Paek & Horvitz (2003) presents a decision theoretic approach to the grounding decision problem, based on the framework of *decision making under uncertainty*. According to this proposal, the optimal grounding action *GA* should satisfy the Principle of Maximum Expected Utility (MEU), which can be defined as follows: *Choose an action a, so that the expected utility EU(a) is maximized*. When making this decision, the world may be in one of the states h_1, h_2, $h_3 \ldots h_n$, and this state may have an impact on the effect of the action taken. This effect can be described by the function $Utility(a,h_i)$, which is the utility for action a under state h_i. Thus, for each action a, the probability for each possible state and the utility for taking action a, given that state, should be summed up:

$$(2) \quad GA = \arg\max_a EU(a) = \arg\max_a \sum_{i=1}^{n} P(h_i) \times Utility(a, h_i)$$

This approach is promising, in that it may account for all decision factors listed above. However, in Paek & Horvitz (2003), the utilities used in the model were estimated directly by the user (via a GUI) and were not derived from data.

2 The proposed model

In this paper, we will show how the utilities may be estimated directly from collected dialogue data. To do this, the problem will be described as that of minimising costs: *Choose a grounding action a, so that the sum of all task-related costs and grounding costs is minimised, considering the probability that the recognition hypothesis is correct.* Thus, the world may be in two states (*correct* and *incorrect* recognition), and a probability measure for these states is needed, as well as a cost function for calculating the costs of the different grounding actions, given these states. The problem is expressed in the following equation (where *P(incorrect)* equals 1-*P(correct)*):

$$(3) \quad GA = \arg\min_a \begin{pmatrix} P(correct) \times Cost(a, correct) + \\ P(incorrect) \times Cost(a, incorrect) \end{pmatrix}$$

To select the optimal grounding action according to equation (3), a probability measure of the state *correct* is needed, as well as a cost function for calculating the costs of the different grounding actions, given these states.

In this paper, we will assume that *P(correct)* can be derived from the speech recognition confidence score. While confidence scores typically delivered by speech recognisers should not be used as a direct measure of probability, it should be possible to derive probabilistic scores (Jiang, 2005).

3 Data

The model presented in this paper will be applied to data collected using the HIGGINS spoken dialogue system developed at KTH (Edlund et al., 2004). The initial domain for the system developed within the project is pedestrian city navigation and guiding. A user gives the system a destination and the system guides the user by giving verbal instructions. The system does not have access to the user's position. Instead, it has to figure out the position based on the user's descriptions of the surroundings. Since the user is moving, the system continually has to update its model of the user's position and provide new, possibly amended instructions until the destination is reached. For simulation, a 3D model of a virtual city is used. Example (1) above is typical for this domain. A typical dialogue consists of three main *phases* or *sub-tasks*: a *goal assertion phase*, a *positioning phase*, and a *guiding phase*.

A version of the HIGGINS system, with different sets of handcrafted confidence thresholds for making grounding decisions, was evaluated with users. The evaluation involved 16 participants, all native speakers of Swedish. The collected data consists of 2007 user utterances. A more detailed description of the data collection is provided in Skantze (in press).

4 Cost measure and functions

The model presented in this paper relies on a unified cost measure, which may be used for estimating both the task-related costs and the cost of grounding actions. The ultimate measure of cost would be the reduction of user satisfaction. However, user satisfaction is practically only obtainable on the dialogue level, and we need a much more detailed analysis. A cost measure that is relevant for both grounding actions and the task, and that is obtainable on all levels of analysis, is *efficiency*. This is reflected in the *principle of least effort* (Clark, 1996): "All things being equal, agents try to minimize their effort in doing what they intend to do". Thus, efficiency and user satisfaction should correlate to some degree, at least in a task-oriented dialogue setting as the one used in this paper. In the data collected here, the best predictor for user satisfaction was the *total number of syllables* uttered (from both the user and the system) ($R^2 = 0.622$). The impact of efficiency on user satisfaction in task-oriented dialogue has also been reported in other studies, such as Bouwman & Hulstijn (1998).

Using efficiency as a cost measure, we will analyse the consequences of different actions, given the correctness of the recognition hypothesis. The actions that will be considered are the ones listed in example (1): ACCEPT, DISPLAY, CLARIFY and REJECT. Table 1 summarises these costs based on a set of parameters, which are all average estimations over a set of dialogues.

Table 1: Costs for different grounding actions, given the correctness of the recognition (COR=Correct, INC=Incorrect).

Action,Hyp	Costs
ACCEPT,COR	No cost
ACCEPT,INC	The number of extra syllables the misunderstanding adds to the dialogue (*SylMis*).
DISPLAY,COR	Grounding dialogue *(SylDispCor)*.
DISPLAY,INC	Grounding dialogue (*SylDispInc)*. Risk that the user does not correct the system *(P(Fail|Disp,Inc))* times the consequences of a misunderstanding (*SylMis*).
CLARIFY,COR	Grounding dialogue *(SylClarCor)*. Risk that the user does not confirm the system *(P(Fail|Clar,Cor))* times the syllables for recovering the rejected concept *(SylRec)*.
CLARIFY,INC	Grounding dialogue *(SylClarInc)*
REJECT,COR	The number of syllables it takes to receive new information of the same value as the rejected concept *(SylRec)*.
REJECT,INC	No cost

The costs for DISPLAY and CLARIFY may need some explanation. In HIGGINS, a concept that is displayed is treated as correct unless the user initiates a repair. A concept that is clarified is treated as incorrect unless the user confirms it. Thus, they can be said to *fail* if the user does not correct a displayed misunderstanding or confirm a clarification of a correct concept. The number of syllables an average grounding dialogue takes involves both the grounding act and possible responses. For example, the following clarification dialogue involves 2 syllables (*SylClarCor*):

(4) S: *Red?*
U: Yes

Using these costs and equation (3) above, cost function may be defined for the different actions, as shown in Table 2.

Table 2: Cost functions for different grounding actions.

Action	Expected cost
ACCEPT	*P(incorrect)* x *SylMis*
DISPLAY	*P(correct)* x *SylDispCor* + *P(incorrect)* x *(SylDispInc* + *P(Fail|Disp,Inc)* x *SylMis)*
CLARIFY	*P(correct)* x *(SylClarCor* + *P(Fail|Clar,Cor)* x *SylRec)* + *P(incorrect)* x *SylClarInc*
REJECT	*P(correct)* x *SylRec*

5 Parameter estimation from data

To show how these parameters may be estimated from data, we will make a task analysis specific for the navigation domain presented here. We will start with the positioning phase of the dialogue, i.e., when the user describes her position, as in example (1) above.

The parameter *SylRec* describes the number of syllables it will take to get the same amount of information after a concept has been rejected. This parameter is highly context dependent – it depends on how much information the hypothesised concept provides (its *information gain*), compared to the average concept. This proportion will be referred to as *ConValueH*. The system and the user spent on average 15.0 syllables per important concept [1] accepted by the system. We will refer to this as *SylCon*. Based on these two parameters, *SylRec* can be calculated as follows:

(5) $SylRec = SylCon \times ConValueH$

How can *ConValueH* be estimated for the positioning phase? The purpose of the positioning phase is to cut down the number of possible user locations. Thus, the value of a concept can be described as the proportion of the set of possible user locations that are cut down after accepting it, compared to the average concept. The proportion of possible locations that are reduced on average after a single concept is accepted can be estimated from data (*CutDownA*). The dialogue system can then use the domain database to calculate the proportion of possible locations that would be cut down if the hypothesised concept would be accepted (*CutDownH*). By accepting *ConValueH* number of average concepts, each leaving a proportion of 1 - *CutDownA* possible locations, a proportion of 1 - *CutDownH* locations should be left. This is expressed in the following formula:

(6) $(1 - CutDownA)^{ConValueH} = (1 - CutDownH)$

By combining equations (5) and (6), *SylRec* can be calculated with the following formula:

(7) $$SylRec = SylCon \times \frac{\log(1 - CutDownH)}{\log(1 - CutDownA)}$$

We will now turn to the parameter *SylMis,* which describes the number of extra syllables a misunderstanding adds to the dialogue. The risk of accepting an incorrect concept during the positioning phase is that the set of possible user positions may be erroneously constrained. If this happens, the positioning often has to start all over again. Thus, *SylMis* should reflect the number of syllables a complete positioning takes (on average 97.0, which we will refer to as *SylPos*). However, the set of possible user locations does not *need* to be erroneously constrained when accepting an incorrect concept (the user may actually see a red building, even if this was not what she said). The probability that the correct position actually is lost can be described by the parameter *CutDownH* defined above, i.e., the proportion of possible locations that is reduced if the hypothesised concept is accepted. Thus *SylMis* can be calculated as follows:

[1] By important concept, we mean concepts that contribute in the current task. In this example, RED is important, but not BUILDING, since there are buildings everywhere.

(8) *SylMis* = *SylPos* x *CutDownH*

The rest of the parameters can be calculated from the data by counting the number of syllables spent on the grounding sub-dialogues and the number of times they failed. These parameters are shown in Table 3. *SylGA* is the number of syllables involved in the grounding act (in the case of DISPLAY or CLARIFY).

Table 3: Estimation of parameters.

Parameter	Value
SylClarCor	*SylGA* + 1.4
SylClarInc	*SylGA* + 2.1
SylDispCor	*SylGA* + 0.1
SylDispInc	*SylGA* + 1.2
P(Fail|Clar,Cor)	0.33
P(Fail|Disp,Inc)	0.82

The high value of *P(Fail|Clar,Cor)*, and especially *P(Fail|Disp,Inc)*, might be explained by the fact that the system did not use an elaborate prosodic model for the realisation of fragmentary DISPLAY and CLARIFY acts. Also, the use of such fragments is still very uncommon in dialogue systems, which often resulted in that the users did not recognise their function.

We will now consider two examples where the concept information gain differs a lot (the concepts under question are underlined):

(9) I can see a <u>mailbox</u> (*CutDownH* = 0.782; *SylGA* = 2)

(10) I can see a <u>two</u> storey building (*CutDownH* = 0.118; *SylGA* = 1)

CutDownA can be estimated from data as 0.336. Using these parameters, the cost function for the different grounding actions, depending on *P(correct)*, can be calculated to find out which action has the least cost for each value of *P(correct)* and thus derive confidence thresholds, as shown in Figure 1 and Figure 2. As can be seen in these figures, example (9) has a much higher information gain and thus a wide confidence interval where a clarification request is optimal, whereas example (10) has less information gain and is optimally either accepted or rejected, but never clarified.

In the previous examples, we have only considered the positioning phase of the dialogues. However, there is another important phase, the goal assertion phase:

(11) U: I want to go to an <u>ATM</u> (*SylGA*=3)

If this hypothesis would constitute a misunderstanding, it would lead to much higher costs than a misunderstood positioning statement. In this case, we can define *SylMis* as the number of syllables it takes on average until the user has reached the (incorrect) goal or restated the goal, which can be estimated to 261.6 from the data. We will assume that *SylRec* is equal to *SylCon* (15.0), and that the other parameters are the same as in the positioning phase. The cost functions and thresholds for grounding

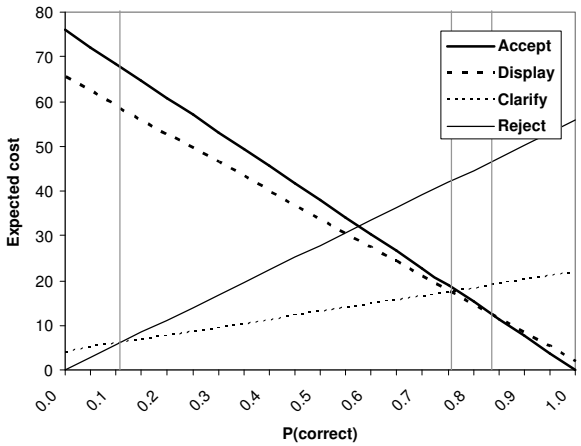

Figure 1: Cost functions and confidence thresholds for grounding the concept MAILBOX after "I can see a mailbox".

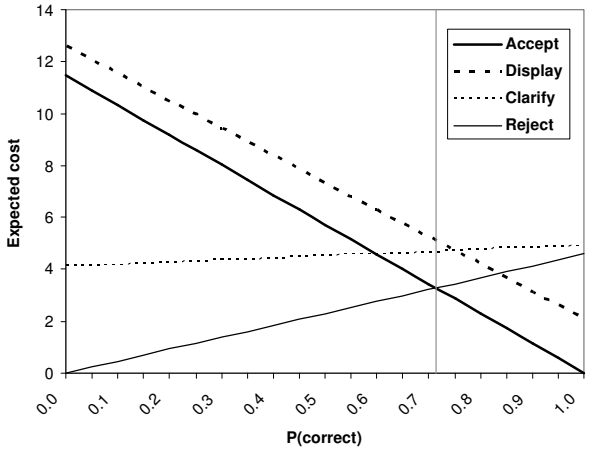

Figure 2: Cost functions and confidence thresholds for grounding the concept TWO after "I can see a two storey building".

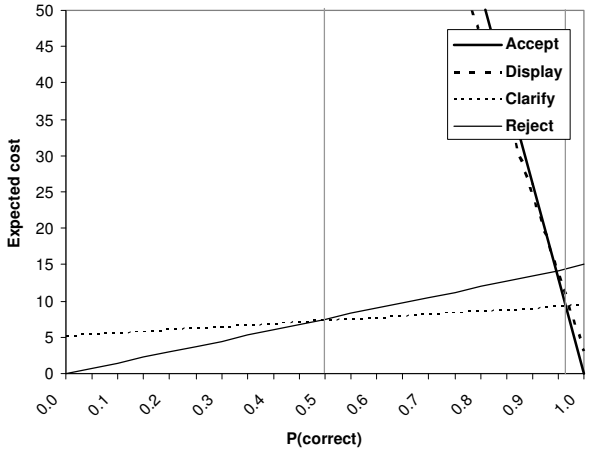

Figure 3: Cost functions and confidence thresholds for grounding the concept ATM after "I want to go to an ATM".

"ATM" in the example above are shown in Figure 3. Due to the high cost of misunderstandings, a simple accept requires a very high confidence, and goal assertions will therefore most often be clarified.

6 Discussion

The graphs presented above, and the calculation of thresholds, are of course only useful for illustrative purposes. A dialogue system would just calculate the most optimal action, given the value of *P(correct)*. It should be noted that these estimations are based on the data collected with hand-crafted confidence thresholds. If the derived thresholds would be applied to the system, the parameters values would change, thus affecting the thresholds. This means that the presented model should be derived iteratively, using bootstrapping, and the parameter values presented here are just the first step in such an iteration. To estimate the parameters, transcription of the dialogues and some annotation is needed. However, given that the logging is adapted for this, we believe that this can be done rather efficiently.

The functions presented in Table 2 describe general characteristics of the grounding actions and should be applicable to many different dialogue domains. However, the parameter estimation presented here is specific for the navigation domain. For some domains, it may be more problematic to use syllables as a general measure.

There are some simplifying assumptions in the model presented above. First, only one concept in the hypothesis is considered as correct or incorrect. It would of course also be possible to consider some concepts as correct and some concepts as incorrect. In such concept-level error handling (Skantze, in press), it is for example possible to clarify one concept while silently accepting or rejecting another. The model presented here could be extended to also cope with several concepts in an utterance with different probabilities, as in the following example (with probabilities in parenthesis):

(12) U: I can see a red building to the left
[RED (0.8) LEFT (0.2)]

In this case, we should consider 4 possible states instead of 2, 16 actions instead of 4, and 64 costs instead of 8. Here are some examples of the actions that should be considered:

Red? (CLARIFY RED, ACCEPT LEFT)
Do you have the red building on your left?
(DISPLAY RED, CLARIFY LEFT)
A red building on your left?
(CLARIFY RED, CLARIFY LEFT)

Another simplification is that temporal modelling of grounding (as discussed in Paek & Horvitz, 2003) is not considered, i.e., the fact that the utility of grounding actions change when they are repeated subsequently. However, it should be possible to account for this by conditioning the parameters, depending on the order in which the grounding action is taken. An elaborate model of *P(correct)* could also take this into account.

A more complex approach to grounding decisions is to use POMDP models (Williams & Young, 2007). The strength of such models is that they account for parallel recognition hypotheses and planning. The model presented here is much simpler and includes more bias. However, it requires less resources and is easier to apply and scale.

Of course, the presented model also remains to be evaluated, for example by comparing the performance of a system using this model with a system based on handcrafted thresholds.

References

Bohus, D., & Rudnicky, A. (2001). Modeling the cost of misunderstandings in the CMU Communicator dialog system. In *Proceedings of ASRU-2001*. Madonna di Campiglio, Italy.

Bohus, D., & Rudnicky, A. (2005). A principled approach for rejection threshold optimization in spoken dialog systems. In *Proceedings of Interspeech-2005*. Lisbon, Portugal.

Bouwman, G., & Hulstijn, J. (1998). Dialogue strategy redesign with reliability measures. In *Proceedings of the First International Conference on Language Resources and Evaluation*. Granada, Spain.

Bouwman, G., Sturm, J., & Boves, L. (1999). Incorporating confidence measures in the dutch train timetable information system developed in the Arise project. In *Proceedings of ICASSP'99*.

Clark, H. (1996). *Using language*. Cambridge University Press.

Edlund, J., Skantze, G., & Carlson, R. (2004). Higgins - a spoken dialogue system for investigating error handling techniques. In *Proceedings of ICSLP*. Jeju, Korea.

Jiang, H. (2005). Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4).

Paek, T., & Horvitz, E. (2003). On the utility of decision-theoretic hidden subdialog. In *ISCA Workshop on Error Handling in Spoken Dialogue Systems*.

Skantze, G. (in press). Galatea: A discourse modeller supporting concept-level error handling in spoken dialogue systems. To be published in Dybkjær, L., & Minker, W. (Eds.), *Recent Trends in Discourse and Dialogue*. Springer.

Williams, J. D., & Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2).