An Empirical Study of Speech Recognition Errors in a Task-oriented Dialogue System

Marc CAVAZZA
School of Computing and Mathematics,
University of Teeside,
TS1 3BA, Middlesbrough, United Kingdom,
m.o.cavazza@tees.ac.uk

Abstract

The development of spoken dialogue systems is often limited by the performance of their speech recognition component. The impact of speech recognition errors on dialogue systems is often studied at the global level of task completion. In this paper, we carry an empirical study on the consequences of speech recognition errors on a fully-implemented dialogue prototype, based on a speech acts formalisms. We report the impact of speech recognition errors on speech act identification and discuss how standard control mechanisms can participate to robustness by assisting the user in repairing the consequences of speech recognition errors.

Introduction

The development of spoken dialogue systems is faced with limitations in speech recognition technologies that make recognition errors a recurring problem for any dialogue system. Several studies have shown little correlation between speech recognition scores and user satisfaction, or the ability to complete the tasks underlying spoken dialogue [Yankelovich et al., 1995] [Dybkjaer et al., 1997], suggesting that a certain level of errors should not prevent spoken dialogue systems from being successful.

However, most of the studies on speech recognition errors have concentrated either on parsing incomplete utterances or on global dialogue robustness, i.e. at task completion level [Allen et al., 1996] [Stromback and Jonsson, 1998] [Brandt-Pook et al., 1996].

In this paper, we investigate the impact of speech recognition errors on a fully-implemented prototype for a task-oriented dialogue system. This system supports a conversational character for Interactive Television and is based on a speech acts formalism. We report a first empirical study on the consequences of speech recognition errors on the identification of speech acts, and the conditions under which the system can be robust to those errors.

1 System Overview

The VIP ("Virtual Interactive Presenter") system is a dialogue-based interface to an Electronic Programme Guide (EPG). One main advantage of human-computer dialogue is that it breaks down the information exchange into elementary units that correspond to the actual criteria on the basis of which TV programmes are selected, i.e. individual features such as the cast, the movie genre, its rating, etc. It assists the user in progressively refining the programme description without requiring explicit knowledge of the editorial categories used to index the EPG.

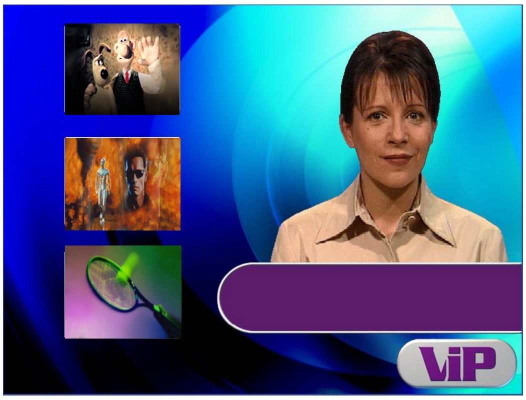

Figure 1. The System Interface.

Related applications, i.e. dialogue systems involving the choice of a film or TV programme have been also described in [Hagen, 2000] [Ludwig et al., 2000]. Our system is a mixed-initiative conversational interface organised around a human-like character with which the user communicates through speech recognition [Nagao and Takeuchi, 1994] [Beskow and McGlashan, 1997]. The interface is based on the Microsoft Agent™ system with a set of animated bitmaps acquired from a real human subject (Figure 1).

An example dialogue illustrating the system capabilities is presented below (this example has been obtained with keyboard input only). After greetings by the system, the user opens the dialogue with a first request (U1). Even though the system's goal is to refine the selection, it offers potential choices even at early stages of the dialogue, e.g. after only the programme genre has been specified (S2). As the dialogue progresses, it gives a count of programmes matching the current criteria and proposes the best one. The system acknowledges the most specific selections only (S10, S14). Also, the system takes initiative whenever necessary (S12). It repairs non-productive dialogue, i.e. when several utterances have not altered the programme description ("is this programme all right, then?"). Whenever high-level categories are rejected by the user, leaaving the system without sufficient criteria to filter the EPG contents, it re-starts the dialogue ("what would you like then?", S12).

U1: Do you have any sports programmes
S2: *I have found 5 programmes for this choice. Would you like to watch "Keegan's greatest games"?*
U3: Can I have some cricket instead?
S4: *I can suggest this programme with cricket: "Sportstalk"*
U5: Who's conducting?
S6: *"Sportstalk" features Donald*
U7: Anything else?
S8: *I have no other programmes for that selection*
U9: Is there anything funny, then?
S10: I can recommend the following comedy: "Starting Out"
U11: I do not want a sitcom
S12: *What would you like to see, then?*
U13: A funny movie
S14: *I would suggest this programme with comedy: "Analyze This"*
U15: Can I have another one?
S16 *There are 1 other programmes for this choice. What about the following programme: "Casablanca"?*
U17: I want a western instead
S18: *I would suggest the following western: "Unforgiven"*

The software architecture is a pipeline comprising speech recognition, parsing and dialogue. In the next sections, we describe each of these components from the perspective of speech recognition errors. Finally, we discuss the impact of speech recognition errors on example dialogues and the mechanisms that contribute to dialogue robustness.

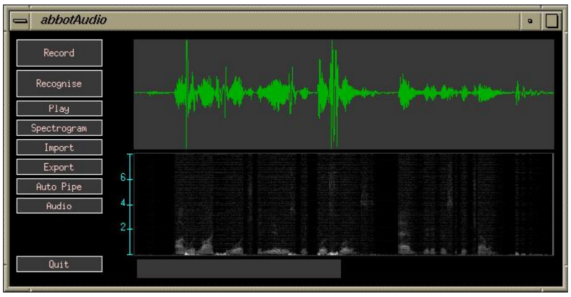

Figure 2. The ABBOT Interface.

2 The Speech Recognition Component

Speech recognition is based on the ABBOT system [Robinson et al., 1996]. A specific ABBOT version has been developed for the VIP prototype, VIP-ABBOT, with a test vocabulary of 300+ words (Figure 2). This version is based on a trigram model, trained on a small corpus of 200 user questions and replies, using data from six speakers (average recording time is twelve minutes). Though the size of the corpus is in principle too small to obtain an accurate language model, the VIP-ABBOT system achieves a satisfactory performance. Global speech recognition accuracy has been tested as part of the development of the VIP-ABBOT version. The recognition accuracy varied across tests from 65 to a maximum 80 % (at this stage only laboratory conditions with non-noisy environments and good quality microphones have been considered). The system outputs the 1-best recognised utterance, which is passed to the dialogue system via a datagram socket.

We have assembled an evaluation corpus of 500 utterances, collected from five speakers including one non-native speaker. Including a non-native speaker was an empirical way of increasing the error rate. Other researchers have suggested varying parameters of the speech recognition system, such as the beam width [Boros et al., 1996], as a method to increase word error rate, in order to collect error corpora. However, they have not documented whether the kind of errors induced in this way actually reproduce (in terms of distribution) those obtained during the actual use of the system. On the other hand, recognition errors obtained with native and non-native speakers appear similar in our experience, the overall error rate just being higher in the latter.

For the whole corpus, approximately 50% of recognised utterances contain at least one speech recognition error.

3 Integrated Parsing of User Utterances

Strictly speaking, a significant proportion (around 50%) of the recognition hypotheses produced by VIP-ABBOT are ungrammatical. For obvious reasons, and since the early stages of system development, we have abandoned the idea of producing a complete parse for the speech input, not so much because user expressions themselves could be ungrammatical but rather because recognised utterances were most certain to be, considering the error rate.

One of the key questions for parsing, especially in the case of dialogue, where the average utterance length is 5-7 words, is whether complete parsing is at all necessary [Lewin et al., 1999]. We have implemented a simplified parser based on a variant of Tree-Adjoining Grammars [Cavazza, 1998], This syntactic formalism being lexicalised has interesting properties in terms of syntax-semantics integration. This lexicalised formalism, combined with a simple bottom-up parser, is well adapted to the partial parsing of ungrammatical utterances (Figure 3).

The main goal of parsing is to produce a semantic structure from which speech acts can be identified. Semantic features are aggregated as the parsing progresses following the syntactic operations. As a result, the parser produces a feature structure whose semantic elements can be mapped to the descriptors indexing the programmes in the EPG, such as genre (e.g. "movie", "news", "documentary"), sub-genre (e.g., "comedy", "lifestyle"), cast (e.g., "Jeremy Clarkson"), channel ("BBC one"), rating (e.g., "caution", "family"), etc.

Whenever the parser fails to produce a single parse, the semantic structures obtained from partial parses are merged on a content basis. For instance, descriptors such as "cast" or "channel" are attached to programme descriptions, etc. This process confers a good level of robustness and tolerance to ungrammaticality. This kind of approach, where dialogue strategy is privileged over parsing was inspired from early versions of the AGS system [Sadek, 1999]. These semantic structures are used to generate search filters on the EPG database, which correspond to semantic descriptions of the user choice. They are also used for content-based speech act identification, by comparing the semantic contents of successive utterances [Cavazza, 2000].

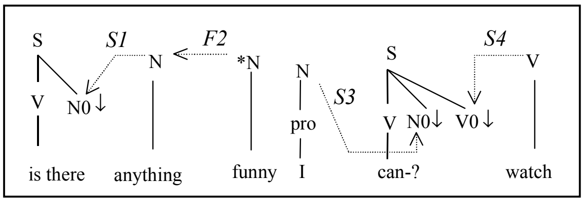

Figure 3. Parsing of an Utterance

4 The Dialogue Process

The dialogue strategy has been determined mainly from the task model. As the task is to progressively refine a programme description by using elementary dialogue acts, we have adopted a speech acts based approach [Traum and Hinkelmann, 1992]. Each speech act corresponds to a specific construction operation – it is possible to map communicative operations (rejection, implicit rejection, specification, etc.) to the updating of the programme description, which is a filter through which the EPG database is searched.

We are using a content-based approach to the identification of speech acts [Cavazza, 2000]. This method has similarities with the one previously described by Maier [1996]. Another source of inspiration was the work of Walker [1996], though it was restricted to the recognition of acceptance rather than a complete set of speech acts. Figure 4 shows the construction of search filters from the semantic contents of user utterances. Once a new utterance is analysed, its semantic contents are compared with the active search filter, which has been constructed from previous user utterances, and this comparison determines speech act recognition. For instance, when the last utterance contains semantic information for a programme sub-genre, the speech act is a *specification*. Explicit rejections are signalled by markers of negation, while *implicit rejection* speech acts are recognised when the semantic contents of the latest utterance overwrite the descriptors of the current filter (this is the case when, for instance, when the current filter contains the comedy sub-genre and the user asks "can I have a western?").

In this context, speech acts provide a unified and consistent way to determine the most appropriate answer to the user as well as the way in which the search filter should be updated at each dialogue turn. In the next section, we propose a first empirical categorisation of speech recognition errors according to their impact on the dialogue process.

5 From Speech Recognition Errors to Speech Acts Recognition Errors

Traditional error metrics used in speech recognition such as "word accuracy" are not reliable to measure the global consequences of speech recognition errors on the dialogue process. This is why it has been proposed that a "concept accuracy" be used in place of a word accuracy. These two metrics appear however to be linearly correlated [Boros et al., 1996].

Word errors result in semantic errors, which in turn result in speech act recognition errors. It is the latter phenomenon that we are investigating, though through a largely empirical approach.

We have developed the following experimental approach to assess the consequences of speech recognition errors. Within a set of possible "test" dialogues, we have been running each user reply on several users to collect an error corpus. Then the impact of errors at critical points of dialogue can be tested on the system using textual input, entering the utterances comprising speech recognition errors. This also makes possible to explore the consequences of a set of errors at various stages of the dialogue process (see below).

In doing so, we are essentially exploring system behaviour in a glass box approach: this does not constitute an evaluation method for dialogue performance [Walker et al., 1997]. The only global metric that can be used in our case is dialogue length, as a measure of the extra dialogue turns required to repair the consequences of a speech recognition error.

5.1 "Fatal" Speech Recognition Errors

Obviously, the worst errors are those which terminate the dialogue by selecting a programme that the user intended to reject. This happens when the semantic structure produced from the mis-recognised utterance can be interpreted as an acceptance speech act. In the following short dialogues, the user implicitly rejects the selection offered by the system. He does so by specifying either the parental rating (U21) or the sub-genre (U25). However, due to speech recognition errors, specific semantic information disappears in both cases, while the insertion of the word "good" results in the system producing a semantic structure that is recognised as an *acceptance* speech act.

U19: Can I have a movie?
S20: *I have found 22 programmes for that selection. I would recommend: "Midnight Fear"*
U21: **I want a movie my good can watch** (REF: I want a movie my kids can watch)
S22: *Here is "Midnight Fear", then. Goodbye!*

U23: Do you have any movies?
S24: *I found 22 programmes for your request. I would suggest: "Twelve Monkeys"*
U25: **Good if I mean anything for long** (REF: could you find me anything funny)
S26: *Here is "Twelve Monkeys", then. Goodbye!*

5.2 User Correction of Mis-recognition

Some speech recognition errors result in an incorrect speech act being recognised, but can be easily corrected by the user at the expense of a few additional dialogue turns, as shown by the following example (especially U31).

U27: I would like to watch a movie
S28: *I found 22 programmes for this choice. I can recommend: "The Day of the Jackal"*
U29: **Can I have worse him?** (REF: can I have a western?)
S30: *What about the following programme: "Midnight Fear"*?

U31: No, I want a western
S32: *I can recommend this programme which is a western: "Unforgiven"*

The next dialogue illustrates an important aspect of speech recognition errors. Their impact actually varies according to the dialogue stage at which they take place. The same kind of error that resulted in premature acceptance has less dramatic consequences when it occurs in the opening request. It results in a partial understanding, where the EPG is searched for the right genre ("movie") but without the right parental rating. This additional criterion can thus be subsequently added by the user, resulting in a successful search at the price of a longer dialogue.

U33: **Is there a movie my good can watch?** (REF: is there a movie my kids can watch?)
S34: *I have 22 programmes for your request. Would you like to watch: "Midnight Fear"*
U35: What is its rating?
S36: *Its rating is caution*
U37: No, I want a movie my kids can watch
S38: *I would recommend this programme corresponding to that "family" rating: "Rio Bravo"*

5.3 System Detection of Mis-recognition

Some dialogue control mechanisms originally developed to monitor dialogue progression can also assist in recovering from speech recognition errors. The system can detect dead ends in dialogue, for instance when not enough information is available for programme selection. The speech recognition error in U39 results in a semantic structure rejecting sports programmes as "boring". In the absence of any category selected, the system returns to the user (S40), who can then correct the selection: an explicit request for the sports category over-writes its previous rejection.

U39: **What kind of sports boring you have?** (REF: what kind of sports programmes do you have)
S40: *What would you like to see, then?*
U41: **A sports programme**
S42: *I could find 5 programmes for this choice. I can recommend: "Row Z"*

Also, the system reacts to non-productive dialogue, i.e. utterances that simply repeat categories already represented in the programme descriptors. This has an interesting side effect, in that it can detect deletion of negation markers, as in the example below.

S43: *What about the following programme: "Countryfile"*
U44: **Again want a documentary** (REF: I don't want a documentary)
S45: *Is this programme all right, then?*

5.4 "Semantic" Robustness

The basis for robustness can be seen as a combination of shallow processing and natural redundancy of information in some of the user replies. For instance, in U52, even though the entry "role" as been deleted from the recognised utterance, there are sufficient semantic cues to activate a "cast" descriptor.

U46: I would like to watch a movie
S47: *I found 22 programmes for this choice. I would recommend: "Midnight Fear"*
U48: **Good if I mean anything funny** (REF: could you find me anything funny)
S49: *I can recommend this programme which is a comedy: "Casablanca"*

U50: what kind of movies do you have
S51: *I have found 22 programmes for this choice. I would suggest: "Twelve Monkeys"*
U52: **Who is playing the title all?** (REF: who is playing the title role?)
S53: *"Twelve Monkeys" features Bruce Willis*

U54: Do you have any documentaries
S55: *I have 2 programmes for your request. My first selection is: "Ground Force"*
U56: **An old gardening is not a little A** (REF: no, gardening is not OK)
S57: *I can recommend this programme, which is not a gardening: "Fangs"*

Another element of robustness lies in the surface form of some linguistic expressions for acknowledgement, acceptance and rejection, which often contains redundant information: "yes, this is fine", "no, I don't like that one", "no, gardening is not OK", etc.

User: IS THERE ANYTHING FUNNY I CAN WATCH
Recognised: THE IS THERE ANYTHING FUNNY I CAN WATCH
Semantics: `(((QUESTION) (EXIST) (PROGRAMME ((CHOICE+))) (DET))`
`((GOOD) (SUB_GENRE COMEDY)) ((AUDIENCE USER)) ((QUESTION))`
`((VIEW)))`
Filter: `((SUB_GENRE COMEDY))`
Speech Act: `(INITIAL (SUB_GENRE COMEDY) SEARCH)`

System: I found 4 programmes corresponding to that selection. What about: “Starting Out”?

User: NO I DON’T WANT A SITCOM
Recognised: AN OLD I DON'T WANT A SITCOM
Semantics: `((CHOICE+) (NEGATIVE) (AUDIENCE ((AUDIENCE USER)))`
`(PROGRAMME`
`((SUB_GENRE COMEDY) (GENRE ENTERTAINMENT) (INDET))))`
Filter: `((GENRE ENTERTAINMENT) (SUB_GENRE COMEDY) (NEGATIVE))`
Speech Act: `(REJECT GENRE NIL NO_SEARCH)`

System: What would you like to see, then?

User: A COMEDY MOVIE
Recognised: A COMEDY MOVIE
Semantics: `(((SUB_GENRE COMEDY) (INDET)) ((GENRE MOVIES)))`
Filter: `((GENRE MOVIES) (SUB_GENRE COMEDY))`
Speech Act: `(SPECIFY SUB_GENRE COMEDY SEARCH)`

System: My first selection is this programme with comedy: “Casablanca”

User: YES THIS IS FINE
Recognised: YES THE IS FIND THAT
Semantics: `(((GOOD) (POSITIVE)) ((DET)) ((ATTRIBUTE)))`
Speech Act: `(SIMPLE-ACCEPT)`

System: Here is “Casablanca”, then. Goodbye!

Figure 4. A Sample Dialogue with Internal Representations

5.5 “Accidental” Robustness

Another form of robustness appears to be rather accidental. It happens when a incorrect speech act is recognised, but its effects in terms of system response are a subset of those of the correct speech act. In the following example, the user intends to reject the proposal (“Salvador”) by specifying a parental rating. This is actually interpreted as a simple rejection of the programme instance (*Other* speech act), preserving the current “movie” category selected. The new proposal happens to match the criteria by chance (i.e., the parental rating is not grounded in the system reply, see e.g. S38).

U58: What movies do you have?
S59: *I found 22 programmes for this choice. I can suggest: “Salvador”*
U60: **Other one like is this to there it is** (REF: I don’t want my kids to see this)
S61: *I have 21 other programmes for that selection. What about the following programme: “Casablanca”*

Conclusion

We have explored the consequences of speech recognition errors in terms of content-based speech acts recognition. There appears to be a number of factors that support the robustness of the system to speech recognition errors, among which the fact that dialogue control mechanisms triggered by speech act recognition can contribute to repairing the consequences of speech recognition errors. Some improvement is possible in the treatment of errors involving mismatches between categories and connotations (such as “funny motoring”), by including semantic consistency checks. On the other hand, errors involving wrongful acceptance and dialogue termination appear difficult to deal with.

Finally, Fischer and Batliner [2000] have investigated which system replies are most likely to upset the user. These replies cannot always be always avoided, though, precisely because they are used to repair incorrect understanding or inconsistent one. It is thus important to investigate whether speech recognition errors increase the occurrence of these upsetting replies (apart from the unavoidable and necessary repairs). Obviously, in our context the most upsetting cases are the selection of a programme explicitly rejected by the user. However, It would also be necessary to explore whether the repair mechanisms described above are well accepted by the users.

Acknowledgements

James Christie at Cambridge University Engineering Department developed the VIP-ABBOT version and produced some of the global speech recognition results. Tony Robinson (now at Softsound Ltd) is thanked for his assistance in using the ABBOT system. Steve Francis developed the user interface and the character. EPG data has been provided by the BBC: David Kirby, Matthew Marks and Adam Hume are thanked for their support. This work has been funded in part by the DTI, under the DTI/EPSRC “LINK Broadcast” Programme.

References

James F. Allen, Brad Miller, Eric Ringger, and Teresa Sikorski (1996). Robust Understanding in a Dialogue System. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, San Francisco, pp. 62-70.

Jonas Beskow, and Scott McGlashan (1997). Olga: A Conversational Agent with Gestures. In: *Proceedings of the IJCAI'97 workshop on Animated Interface Agents - Making them Intelligent*, Nagoya, Japan, August 1997.

Manuela Boros, W. Eckert, F. Gallwitz, G. Gorz, G. Hanrieder, and H. Niemann (1996). Towards Understanding Spontaneous Speech: Word Accuracy Vs. Concept Accuracy. *Proceedings of the Int. Conf. on Spoken Language Processing (ICSLP’96)*, Philadelphia, pp. 1009-1012.

Hans Brandt-Pook, Gernot A. Fink, Bernd Hildebrandt, Franz Kummert, and Gerhard Sagerer. (1996). A Robust Dialogue System for Making an Appointment. *Proceedings of the Int. Conf. on Spoken Language Processing (ICSLP’96)*, Philadelphia, pp. 693-696.

Marc Cavazza, (1998). An Integrated TFG Parser with Explicit Tree Typing. In: *Proceedings of the fourth TAG+ workshop, IRCS, University of Pennsylvania*, pp. 34-37.

Marc Cavazza (2000). From Speech Acts to Search Acts: a Semantic Approach to Speech Act Recognition. Proceedings of GOTALOG 2000, Gothenburg, Sweden, pp. 187-190, June 2000.

Kerstin Fischer and Anton Batliner (2000). What Makes Speakers Angry in Human-Computer Conversation. *Proceedings of the Third Human-Computer Conversation Workshop (HCCW)*, Bellagio, Italy, pp. 62-67.

Eli Hagen (2000). A Flexible Spoken Dialogue Manager. *Proceedings of the Third Human-Computer Conversation Workshop (HCCW)*, Bellagio, Italy, pp.68-73.

Ian Lewin, Ralph Becket, Johan Boye, David Carter, Manny Rayner, and Mats Wiren (1999). Language processing for spoken dialogue systems: is shallow parsing enough? *Accessing Information in Spoken Audio: Proceedings of ESCA ETRW Workshop*, Cambridge, pp. 37--42.

Bernard Ludwig, Martin Klarner, Heinrich Niemann and Gunther Goerz (2000). Context and Content in Dialogue Systems. *Proceedings of the Third Human-Computer Conversation Workshop (HCCW)*, Bellagio, Italy, pp. 105-111.

Elisabeth Maier (1996). Context Construction as Subtask of Dialogue Processing: the VERBMOBIL Case. *Proceedings of the Eleventh Twente Workshop on Language Technologies (TWLT-11), Dialogue Management in Natural Language Systems*, University of Twente, The Netherlands, pp. 113-122.

Katashi Nagao and Akikazu Takeuchi,(1994). Speech Dialogue with Facial Displays: Multimodal Human-Computer Conversation. In: *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*, pp. 102-109.

Tony Robinson, Mike Hochberg and Steve Renals (1996). The use of recurrent neural networks in continuous speech recognition. In: C. H. Lee, K. K. Paliwal and F. K. Soong (Eds.), *Automatic Speech and Speaker Recognition – Advanced Topics*, Kluwer.

Lena Stromback and Arne Jonsson. Robust interpretation for spoken dialogue systems. (1998). *Proceedings of ICSLP'98*, Sydney, Australia.

David Traum and Elisabeth A. Hinkelman (1992). Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence*, vol. 8, n. 3.

Marilyn A. Walker (1996). Inferring Acceptance and Rejection in Dialogue by Default Rules of Inference. *Language and Speech*, 39-2.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm and Alicia Abella (1997). PARADISE: A Framework for Evaluating Spoken Dialogue Agents. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 271-280.

Nicole Yankelovich, Gina-Anne Levow and Matt Marx (1995). Designing Speech Acts: Issues in Speech User Interfaces. *Procedings of CHI'95*, Denver.