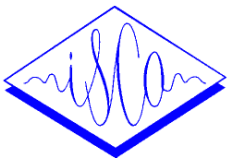
ISCA Archive
http://www.isca-speech.org/archive



6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

VISUALISATION OF SPOKEN DIALOGUES

BODA Péter Pál

Nokia Research Center, Speech and Audio Systems Laboratory, Helsinki, Finland
Email: peter.boda@nokia.com

ABSTRACT

Evaluation of spoken dialogue systems is in the main interest of the application developer. During the whole development cycle reliable and straightforwardly interpretable measures are needed in order to identify weak points of the user interface and the actual implementation. This paper proposes two issues. First, a visualisation method is introduced in order to follow what exactly happens in spoken dialogues during the interaction and thus giving means for system developers to explore dialogue paths explicitly in a statistical sense. The formalism of the visualisation process is explained and examples are given for different interaction types. The second issue is the introduction of a measure derived directly from the visualisation method. This measure is a sort of combination of traditional success rate and average number of system-user turns. The measure provides means for system developers to compare the average system performance to a pre-defined interaction described by a dialogue path. Example is given with a speaker-independent name dialling application.

1. INTRODUCTION

Application developers must cope with pre-mature systems during the process of development. In order to identify the weak points of the implementation, reliable measures are needed both on the overall and down on the individual module levels. This view was expressed by the work of Fraser and Simpson [1] when the concept of glass-box/black-box evaluation was introduced. According to this methodology, several system level (black-box) performance figures were used along with module-dependent figures (glass-box). Using the glass-box measures, such as speech recognition accuracy, semantic concept accuracy, success of database fetch, etc., a relatively thorough picture could be achieved about the system. However, even with figures such as interaction success rate, number of dialogue turns, number of help requests, the system developer has difficulties to see in on the task level where exactly things go wrong.

There are numerous works on evaluation of spoken dialogue systems and most of the joint EU activities in Europe (SUNDIAL, DISC, EAGLES, ARISE, etc.) and (D)ARPA funded projects in the USA are/were placing great emphasis on methods and metrics used for dialogue system assessment. Overall assessments are achieved by conducting both objective and subjective evaluations. In [2] a framework was presented to determine how subjective measures correlate with objectively derived figures. Kamm et al. in [3] show through examples how parallel evaluation of subjective and objective measures enable system designers to find weak points of the user interface and the actual implementation.

The work presented in [4] introduced a unified database structure to enable system developers to gather data from specific components of a system and organise them in a relational way. Organising, maintaining and visualising the data derived from a dialogue system helps in reproducing dialogue paths and evaluating system performance.

This paper places emphasis on the visualisation aspect of spoken dialogues. A method is introduced for explicit visualisation of dialogue paths logged in a spoken dialogue system. Task level indication of dialogue paths in a chart as a function of system-user turns enables designers to investigate system performance and interaction patterns in a statistical sense. The visualisation explicitly draws the dialogue paths, i.e. the successful or failed acquisition of semantic units present in the application and their corresponding verifications during an interaction. Spoken dialogue systems are considered here as over-the-telephone applications, nevertheless the same methods can also be adapted for non-telephony, for instance desktop- or device-based implementations.

Using the visualisation tool a measure can be defined that combines the success rate and the number of dialogue turns in a weighted average manner. Also, the proposed measure gives a view about the relative goodness of the system compared to the neutral case when the system drives the interaction and successfully understands and explicitly verifies all the semantic units.

The paper is organised as follows. In Section 2 the basics of the visualisation method is outlined with several examples for different interaction styles. Section 3 describes the derivation of the proposed measure followed by Section 4 showing data from an experimental system. The final section presents a summary of the paper with conclusions and directions of further development of the method.

2. A METHOD OF VISUALISATION

This section defines the formalism and elements used in the visualisation process by presenting several examples for different interaction styles. The method to be introduced enables the visualisation of various interaction styles (system- and user-driven, menu-based, explicit and implicit verification), use of barge-in, negations by the user, error situations, dialogues reaching the maximal number of trials, hang-up cases and the number of times a certain dialogue path is taken in a statistical sense. All these are clearly visible from a chart as examples will show.

2.1 Definitions

System-user turns. A dialogue between the system and the user consists of turns. A turn is defined here as one pair of system-user utterance, i.e. a system prompt (question, error or help announcement) followed by a user input, which is purely verbal in our case. The entire length of the dialogue can be measured either in relative time or in terms of turns. For the sake of easy interpretation turns are used in the visualisation chart.

Semantic units. Other elements of a dialogue are the semantic units which represent meaningful entities used for task completion. These semantic units are typically empty in the beginning of the interaction, unless the system designer wants them to be filled up with a default value (e.g. *date* can be assigned with the current date). The definition of a task depends on the actual domain and the application itself, nevertheless, a task can only be performed only if all the semantic units are filled up by the user. Taking a classical example, in a timetable inquiry system the minimal set of semantic units are *departure, destination, date* and *time*. In the visualisation chart semantic units are indicated with *SU_n* where *n* is an integer non-zero number.

Verification. Before executing the task (e.g. doing a reservation, fetching data from a timetable, etc.), verification of semantic units are often necessary depending on the severity of the action to be carried out. In most of the cases explicit verification is applied and it is rather straightforward to implement. Handling explicit verification is similar to the acquisition of a semantic unit. In the proposed visualisation method explicit verification is included on the semantic unit level. *VF_n* represents the verification item for the semantic unit *SU_n*. If the speech recogniser provides a confidence measure along with the recognised word/utterance, the dialogue manager can decide whether verification is necessary or not. Applying a two-level threshold scheme for confidence measures can accelerate the interaction considerably: if the measured confidence is higher than the upper level threshold the verification of the semantic unit is not necessary; if it is below the lower threshold, the recognition result is rejected and the user is asked once more to give the required semantic unit; verification is invoked only if the confidence measure is in the middle range. In the visualisation a verification item is represented as *(VF_n)* if it is allowed to be skipped due to a high enough confidence measure for *SU_n*.

Using implicit verifications in a dialogue is more problematic. In these cases the system intends to acquire a semantic unit while in the same prompt an already understood semantic unit is stated (e.g. "When do you want to travel to Helsinki?"). Difficulties of using implicit verification in the train timetable inquiry domain were pointed out by Sturm et al. in [5]. Evaluation of their systems in terms of success rate and dialogue turns in relation to the applied interaction and verification strategies indicates the necessity and importance of complex considerations. Visualisation could provide additional insight into identifying such problems in the implementation.

Hang-up cases. Hang-up is one very important non-verbal user reaction when the user's response to a system output, e.g. to a wrongly recognised name, is the immediate interruption of the call. This user reaction is also indicated in the visualisation chart as one possible termination.

Maximum number of trials. It is up to the system designer how many unsuccessful user trials are allowed within an application. The designer can specify beforehand that it is not worth to continue the speech-enabled interaction if the number of trials, say, reaches 3. After that transferring the call to a human operator seems to be a reasonable choice. In the visualisation process this can be indicated by limiting the active area of the chart. In the figures below any dialogue path ending in the shaded area is counted as unsuccessful or as a transferred-to-an-operator call.

2.2 Examples

System-driven mode. This is the interaction type which can be visualised in the most straightforward way. In Figure 1 below dialogue paths are depicted. A dialogue path is defined as the track of the interaction: at what system-user turn the system interprets the user's input as one of the semantic units, verifications or hang-up. On the *y* axis the semantic units and their corresponding verifications present in the application are plotted. The term semantic unit is used even if there is no speech understanding module in the application. The semantic units and their corresponding verifications are indicated on the *y* axis according to the order as the system request them in system-driven mode. The hang-up case is also present in the bottom of the *y* axis. On the *x* axis the number of system-user turns are indicated.

The *ideal* interaction from the user's point of view is the shortest. In this case the system recognises all the required semantic units in one step without any verification. In Figure 1 an ideal path is depicted with a thin line. It begins from the (*0,Start*) point of the chart and ends on the upper edge of the chart with system-user turn equals to *1*. In the example below the verification can be skipped, indicated by (*VF_1*), if high enough confidence is measured for *SU_1*.

A *neutral* interaction is depicted with a thick line in Figure 1. The neutral naming convention is used since in this case the user does not achieve much acceleration with the system. Here the system first asks for *SU_1* then for its verification. This is a diagonal path, starting from (*0,Start*). After the acquisition of the semantic unit the path arrives to (*1,SU_1*) and after successful verification ends at (*2,(VF_1)*) on the upper edge. The ideal interaction is faster than the neutral one and any other path longer than the neutral is due to subsequent trials.

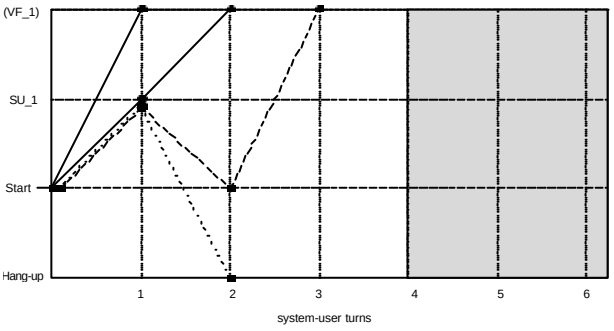


Figure 1: Example dialogue paths for system-driven case

If the user *rejected* the recognised item in the verification stage, the dialogue path from (*1,SU_1*) turns down to (*2,Start*) and the system starts for the second time to ask for *SU_1*. This case is indicated with a dashed line. Negations are straightforward to count in the visualisation chart since they are indicated with those descending arcs which do not end on the lower edge.

The *hang-up* case is also plotted in the chart in a way that the path ends on the lower edge of the figure. A hang-up case can occur any time during the dialogue and is an important measure when systems are analysed. In the example below the path resulting in hang-up is plotted with a dotted line.

Visualisation of barge-in. The interaction time can be shortened and thus more natural dialogues can be allowed by utilising barge-in. Users prefer to talk over system prompts even though they are not aware of the barge-in capability of the system [7]. Usage of barge-in can be straightforwardly displayed in the visualisation chart. Figure 2 shows some examples. The grids of the sample dialogue paths (thin and dotted lines) are placed for the *SU_1* item backwards from the (*1*,*SU_1*) point. The distances are inversely proportional to the elapsed time between the beginning of the system prompt and the instance the barge-in occurs. The same displaying method can be seen for the verification stages close to (*1*,*(VF_1)*) and (*2*,*(VF_1)*). Barge-in cases are to be plotted for each system-user turn independently, in other words, the acceleration gained in one step is not accumulated to the next dialogue step.

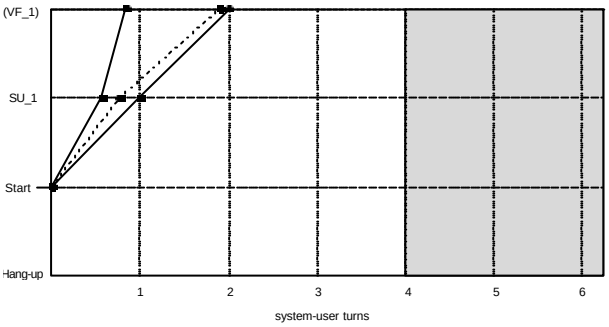


Figure 2: Indicating barge-in for individual dialogue paths

Menu-based interaction. Although user interfaces with menu-based implementation might represent rigid interaction styles, nevertheless they are preferred due to high task completion rate. The users are offered to choose exactly one item from a menu. The selection of an item is triggered by the recognition of a single keyword or phrase. In more complex implementations a grammar is applied where the keywords or phrases are embedded in natural expressions. Depending on the chosen menu item the interaction continues with a sub-dialogue.

If a menu at a stage of the dialogue is considered as a semantic unit, then the value of the semantic unit can be any of the possible items offered by the menu. In the visualisation chart these values, i.e. the instantiation of possible menu items, are also drawn on the *y* axis. They are indicated as *M_n* where *n* is a non-zero integer number. Above the individual menu items the sub-dialogue elements for the corresponding menu item are listed as in the earlier examples.

The example shown in Figure 3 has three menu items. *M_1* and *M_2* could represent, say, weather and movie information services, respectively. For the weather service a city must be specified with *M_1_SU* and for the movie information service a time must be given by *M_2_SU*. The exit from the system is implemented with the third menu item titled as *M_3*.

The selection of a menu item is indicated with a small circle on the corresponding grid. In the first example (thin line) the user chooses the movie service and arrives to (*1*,*M_2*). From here on the user is in the movie sub-dialogue and a system-driven interaction takes place. After successful completion the system prompts the user with the initial menu in (*2*,*Start*). Then the user chooses the exit menu item and the dialogue path ends on the upper edge in (*3*,*M_3*). A similar dialogue path is depicted with the thick line, however, here the user chooses the first menu item. In the dialogue path of this latter example the horizontal move indicates an error situation: the system could not interpret the user's response for some reason (e.g. no speech detected, timeout, OOV, etc.).

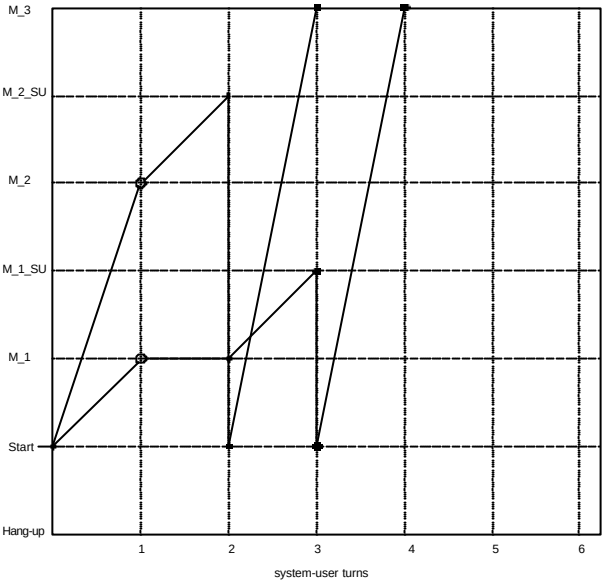


Figure 3: Visualisation of menu-based interactions

Implicit verifications and user-driven mode. These two cases can be handled in the same way. When implicit verification strategy is used, the semantic unit to be verified can be attached to a question, in theory, for any other semantic unit within a system prompt. This means that all the possible combination pairs of a semantic unit *SU_i* and a verification item *VF_j*, where $i \neq j$, must be indicated on the *y* axis.

For user-driven mode a similar grouping of semantic units can be applied, however, now all the possible combination pairs, triples, etc. are necessary to list on the *y* axis. In practice, most probably only pairs of semantic units are enough to indicate on the *y* axis. Only log information will reveal whether all the possible combinations occur in real usage or not.

3. A NOVEL MEASURE

This section introduces a measure derived from the above discussed visualisation chart. The rationale behind the here introduced measure is that in traditional success rate computation all the sessions are counted as successful if the goal of the application is reached, regardless of the dialogue length. In this sense all the sessions ending on the upper edge of the visualisation chart are successful. However, from qualitative point of view, these dialogues are not equal. Users prefer those dialogues which tend to be ideal ones, that is resulting in minimal interaction time.

With an averaged computation dialogues identical or close to the ideal interaction can be emphasised over the ones with longer interaction time. The following procedure is proposed:

- the number of dialogues ending at the end grid of the ideal path are weighted with 1;

- the number of dialogues which end as neutral ones in the end of the diagonal path are weighted with zero;
- the number of dialogues with more steps than *m*, the number of maximum trials, are weighted with –1;
- the number of dialogues ending between the neutral case and the one with *m+1* steps are weighted proportionally with a value between 0 and –1, respectively;
- the number of dialogues ending between the ideal case and the neutral one are weighted proportionally with a number between 1 and 0, respectively;
- the number of dialogues ending with hang-ups on the lower edge of the chart are weighted with –1;
- sum these weighted values and divide with the total number of interactions.

In this procedure the reference dialogue path is chosen to be the neutral case and weighed with zero. It is up to the system developer which dialogue path is chosen to be the reference one. The measure resulting from the above calculation will be between +1 and –1. Its interpretation is as follows: the closer the measure to the maximum value is, the more ideal the system performance. In case the measure is around zero the average interaction is close to the reference neutral case, thus the system does not provide much benefit compared to a slow and rigid question-answer dialogue. Near –1 measure indicates disastrous system performance with few successful but long dialogues.

4. EXPERIMENTS

In an earlier paper we presented usability results with two of our experimental name dialling systems [6]. The figures below are derived from a third one used within our laboratory in the past 10 months. In Figure 4 the occurrences of dialogue paths are indicated with proportional shading. The more times a dialogue path occurs, the lighter its colour. In this chart altogether 3877 calls are displayed (1567 ideal calls, 953 neutral ones, 159 ideal calls for the second trial, etc. and 1008 hang-ups). The above proposed measure with these data yields 0.11. This result can be interpreted as follows: the average dialogue paths is close to the reference neutral path, thus not much acceleration is achieved. There is still room for improvement to get closer to the ideal case when the user is connected to the required person without explicit verification.

Figure 4: Dialogue paths for a name dialling system

The traditional success rate is 74% in our example application, but it does not tell much alone. The average number of dialogue turns is 1.86 which indicate that in the average sense users experience a two-step dialogue with giving a name and explicitly verifying it.

5. CONCLUSIONS

A visualisation method to help evaluation of spoken dialogue systems was introduced. Displaying explicitly dialogue paths in a statistical sense is targeted as a supplementary tool for system designers to identify potentially problematic points of the implementation. The method enables the visualisation of various interaction styles and user actions. Furthermore, a measure derived directly from the visualisation chart was introduced. In this measure the number of successful dialogues is weighted with a value proportional to the length of the individual dialogues. The measure provides means for system developers to calculate an average dialogue path and compare it to the reference one (slow question-answer style) and the ideal one (everything is understood in one step).

The plan for the future includes evaluation of applications under development in our laboratory using the visualisation method. It is also interesting to see whether the new combined measure can be usefully applied in system development and how different systems can be compared.

Acknowledgement. The help and comments of the author's colleagues, Dari Trendafilov and Vétek Ákos, are kindly appreciated.

6. REFERENCES

1. Fraser N.M. & Simpson A., "Black Box and Glass Box Evaluation of the SUNDIAL System", Proceedings of 3rd European Conference on Speech Communication & Technology, Berlin, Germany, pp. 1423-1426., 1993.

2. Walker M., Litman D., Kamm C. & Abella A., "Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies", *Computer, Speech and Language*, pp. 317-347., 1998.

3. Kamm C., Narayanan S., Dutton D. & Ritenour R., "Evaluating Spoken Dialog Systems for Telecommunication Services", Proceedings of Eurospeech'97, Rhodes, Greece, pp. 2203-2206., 1997.

4. Chih-mei Lin, Narayanan S. & Ritenour R., "Database Management and Analysis for Spoken Dialog Systems: Methodology and Tools", Proceedings of Eurospeech'97, Rhodes, Greece, pp. 2199-2202., 1997.

5. Sturm J., den Os E. & Boves L., "Issues in Spoken Dialogue Systems: Experiences with the Dutch ARISE System", Proceedings of ESCA Workshop on Interactive Dialogue in Multi-Modal Systems, Kloster Irsee, Germany, pp.1-4., 1999.

6. Dobrin C., Boda P. & Laurila K., "On Usability of Name Dialling", Proceedings of ASRU'99, Keystone, Colorado, USA, 1999.

7. Heins R., Franzke M., Durian M. & Bayya A., "Turn-Taking as a Design Principle for Barge-In in Spoken Language Systems", *International Journal of Speech Technology 2*, Kluwer Academic Publishers, pp. 155-164., 1997.