Automated Wizard-of-Oz for Spoken Dialogue Systems

Giuseppe Di Fabbrizio *Gokhan Tur* *Dilek Hakkani-Tür*

AT&T Labs - Research
180 Park Avenue, Florham Park, NJ 07932 - USA
{pino,gtur,dtur}@research.att.com

Abstract

Designing and building natural language spoken dialogue systems require large amounts of speech utterances, which adequately represent the intended human-machine dialogues. For this purpose, typically, first a "Wizard-of-Oz" data collection is performed, and then the collected data is transcribed and labeled by expert labelers. Finally, the data is used to train both the speech recognizer and the spoken language understanding stochastic models. Data collection and labeling is an expensive and time consuming manual process. In this paper we propose a completely Automated Wizard, which is capable of recognizing and understanding application independent requests reusing the previously labeled and transcribed data from similar domains, and improving the informativeness of the collected data. We demonstrate that, in the context of automated call routing, compared to the existing data collection systems, the Automated Wizard better captures the user intentions and produces substantially shorter interactions resulting in a better user experience and a less intrusive approach.

1. Introduction

Natural language spoken dialogue systems (SDS) aim to identify user intents and take actions accordingly, to satisfy their requests. In this paper we consider automated call routing systems where the task is to classify the intent of the user, expressed in natural language, into one or more predefined call-types. Then, according to the utterance classification call-type, contextual information and other service parameters, the dialogue manager (DM) would decide on the next prompt or direct the call to the correct destination such as an automated Interactive Response System (IVR) or a specific call center live operator. As a call classification example, consider the utterance *I would like to know my account balance*, in a customer care banking application. Assuming that the utterance is recognized correctly, the corresponding intent or the call-type would be *Request(Account_Balance)*. The action would be to prompt for the account number and provide the account balance or route the call to the Billing Department.

Typically some initial task data are needed for designing such a system and determining the nature of the call-types. This data can then be used in training the automatic speech recognition (ASR) and the spoken language understanding (SLU) classifier models to bootstrap the initial version of the system. Since human-human interactions are very different from human-machine interactions in terms of style, language and linguistic behavior [5][7], initial data is collected via "Wizard-of-Oz" systems. In such systems, the users only interact with a hidden human agent who simulates the behavior for the system in a way that the caller believes he is interacting with the real system. The amount of data required to properly capture the caller's naturally expressed intentions varies and depends on the application domains. Best practice in the natural language service field suggests thousands of utterances to be collected and labeled to bootstrap a system in order to have a reasonable ASR and SLU coverage. In these real-world service scenarios, Wizard systems tend not to scale in terms of cost and time required to complete the initial data collection.

For routing applications, where the user intentions are typically expressed in the first few turns of the dialogue, a simpler approach, called "Ghost Wizard", has been used in the AT&T natural language data collections without requiring a human 'behind the curtains'. In that case, the initial system greets the users and records one or two user's responses. Although a Ghost Wizard approach scales better for large data collections, blindly recording the caller's utterances does not keep in consideration speech recognition problems that may occur with a real system. Furthermore, the Ghost Wizard does not handle generic discourse illocutionary acts like vague questions, greetings, thanks and agreements that have little or no relevance for the actual service task. Also in cases where the user has a specific request in the first turn, the Ghost Wizard may result in user annoyance when it asks the user for the intent the second time.

In this paper, we propose a completely Automated Wizard, which is capable of recognizing and understanding application independent requests by reusing the previously labeled and transcribed data and providing some level of dialogue repairs to effectively encourage the users to express the reason of the call to the machine. This improves the informativeness of the data collected, since now we would collect more task-specific intents which are crucial for the design of the application. Furthermore, it uses general ASR and SLU models, which means a system is deployed for the application from the very first day, so it is straightforward to replace this with improved task specific DM, ASR and SLU models.

The organization of this paper is as follows: Section 2 describes briefly the AT&T SDS, which we use in this study, and its main components, ASR, SLU, and DM. In Section 3 we present our method to employ a smarter Ghost Wizard. Section 4 presents our experiments using real data from a customer care application.

2. AT&T Spoken Dialogue System

The AT&T SDS is part of the AT&T VoiceTone® service available to the AT&T's business customers [1]. At a very high level of abstraction, the AT&T SDS is a phone-based VoiceXML [12] system with specific extensions to address natural language needs. Typically, once a phone call is established, the dialogue manager prompts the caller and activates the top level ASR grammar. The caller's speech is then transcribed and sent to the SLU which replies with a semantic representation of the utterance. Based on the SLU reply and the implemented dialogue strategy, the DM engages in a mixed initiative dialogue to drive the user towards the

goal. The DM iterates the previously described steps until the call reaches a final state (e.g., the call is transferred to a customer service representative (CSR), an IVR or the caller hangs up).

In the Automated Wizard case, automatic utterance recording is enabled during the speech recognition process, where the ASR is also responsible for end pointing the caller's utterance boundaries. The dialogue trace is logged into the system using specific dialogue markers to easily identify the execution of the dialogue. A brief description of the system components is provided in the next sections.

2.1. Automatic Speech Recognition

Robust real-time speech recognition is a critical component of a spoken dialogue system. The speech recognizer uses trigram language models based on Variable N-gram Stochastic Automata (VNSA) [2]. The acoustic models are subword unit based, with triphone context modeling and variable number of Gaussians (4-24) [11]. The output of the ASR engine (which can be the best word string or a lattice) is then used as the input of the SLU component.

2.2. Spoken Language Understanding

In a natural spoken dialogue system, the definition of "understanding" depends on the application. In this work, we focus only on goal-oriented call classification tasks, where the aim is to classify the intent of the user into one of the predefined call-types [3]. Classification is employed for all utterances in all dialogues as seen in the sample dialogue in Figure 2. Thus all the expressions the users can utter are classified into pre-defined call-types (e.g., *Request (Account_Balance)*) before starting an application. Even the utterances which do not contain any specific information content get a special call-type (e.g., *Hello*). In this study, we have used an extended version of a Boosting-style classification algorithm for call classification and used word n-grams as features.

2.3. Dialogue Manager

In a mixed-initiative spoken dialogue system, dialogue management is the key component responsible for the human-machine interaction. The DM keeps track of the specific discourse context and provides disambiguation and clarification strategies when the SLU call-types are ambiguous or have associated low confidence scores [6]. It also extracts other information from the SLU response, such as the named entities, in order to complete the information necessary to provide a service.

3. Wizard-of-Oz Approaches

3.1. Ghost Wizard

In the literature, in order to determine the application-specific call-types, first a "wizard" data collection is performed [4]. In this approach, a human, i.e. wizard, acts like the system, though the user of the system does not know about this. It is also possible to implement a "Ghost Wizard" approach, where no human is needed. The open ended *How may I help you?* prompt first welcomes the users, and after the first or second turn directs the call to a human agent. Both methods turned out to be better than recording user-agent (human-human) dialogues, since the responses to machine prompts are significantly different than responses to humans, in terms of language characteristics and linguistic behavior [5][7]. Figure 1 presents an example dialogue from a ghost wizard system. The first and second prompts are always the same independent of what the user says. Then the caller is always directed to a human.

- **System:** How may I help you?
- **User:** I'd like to know what the interest rate is at right now
- **System:** Sorry, I could not get that, how may I help you?
- **User:** I need information about rates

Figure 1. An Example of a natural language dialogue with a Ghost Wizard

It is evident that the second turn is not necessary. The initial intent was expressed already, so, in the subsequent interaction, the annoyed caller provides a shorter and simpler reply probably assuming that the system is not really capable of understanding him. A real system would be able to address correctly the request without further clarifications. The second utterance is a ghost wizard artifact and should not be included in the training set to avoid alteration of the natural distribution of the call-types.

3.2. Automated Wizard-of-Oz

The proposed Automated Wizard does not handle task specific requests, since there is no labeled data to train it, but it can handle task independent cases, such as requests for talking to a human agent, and discourse call-types, such as *Hello*. Figure 2 presents a dialogue which is more useful while designing the application and training the ASR and SLU models.

- **System:** How may I help you?
- **User:** Hello?
- *Discourse Call-type: Hello*
- **System:** Hello, how may I help you?
- **User:** I need to talk to somebody.
- *Task-independent Call-type: Request(Call_Transfer)*
- **System:** You can go ahead and speak naturally to me. Just briefly tell me how I may help you …
- **User:** I need to know my account balance.
- *Task-specific Call-type: Request(Account_Balance)*

Figure 2. An Example of Natural Language Dialogue with the Automated Wizard

In this example the Automated Wizard first plays the prompt "*How may I help you?*", then the user's response is recognized by the application independent ASR, and the ASR output is classified using the SLU. If the SLU class is not understood with high enough confidence or understood as "task specific", then the aim of wizard data collection is realized, so the customer is transferred to a CSR without further hassle. For example for the dialog in Figure 1, the user would be transferred to the CSR after the first turn. If the SLU class is "generic", this means that the user either uttered a discourse sentence, such as "Hello", or asked for an application independent request, such as a vague intent (e.g., "*I have a question*") or a request to speak to a live customer service representative. In the final two cases, the user is re-prompted, and the dialogue can proceed for several more turns. As seen in the example dialogue in Figure 2, this ensures more informative data collection than the ghost

wizard approach, with less hassle to the user.

This approach also eliminates the cost of the Wizard-of-Oz approach, as there is no human involved. It can also be used to improve the Wizard-of-Oz approach, by allowing the customer service representative to hear the actual user's request playing back the utterance during the transfer.

The system also re-prompts the user if the intent is not classified with a satisfactory confidence, the user was silent, the ASR rejected the utterance or other irrelevant discourse call-types were detected. As a result, the Automated Wizard optimizes data collection by eliminating uninformative utterances from transcription or labeling.

In our previous complementary work [9], we have proposed reusing previously transcribed and labeled data to bootstrap spoken dialog systems.

One other work in the area of error handling strategies is described in [10]. Although this work introduces a real ASR to report the recognized utterance to the human Wizard, the final dialogue strategy decisions are biased by a human instead of relying upon a fully functional SLU. In our case, the DM introduces an error recovery strategy based on ASR and SLU rejections which is very close to the actual final system behavior.

4. Experiments And Results

4.1. Ghost and Automated Wizards Comparison

To characterize and compare the performances of the new Automated Wizard with the existing system, an experiment was conducted using a Ghost Wizard-collected corpus from a financial domain application with two dialogue turns. 11,653 utterances were transcribed, labeled and used to simulate the same task in the Automated Wizard environment. 5,891 of them are from the first turn, and 5,762 from the second turn. The difference is due to the hang-ups of the users after the first turn. The average number of words per utterance is 10.4. In this application, there are 36 call-types; 11 of them are generic, covering 37% of the test set. The speech recordings have been sent to the generic ASR and the resulting recognized string has been passed to the wizard SLU classifier. The resulting call-types were finally used to simulate the dialogue interaction.

To bootstrap a statistical language and an acoustic model for ASR, we used human-machine utterances from previous spoken dialogue data. ASR word accuracy is found to be 72.1% on this corpus.

According to the human labels, 45% of the users responded to the first prompt with a generic utterance, whereas this number reduces to 17% in the second turn. The rest of the users in both turns responded with a task specific intent. 70% of the users, who responded to the first prompt with a generic utterance, responded to the second prompt with a task specific utterance. This is actually an upper bound performance for the Automated Wizard. If the wizard can correctly detect the 55% of the users who responded to the first prompt with a task specific request, then we can directly transfer those users to the representative without further hassle.

The SLU classifier model is trained using human-machine utterances from existing natural language applications from both telecommunication and health insurance domains. The application specific call-types are mapped to a single call-type called "specific" whereas the generic (e.g., application independent) call-types are kept as is.

Figure 3 and Figure 4 show the detailed distribution of the call-types assigned by the SLU when the ASR output and the manually transcribed utterances are used respectively. Generic requests are further divided into: CSR, discourse (No Info), and vague questions (Vague). Rejections, due to low confidence classifications or silence timeouts, are included in the 'reject' label. According to the actual Automated Wizard call flow, the user may get prompted a third time if a sequence reject-vague (or vice versa) was detected, but since those are very rare occurrences, we limited our evaluation to the first two turns. The diagrams show comparable performances in both transcribed and automatically recognized output, possibly due to the compensating effect of the SLU.

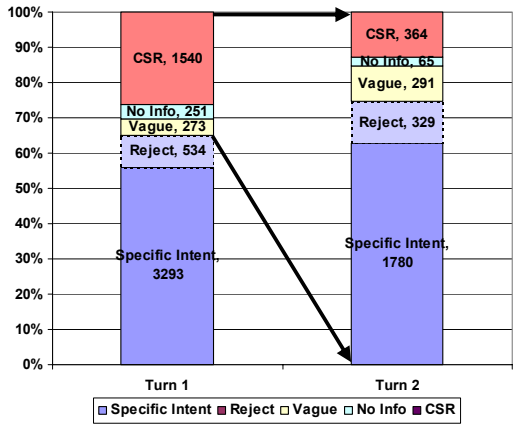

Figure 3. SLU Call-types Distribution with ASR output

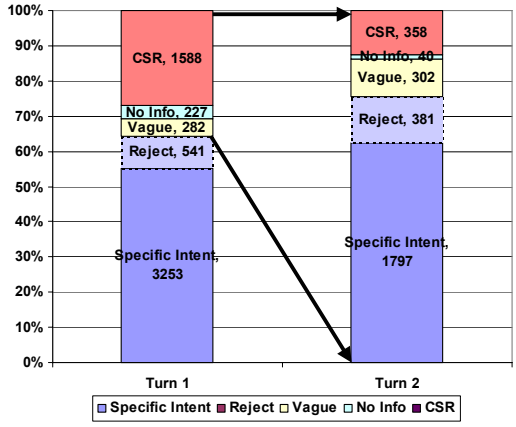

Figure 4.SLU Call-types Distribution with Transcribed Utterances

As presented in Figure 3 and Table 1, the SLU of the Automated Wizard classifies 65% of the first turn utterances as *specific* call type and *rejections,* using a rejection threshold of 0.3. In this case rejections are intuitively considered domain specific call types since the generic classifier reported a low classification confidence. The accuracy of this set is found to be 70%. This implies a truthful reduction of 65% x 70% = 45.5% of the need for the second dialogue turn. This indicates the ratio of callers transferred to a CSR after the first turn since they have already expressed a task specific intent. The accuracy for the first dialogue turn utterances classified as *generic* call type is 72%. The errors made for the specific utterances can be considered as missed chances, whereas the

errors for the generic call-types can result in re-prompting the user who has already expressed a task specific intent. For the Automated Wizard, the missed chances are more important than re-prompts, since all users are re-prompted in the Ghost Wizard approach. Our aim is to reduce re-prompts without increasing the missed chances to collect task specific, informative utterances.

	Turn 1		*Turn 2 (all)*		*Turn 2 (gen)*	
	Ratio	*Acc.*	*Ratio*	*Acc.*	*Ratio*	*Acc.*
Generic (no info+vague+CSR)	35%	72%	20%	30%	28%	31%
Specific+Rejections	65%	70%	80%	93%	72%	94%

Table 1. SLU accuracies for generic and specific call-types, for the first and second turns using ASR output

When all the utterances in the second turn are considered, the SLU classifies 80% of them as task specific with an accuracy of 93%. On the other hand, the accuracy for the utterances classified as generic is only 30% mainly due to out-of-domain utterances.

When we consider only the second turn utterances of the users, whose responses to top prompt are found to be generic, we see that 72% of them are now task specific with an accuracy of 94%. The accuracy for the generic utterances is 31%.

4.2. Automated Wizards Comparison

A further experiment briefly compares data collections in two substantially different domains. The Automated Wizard was used for two data collections and recorded around 10,000 dialogues for each campaign during a two week period. Real time reporting monitored the progress of the collection summarizing total number of calls, partitioning the utterances by call-type and identifying the uncollaborative calls based upon the calls routed without classified user's intent. Table 2 shows a comparison of the two data collections. Wizard 1 has been used in the consumer retail domain, while Wizard 2 collected data for a telecommunication customer care service. Utterances are longer (33 sec) in the second application due to the more articulated requests from the callers. The first application was simpler in terms of number and varieties of requests. Few call-types covered most of the Wizard 1 needs. This can be seen from the table where only 1.72% of the requests at the first turn were generic and 77.54% were recovered by the second turn. The second application turned out to be more complex and broader in scope. Almost 30% of the initial requests were generic and 46.83% provided application specific intents in the second turn.

	Wizard 1	Wizard 2
Average Turns Duration (sec)	24	33
Generic Intent 1st turn	1.72%	29.58%
Specific Intent 2nd turn	77.54%	46.83%

Table 2. Automated Wizard Data Collection Comparisons

In both cases the duration of the calls was rather short, most of the calls were limited to a single turn interaction, and the resulting data collected were focused more on the application specific tasks.

5. Conclusions

In this paper, we have presented an Automated Wizard-of-Oz data collection approach for spoken dialogue systems. We have used generic ASR and SLU models to recognize utterances with legitimate user intentions and reject or re-prompt user's queries with vague or undefined content. The Automated Wizard provided an overall 72.1% word accuracy on a specific task domain with its generic ASR language model and more than 70% SLU accuracy in classifying generic and specific call-types in the first dialog turn. Compared to the simple "Ghost Wizard" approach, this reduces the length of the interaction substantially with the callers resulting in a better user experience. In the second dialog turn the SLU accuracy increased to 93% for the specific call-types, while it dropped to 30% for the generic ones. We have also compared the performances between two different domains showing that more complex applications tend to capture the user's request in the second turn of the dialogue.

6. References

[1] M. Gilbert, J. G. Wilpon, B. Stern, G. Di Fabbrizio, "Virtual Agents for Contact Center Automation", *IEEE Speech Processing Magazine*, to Appear.

[2] G. Riccardi, R. Pieraccini, and E. Bocchieri, "Stochastic Automata for Language Modeling" *Computer Speech and Language*, vol. 10, pp. 265–293, 1996.

[3] N. Gupta, G. Tur, D. Hakkani-Tür, S. Bangalore, G. Riccardi, M. Rahim, "The AT&T Spoken Language Understanding System", *IEEE Transactions on Speech and Audio Processing*, to Appear.

[4] E. Ammicht, A. L. Gorin, and T. Alonso, "Knowledge Collection for Natural Language Spoken Dialog Systems", in *Proceedings of the Eurospeech*, Budapest, Hungary, September 1999.

[5] A Jönsson and N. Dahlbäck, "Talking to a Computer is not Like Talking to Your Best Friend", *Proceedings of the 1st Scandinavian Conference on Artificial Intelligence*, Tromsø, Norway, March 9-11, 1988.

[6] G. Di Fabbrizio and C. Lewis, "Florence: a Dialogue Manager Framework for Spoken Dialogue Systems", *Proceedings of the 8th International Conference on Spoken Language Processing*, Jeju, Jeju Island, Korea, October 4-8, 2004.

[7] G. Riccardi and A.L. Gorin, "Spoken Language Adaptation over Time and State in a Natural Spoken Dialog System", *IEEE Transactions on Speech and Audio*, vol. 8, pp. 3-10, Jan. 2000.

[8] G. Skantze, "Exploring Human Error Handling Strategies: Implications for Spoken Dialogue Systems", *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, Chateau-d'Oex-Vaud, Switzerland, August 28-31, 2003.

[9] G. Di Fabbrizio, G. Tur, D. Hakkani-Tür, "Bootstrapping Spoken Dialog Systems with Data Reuse", *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, Cambridge, Massachusetts, USA, April 30 - May 1, 2004.

[10] A. L. Gorin, G. Riccardi, and J. H. Wright, "How May I Help You", *Speech Communication*, vol. 23, pp. 113-127, 1997.

[11] V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tür, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi and M. Saraclar. "The AT&T Watson Speech Recognizer", *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Philadelphia, PA, USA, March 18-23, 2005

[12] "Voice Extensible Markup Language (VoiceXML) Ver. 2.0", *W3C Recommendation 16 March 2004*, http://www.w3.org/TR/2004/REC-voicexml20-20040316/