Higgins – a spoken dialogue system for investigating error handling techniques

Jens Edlund, Gabriel Skantze & Rolf Carlson

Centre of Speech Technology, KTH, Sweden
{edlund,gabriel,rolf}@speech.kth.se

Abstract

In this paper, an overview of the Higgins project and the research within the project is presented. The project incorporates studies of error handling for spoken dialogue systems on several levels, from processing to dialogue level. A domain in which a range of different error types can be studied has been chosen: pedestrian navigation and guiding. Several data collections within Higgins have been analysed along with data from Higgins' predecessor, the AdApt system. The error handling research issues in the project are presented in light of these analyses.

1. Introduction

This paper presents the HIGGINS project and the research done within the project. HIGGINS was instigated at CTT (Centre for Speech Technology) in mid-2003, with the aim of investigating error handling techniques in spoken dialogue systems (SDS) on several levels, from specific sub-processes to dialogue level. The practical goal of the project is to build a collaborative dialogue system in which error handling can be tested empirically. We have chosen a domain in which a range of different errors can be studied, and one which can easily be extended in a number of ways: pedestrian navigation and guiding. The dialogue system finds out where a user is and what the user's destination is by speech only, and provides the user with route descriptions. The user can also ask questions about prominent features in the surroundings. The Higgins project builds on previous dialogue system research at KTH and CTT, such as Waxholm, August and, most closely as well as recently, AdApt (for an overview, see [1]).

On a high level, an SDS can be seen as one single process that takes speech as its input and generates some other speech as its output. It may also be viewed as a number of sub-processes, each of which may potentially handle as well as introduce errors. The dialogue system, or any of its sub-processes, can be expected to produce a certain output, given its input. Errors, then, can be defined as deviations from the expected output, in the form of insertions, deletions, and substitutions. We take the position that the human interlocutor is faultless – any errors that occur in human-computer communication are introduced by the system. Human self-corrections, for instance, are not errors, but just another type of input that the system should be built to handle. It is important to note that in an SDS, the relation between some given input and the output it ought to generate is rarely a one-to-one mapping. The input signal, speech, is noisy and unpredictable on many levels, including background noise, channel noise, linguistic and phonetic ambiguity, etc., and in many cases, several different outputs are acceptable. For these reasons, errors may be better viewed as degrees and probabilities than in binary terms.

The Higgins project approaches a number of error handling issues, as well development of base technologies especially interesting from an error handling perspective, such as incremental dialogue system processing, on-line prosodic feature extraction, robust interpretation, and flexible output generation and production.

2. The Higgins Domain

The domain chosen for Higgins is that of pedestrian city navigation and guiding, which is similar to the now classic MapTask domain [2] as well as to a number of guide systems, such as REAL [3]. A user gives the system a destination and the system guides the user by giving verbal instructions. The system does not have access to user positions, but has to rely on their descriptions of their 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. The domain is complex enough to generate a variety of error types, and the surroundings users and system talk about contain landmarks, such as churches, buildings, roads and statues, that are challenging to interpret and represent semantically.

The domain can be extended in several ways, for example by allowing the user to ask for information about landmarks, making the domain span information-seeking dialogue as well as problem solving, which may call for different error handling strategies.

3. Architecture

A system in which combinations of techniques are tested empirically must be able to change its behaviour dynamically, on process level and as a whole. To this end, the well-tested distributed architecture used in AdApt has been extended to add further support for on-line configuration on all levels. Interprocess messaging is encoded in XML, providing easy visualisation of data on all levels, which facilitates system testing and surveillance. More importantly, domain knowledge is consistently represented as XML tree structures throughout the SDS. These structures are used to represent utterance semantics, discourse model, database content, and system output before surface generation. General processes are used to unify the structures during interpretation, discourse modelling, and databases searches. The semantic trees that are constructed during interpretation and discourse modelling can then be used directly to search the domain model for matching structures. For example, the semantics of D-U2 (below) can be used to find all buildings in the domain model that are large and made of wood.

The total domain model for the virtual city can be built by using a custom-made tool. It can also be translated into a VRML/X3D 3D model (Figure 1) of the city that can be used for simulation, which ensures that the system's model of the world reflects the user is walking around in.

4. Data collection

The design and research decisions in the project are based on analyses of three speech corpora collected at CTT. The data collections are presented next, including extracts from the corpora that highlight the research issues discussed in the final sections. All examples are translated from Swedish, including the ASR results.

4.1. ADAPT-1

The AdApt system allows users to browse the real estate downtown Stockholm. ADAPT-1 contains recordings of 26 subjects who spoke to the system in sessions lasting for a minimum of thirty minutes. The system does error handling on dialogue level by asking clarification questions:

A-U1 when was the apartment built
A-S1 i didn't get that could you rephrase
A-U2 when was the blue apartment built

Referential expressions and ellipses are also handled, as in:

A-U3 how much does the red apartment cost
A-S2 the green apartment costs nine hundred thousand
A-U4 i said the red apartment
A-S3 the green apartment costs nine one point one million

4.2. WOZASR

In the WozAsr experiment [4], eight pairs of subjects were asked to give route directions to each other in a simulated campus, resulting in Higgins-like dialogues. The subject giving the directions (the *operator*) could speak directly to the other subject (the *user*), but the user spoke through an ASR and the operator read the ASR results. This way, human handling of ASR errors typical for the Higgins domain could be studied. A dialogue fragment is given below, with ASR results in brackets.

W-O1 […] there you can take left until you reach the next crossing
W-U1 yes [name]
W-O2 which name are you asking about
W-U2 I am in the crossing between two concrete buildings [I am crossing between two concrete buildings]
W-O3 do you have a street on your right hand or is it concrete buildings – is the crossing only to your left
W-U3 the crossing goes in both directions [the crossing goes and two twelve]
W-O4 do you see any street number anywhere
W-U4 number eight on my lef* on my right side and number eighteen straight ahead [number eight to wait right and number eighteen straight ahead]
W-O5 okay eh then turn right on on this street […]

4.3. DESCRIBE 1 & 2

The simulation of the campus in the WozAsr experiments was done with a 2D-map. Thus, the users' descriptions of their positions do not reflect those of a 3D world (that will be used in Higgins) very well. Therefore, the 3D-model described above was created, and corpus of an initial eight followed by another 16 users moving around in the 3D simulation, describing their positions, was collected. Typical utterances include:

Figure 1: The Higgins 3D simulation.

D-U1 I am standing on a lawn and I have a brown building on my left - to the right is a red building
D-U2 a large wooden house
D-U3 on my right side I have a long three storey house that is blue
D-U4 on my left sid* side there is a stone house and also in front of me

5. Research issues

We will start by describing the different aspects of error handling that should be considered when building a dialogue system, and introduce some terms that are useful in the discussion of the Higgins research issues.

For each user utterance in the dialogue, the interpreting processes have to detect errors, and possibly correct them (i.e. delete insertions, re-insert deletions, and replace substitutions). We will call this *early error detection* and *correction*. Early detection and correction leads to a decision to either reject and disregard (parts of) the utterance, or to accept and assume understanding of the utterance. In the first case, the system has to perform *non-understanding recovery* that should both recover the lost information, if needed, and prevent similar future errors from occurring. If, on the other hand, the system decides to understand, error handling may be deferred to a later stage, in which case the system risks misunderstanding. In this case, the understanding of the utterance is typically signalled back to the user, commonly called *grounding* [5] or *feedback* [6]. This way, the user is involved in the error handling. Given the user's reaction to the signal, the system may detect that there was an error in the previous interpretation. We will call this *late error detection*. If an error has occurred at this stage, the system has to perform a *misunderstanding recovery* that should remove the erroneous information from the beliefs held by the system.

5.1. Early detection and correction

Robustness can be seen as the ability of a process to make the best of input that is noisy or unexpected, rather than simply failing when receiving such input. In the terms used here, this could be defined as the combination of early error detection

and correction. Robust parsing is a classic example, but robustness as a concept can be applied to the entire process chain – from large vocabulary ASR to dialogue management (see e.g. [7]). Single salient words (such as "red") may be interpreted as ellipses by the interpreter and passed on to the dialogue manager. The dialogue manager could then use the context to determine the correctness of such a concept.

In Higgins, the large vocabulary ASR KTH LVCSR [8] is used. The output of large vocabulary ASR of unconstrained spoken language is likely to contain errors. A robust interpreter, Pickering [9], has been developed within the Higgins project. To handle the pedestrian navigation domain, some syntactic analysis is needed in order to capture, for example, relations between objects. Pickering can automatically make exceptions from the syntax given in the grammar by handling insertions and non-agreement inside phrases and by combining non-continuous phrases. While deviations from the grammar are allowed by the Pickering interpreter, they are taken into account when the scoring the interpretations. Preliminary tests show that these techniques do indeed improve robustness on when evaluated against the DESCRIBE 2 corpus [9]. An example of where allowing insertions may be used for error detection is shown in D-U3', where utterance D-U3 has been recognised incorrectly:

D-U3' [on my right side I have a flower three storey house that is blue]

A keyword spotter would very likely generate an error on the content word *flower*. Pickering, instead, interprets it as an unexpected word inside the noun phrase. Thus, it does not generate any semantics for it.

There were very few misunderstandings in the WOZASR experiments, which suggest that the subjects were very good at early error detection. Where misunderstanding did occur, as in utterance W-U1, information from sentence structure and dialogue context did not contribute. In other cases, such as W-U2, this information probably contributes to the early error detection. The sentence structure may be used by an interpreter such as Pickering for error detection, but it is not obvious how the context could be used. However, error detection may also be done as a separate processing step. Studies have shown that machine-learning can be used to detect the presence of errors at utterance level in ASR results using contextual information as well as prosody (e.g. [10]), and post-processing of ASR results has been used to increase performance (e.g. [11]). Preliminary tests on an off-the-shelf recogniser indicate that errors on word-level can be detected using machine-learning techniques. Techniques such as these make it possible to correct some errors in the input, but it is also important that information about detected errors is passed on to other modules that may alter their behaviour based on the information (e.g. reliability scores). Currently, we are examining to what extent word level reliability measures can improve the interpreter performance and how it is best used when calculating interpretation reliability.

5.2. Late detection and correction

To handle late detection and correction, there are two central issues: how to select an appropriate amount of grounding for each semantic concept in order to elicit the right response from the user, and how to detect erroneous concepts based on that response.

A discourse modeller has been developed that can join several semantic results from PICKERING into a large discourse model. The discourse modeller also adds grounding information to the semantics. This includes information about who contributed the information, whether it was presupposed, added or requested, the turn and point in time at which it was contributed, and, for user utterances, the reliability measure of the information. This information will be used to determine what needs to be grounded and to remove information that is associated with a turn that turns out to be a misunderstanding from the discourse model. The dialogue manager and generator have the important task of selecting the right amount of feedback. When, for example, referring to an entity, there are several possible realisations that range from a simple pronoun to a complete description of the object. A richer expression facilitates late error detection, but makes the utterance longer. The explicitness should be governed by the consequences of an uncaught misunderstanding as well as by the system's confidence in its interpretation. The system's confidence may also be signalled with prosodic cues, and the output description formalism GESOM [12] is being expanded to provide flexible control over such features.

In AdApt, references to apartments are always grounded with a colour (A-S2). Misunderstandings can easily be corrected by the user (A-U4). Error correction, however, is done without late detection; no distinction is made between for example "no, the red one" and "and the red one". This type off correction works for exchanges where the slot (apartment id in the example) can only take one value at a time. For other tasks, error detection may be necessary. For example, there is a difference between adding "and to the right" and "no to the right" to utterance D-U4, which makes late error detection relevant in the HIGGINS domain. Studies have shown that machine learning may be used for this task [13] and that prosodic information is relevant [14].

5.3. Incrementality

A user turn in a SDS is often defined as an utterance that ends when a certain amount of silence is detected. This approach leads to difficulties, some of which are pointed out in [15]. The following example illustrates the problem of late detection, if utterance D-U1 would have been grounded as one unit:

U1 I am standing on a lawn and I have a brown building on my left. To the right is a red building.
S1 ok, you have a brown building on your left and a green building on your right.
U2 no, red.

Firstly, it is not easy to determine which colour is corrected by the user. Secondly, if the system only wants to detect the correctness of the colours of the buildings, it still has to ground the whole utterance, including the directions of the buildings, in order to relate the colours to the right buildings. If the system had grounded the information immediately and incrementally, these problems could be handled:

U1a I am standing on a lawn and I have a brown building…
S2 mhm, a brown building
U1b (cont): on my left. To the right is a red building…
S3 mhm, a green building
U2: no red

Incremental dialogue processing requires that all input processes work incrementally and fast enough for the feedback to achieve good timing. However, incrementality may also increase efficiency, since much processing can be done while the user is speaking. All components implemented within Higgins support incremental processing.

Incremental dialogue processing opens up several interesting research issues. For example, when is it permissible for the system to barge in? In order to make this decision, semantic content and prosody of user utterances should most likely be considered, and a prosodic feature extractor is under development in connection to the project.

Another question is whether the interpreter should parse syntactic constructs that range over several utterances (as "a brown building … on my left" in U1a-U1b above), or if these should be joined by the dialogue manager. If they are resolved by the interpreter, grammar rules can be written to handle user corrections such as "a green building … no red", which is what the system perceives in U1b-U2 above.

The rapid feedback in utterance S3 and S4 could be realised as short, unobtrusive "mumbling", but it could also be realised multi-modally, as in [16]. For generation and speech synthesis, making the feedback rapid and unobtrusive is a challenging issue which is presently being addressed.

5.4. Error recovery

When non-understandings occur, a common strategy taken in many dialogue systems is to signal non-understanding, as in A-S1, thus encouraging the user to repeat or rephrase the utterance. Analysis of the WozAsr corpus suggests that such signalling of non-understanding is not very common and that it, in general, leads to a decreased experience of task success and to slower error recovery. A common strategy was to, instead of signalling non-understanding, asking a task-related question that forwards the dialogue. An example of this can be seen in utterance W-O4. The operator does not rely on the numbers perceived from W-U3, but asks a question about street numbers to check if the assumption is correct. From the user's perspective, there is no sign of non-understanding in that utterance. Such a *non-understanding recovery* strategy may in some cases be preferable and will be tested in the Higgins SDS.

6. Conclusions

Based on the analysis of the corpuses collected, a number of research issues and possible approaches to them have been outlined for the Higgins project. The main challenge in Higgins is to build a dialogue system capable of using and shifting between a large number of error handling techniques on many levels, and doing user tests with different combinations of these to see how they affect each other and the users under different circumstances. Ultimately, we want to find out how to dynamically control the balance between these techniques depending on the situation.

7. Acknowledgements

This research was carried out at the Centre for Speech Technology, a competence centre at KTH, supported by VINNOVA (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organisations.

8. References

[1] Gustafson, J. (2002). *Developing Multimodal Spoken Dialogue Systems - Empirical Studies of Spoken Human-Computer Interactions*. PhD Thesis, KTH, Stockholm.

[2] Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H., & Weinert, R. (1991). The HCRC Map Task Corpus. *Language and Speech*, 34(4), 351-366.

[3] Baus, J., Kray, C., Krüger, A., & Wahlster, V. (2001). A resource-adaptive mobile navigation system. In *Proceedings of the International Workshop on In-formation Presentation and Natural Multimodal Dialog*.

[4] Skantze, G. (2003). Exploring human error handling strategies: Implications for spoken dialogue systems. In *Proceedings of the ISCA workshop on Error Handling in Spoken Dialogue Systems*.

[5] Clark, H. H. (1996). *Using language*. Cambridge: Cambridge University Press.

[6] Allwood, J., Nivre, J., & Ahlsén, E. (1992). On the semantics and pragmatics of linguistic feedback. *Journal of Semantics.*

[7] Allen, J. F., Miller, B., Ringger, E., & Sikorski, T. (1996). A robust system for natural spoken dialogue. In *Proceedings of ACL*, 62-70.

[8] Seward, A. (2003). *Efficient methods for automatic speech recognition*. PhD Thesis, KTH, Stockholm.

[9] Skantze, G., & Edlund, J. (2004). Combining methods for robust interpretation of spoken language with deep semantic structures. Submitted to *ICSLP 2004*.

[10] Litman, D. J., Hirschberg, J., & Swertz, M. (2000). Predicting automatic speech recognition performance using prosodic cues. In *Proceedings of NAACL*. 218-225.

[11] Ringger, Eric K. & Allen, James F. (1996). A fertility channel model for post-correction of continuous speech recognition. In *Proceedings ICSLP'96*, Philadelphia, PA .

[12] Beskow J., Edlund J., & Nordstrand M. (2004): A model for generalised multi-modal dialogue system output applied to an animated talking head. In Minker, W., Bühler, D., & Dybkjaer, L. (eds) *Spoken multimodal human-computer dialogue in mobile environments*, Dordrecht, The Netherlands, Kluwer Academic Publishers.

[13] Krahmer, E., Swerts, M., Theune, T. & Weegels, M. E. (2001). The dual of denial: Two uses of disconfirmations in dialogue and their prosodic correlates. *Speech Communication*. 36(1), 133-145.

[14] Krahmer, E., Swerts, M., Theune, T. & Weegels, M. E. (2001). Error detection in spoken human-machine interaction. *International Journal of Speech Technology*. 4(1), 19-30.

[15] Allen, James, Ferguson, G., & Stent, A. (2001). An architecture for more realistic conversational systems. In *Proceedings of IUI-01*, 1-8. Santa Fe, NM.

[16] Gustafson, J., Bell, L., Boye, J., Edlund, J., & Wiren, M. (2002). Constraint manipulation and visualization in a multimodal dialogue system. In *Proceedings of Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany.