Experiences of an In-Service Wizard-of-Oz Data Collection for the Deployment of a Call-Routing Application

Mats Wirén,[1] Robert Eklund,[1] Fredrik Engberg[2] and Johan Westermark[2]

[1]Research and Development
TeliaSonera
SE–123 86 Farsta, Sweden

[2]Customer Integrated Solutions
TeliaSonera
SE–751 42 Uppsala, Sweden

firstname.lastname@teliasonera.com

Abstract

This paper describes our experiences of collecting a corpus of 42,000 dialogues for a call-routing application using a Wizard-of-Oz approach. Contrary to common practice in the industry, we did not use the kind of automated application that elicits some speech from the customers and then sends all of them to the same destination, such as the existing touch-tone menu, without paying attention to what they have said. Contrary to the traditional Wizard-of-Oz paradigm, our data-collection application was fully integrated within an existing service, replacing the existing touch-tone navigation system with a simulated call-routing system. Thus, the subjects were real customers calling about real tasks, and the wizards were service agents from our customer care. We provide a detailed exposition of the data collection as such and the application used, and compare our approach to methods previously used.

1 Background and introduction

Spoken-dialogue systems for applications such as customer care increasingly use statistical language models (SLMs) and statistically-based semantic classification for recognition and analysis of utterances. A critical step in designing and deploying such a system is the initial data collection, which must provide a corpus that is both representative of the intended service and sufficiently large for development, training and evaluation.

For at least 20 years, Wizard-of-Oz methodology has been regarded as a superior (though not unproblematic) method of collecting high-quality, machine-directed speech data in the absence of a runnable application.[1] Normally, these data will be useful for several purposes such as guiding dialogue design and training speech recognizers. Still, the Wizard-of-Oz option is often dismissed in favour of simpler methods on the ground that it does not scale well in terms of cost and time (for example, Di Fabbrizio et al. 2005). Consequently, Wizard-of-Oz has typically been used for data collections that are more limited in the number of subjects involved or utterances collected. One exception from this is the data collection for the original AT&T "How May I Help You" system (Gorin et al. 1997; Ammicht et al. 1999), which comprised three batches of transactions with live customers, each involving up to 12,000 utterances. Other well-known instances are "Voyager" (Zue et al. 1989) and the individual ATIS collections (Hirschman et al. 1993) which involved up to a hundred subjects or (again) up to 12,000 utterances.

While it is true that Wizard-of-Oz is a labour-intensive method, the effort can often be motivated on the ground that it enables significant design and evaluation to be carried out before implementation, thereby reducing the amount of re-design necessary for the actual system. However, one should also bear in mind the crucial advantage brought about by the possibility in a production environment of running the Wizard-of-Oz collection *in-service* rather than in a closed lab setting. As we shall discuss, the fact that real customers with real problems are involved instead of role-playing subjects with artificial tasks circumvents the key methodological problem that has been raised as an argument against Wizard-of-Oz, namely, lack of realism.

[1] For backgrounds on Wizard-of-Oz methodology, see Dahlbäck et al. (1993) and Fraser & Gilbert (1991).

Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, pages 56–63,
NAACL-HLT, Rochester, NY, April 2007.

The aim of this paper is to describe our experiences of running a Wizard-of-Oz collection in a production environment with real customers, with the double purpose of guiding dialogue design and collecting a sufficient amount of data for the first training of a speech recognizer. We also review what other options there are for the initial data collection and compare our Wizard-of-Oz approach with those.

The rest of this paper is organized as follows: Section 2 describes the call-routing problem and our particular domain. Section 3 gives an overview of the options for the initial data collection and the major trade-offs involved in selecting a method. Section 4 describes the application that was developed for our Wizard-of-Oz data collection, whereas Section 5 describes the actual data collection, summary statistics for the collected data and some experimental results obtained. Section 6 contains a discussion of our overall experiences.

2 The call-routing task and domain

Call routing is the task of directing a caller to a service agent or a self-serve application based on their description of the issue. Increasingly, speech-enabled routing is replacing traditional touch-tone menues whereby callers have to navigate to the appropriate destinations.

The domain of interest in this paper is (the entrance to) the TeliaSonera[2] residential customer care in Sweden, comprising the entire range of services offered: fixed and mobile telephony, broadband and modem-based Internet, IP telephony, digital television, triple play, etc. Around 14 million calls are handled annually, and before the speech-enabled call-routing system was launched in 2006, touch-tone navigation was used throughout. The speech-enabled system involves an SLM-based speech recognizer and a statistically-based classifier.[3] The task of the classifier is to map a spoken utterance to an application category which corresponds to a self-serve application, (a queue to) a human agent, a disambiguation category or a discourse category. Whereas self-serve applications and service agents are the desired goals to reach, disambiguation and discourse categories correspond to intermediate states in the routing dialogue. More specifically, disambiguation categories correspond to cases where the classifier has picked up *some* information about the destination, but needs to know more in order to route the call. Discourse categories correspond to domain-independent utterances such as greetings ("Hi, my name is John Doe"), channel checks ("Hello?") and meta questions ("Who am I talking to?"). Altogether, there are 124 application categories used by the current classifier.

[2] TeliaSonera (**www.teliasonera.com**) is the largest telco in the Scandinavian –Baltic region.

[3] The speech recognizer and classifier are delivered by Nuance (**www.nuance.com**).

3 Options for initial data collection

Basically, there are three options for making the initial data collection for a call-routing application: to collect human–human dialogues in a call center, to use an automated data-collection application, or to use a Wizard-of-Oz approach. We shall now describe each of these.

3.1 Human–human dialogues

The simplest possible approach to the initial data collection is to record conversations between service agents and customers in a call center. This is an inexpensive method since it does not require any data-collection application to be built. Also, there is no customer impact. However, the data obtained tend not to be sufficiently representative, for two reasons: First, typically only a subset of the services of a call center is carried out by human agents, and hence many services will not be covered. Second, the characteristics of human–human conversations differ from those of human–machine interaction. Still, this option has sometimes been preferred on the grounds of simplicity and lack of negative customer impact.

3.2 Automated applications

Due to the nature of the task, it is easy to put out a fully automated mock-up system in a live service that engages in the initial part of a call-routing dialogue. Typically, such a system will play an open prompt, record the customers' speech, play another prompt saying that the system did not understand, again record the speech, and finally direct all calls to a single destination, such as a general-skills service agent or the entry to the existing touch-tone menu. We estimate that a system of this kind could be implemented and integrated into a call center in about a person week. An example of this approach is the AT&T "Ghost Wizard" (referred to in Di Fabbrizio et al. 2005).

This basic approach can be improved upon by detecting silences and touch-tone events, and in these cases playing designated prompts that try to get the caller on track. Furthermore, if data from previous call-routing applications are available, it is possible to use these to handle domain-independent utterances. Such utterances correspond to discourse categories as mentioned in Section 2, and the idea then is to play prompts that encourage the caller to describe the issue. A description of such an approach is provided by Di Fabbrizio et al. (2005).

A problem with the automated approach is that customer impact can be quite negative, since the application does not actually do anything except for recording their speech (possibly through several turns), and routing them to a "dummy" destination where they will have to start over. Of course, one way of avoiding this is to include a human in the loop who listens to the customer's speech and then routes the call to the right destination. Apparently, this is the approach of Di Fabbrizio et al. (2005), which consequently is not fully automated.

Apart from customer impact, the problem with an automated system is that we do not learn the full story about caller behaviour. In particular, since typically only a minority of callers will state their issue in an unambiguous way within the given few turns, less information about the callers' actual issues will be obtained. In particular, for callers who completely fail to speak or who give no details about their issue, we will have no possibility of finding out what they wanted and why they failed. Furthermore, since the system lacks the ability to respond intelligently to in-domain utterances, no follow-up dialogue such as disambiguation can be collected.

3.3 Wizard-of-Oz

Although Wizard-of-Oz is arguably the best method for collecting machine-directed data in the absence of a running application, it is not without methodological problems. The basic critique has always been aimed at the lack of realism (for example, von Hahn 1986). In a thorough analysis, Allwood & Haglund (1992) point out that in a Wizard-of-Oz simulation, both the subjects and the wizard(s) are playing roles, occupied and assigned. The researcher acting as the wizard is occupying the role of a researcher interested in obtaining "as natural as possible" language and speech data, while playing the role of the system. The subject, on the other hand, is occupying the role of a subject in a scientific study, and playing the role of a client (or similar), communicating with a system while carrying out tasks that are not genuine to the subject, but given to them by the experiment leader (who might be identical with the wizard).

It turns out, however, that a traditional Wizard-of-Oz approach with made-up tasks according to a scenario is anyway not an option when collecting data for deploying a call-routing system. The reason for this is that we want to learn not just *how* callers express themselves, but also *what kind of tasks they have*, which obviously rules out pre-written scenarios. If the existing system uses touch-tone navigation, usually not too much can be ascertained about this, and trying to design a set of tasks just by looking at the existing destinations would miss the point.

By instead integrating a Wizard-of-Oz application in an existing, live service, we can circumvent the key methodological problems, while addressing all the problems of the previously described approaches and even obtaining some independent advantages:

1. Since the callers' experience will be like that of the intended application, albeit with human speech understanding, the customer impact will be at least as good. In fact, it is even possible to issue a kind of guarantee against maltreatment of customers by instructing the wizards to take over calls that become problematic (this is further discussed in Section 4).
2. Since real customers are involved, no role-playing from the point of view of the subjects takes place, and hence the data become highly realistic.
3. The fact that scenarios are superfluous—or even run counter to the goal of the data collection—means that the main source of methodological problems disappears, and that the data collection as such is considerably simplified compared to traditional Wizard-of-Oz.
4. By letting service agents be wizards, we move away even further from role-playing, given that the interaction metaphor in speech-enabled call routing is natural-language dialogue with a (general-skills) service agent.

5. Service agents possess the expertise necessary for a call-routing wizard: they know when additional information is required from the caller, when a call is ready for routing, and where to actually route the call. Hence, wizard guidelines and training become less complex than in traditional Wizard-of-Oz.[4]
6. Service agents have excellent skills in dealing with customers. Hence, during the data collection they will be able to provide valuable feedback on dialogue and prompt design that can be carried over to the intended application.

In spite of these advantages, Wizard-of-Oz appears to have been used only very rarely for collecting call-routing data. The sole such data collection that we are aware of was made for the original AT&T "How May I Help you" system (Gorin et al. 1997; Ammicht et al. 1999). The one disadvantage of the Wizard-of-Oz approach is that it is more laborious than automated solutions, mainly because several person months of wizard work is required. On the other hand, as we have seen, it is still less laborious than a traditional Wizard-of-Oz, since there are no scenarios and since wizard guidelines can be kept simple.

4 Data-collection application

Our data-collection application consists of two parts: The first part is the Prompt Piano Client (PPC), which is running on the service agent's PC. This is essentially a GUI with "keys" corresponding to prerecorded prompts by which the wizard interacts with the caller, thereby simulating the intended system. The PPC interface is shown in PLATE 1. The second part is the Prompt Piano Server (PPS), which is an IVR (interactive voice response) server with a Dialogic telephony board, running Envox, Nuance and Dialogic software. This handles playing of prompts as well as recording of calls. Two kinds of recordings are made: call logs (that is, the callers' speech events as detected by the Nuance speech recognizer) and complete dialogues ("open mic").

To set up a data collection, the contact center solution is modified so that a percentage of the incoming calls to the customer care is diverted to the PPS. The PPS in turn transfers each call to a wizard (that is, to a PPC) using tromboning. Allocation of the wizards is performed by the Telia CallGuide contact center platform using skill-based routing. Whenever a wizard answers a call, two audio streams are established, one from the customer to the wizard so that she can hear the customer's speech, and one from an audio source in the PPS to the customer. An initial open prompt is played automatically by the PPS, and the wizard is then free to start playback of prompts. This is realized by sending control messages from the PPC to the audio source on the PPS via TCP/IP, while listening to the customer throughout.

Depending on the caller's response, different things will happen: If the caller provides an unambiguous description of the issue, the wizard will transfer the call to the correct queue and end the recording by pressing the "end / route customer" button. This signals to the PPS that the call should be released using the Explicit Call Transfer (ECT) supplementary service, freeing the two channels used for the tromboned call in the PPS.

If, on the other hand, the caller does not provide an unambiguous description of the issue, the wizard will play a follow-up prompt aimed at getting more information from the caller by choosing from the buttons/prompts situated to the right (fields II and III of the GUI; see Plate 1). These parts of the GUI are fully configurable; the number and layout of buttons as well as the names of sound files for the corresponding prompts are declared separately. (Declarations include specifying whether the prompt associated with a particular button allows barge-in or not.) Thus, it is possible not just to vary individual prompts, but also to simulate call-routing dialogues to various depths by varying the number of buttons/prompts.

Apart from routing the call, a possible action of the wizard is to enter into the call. This is realized by establishing a two-way direct audio stream with the customer, enabling the parties to talk to each other. As pointed out in Section 3.3, one purpose of this is to let wizards take over calls that are problematic, thereby making sure that callers do not get maltreated during the data collection and reducing the risk that they hang up. A similar functionality was available in the data-collection application for AT&Ts "How May I Help You" system (Walker et al. 2000).

[4] Furthermore, as a side effect, it is possible to facilitate the subsequent process of manually tagging the data by keeping track of where each call is routed.

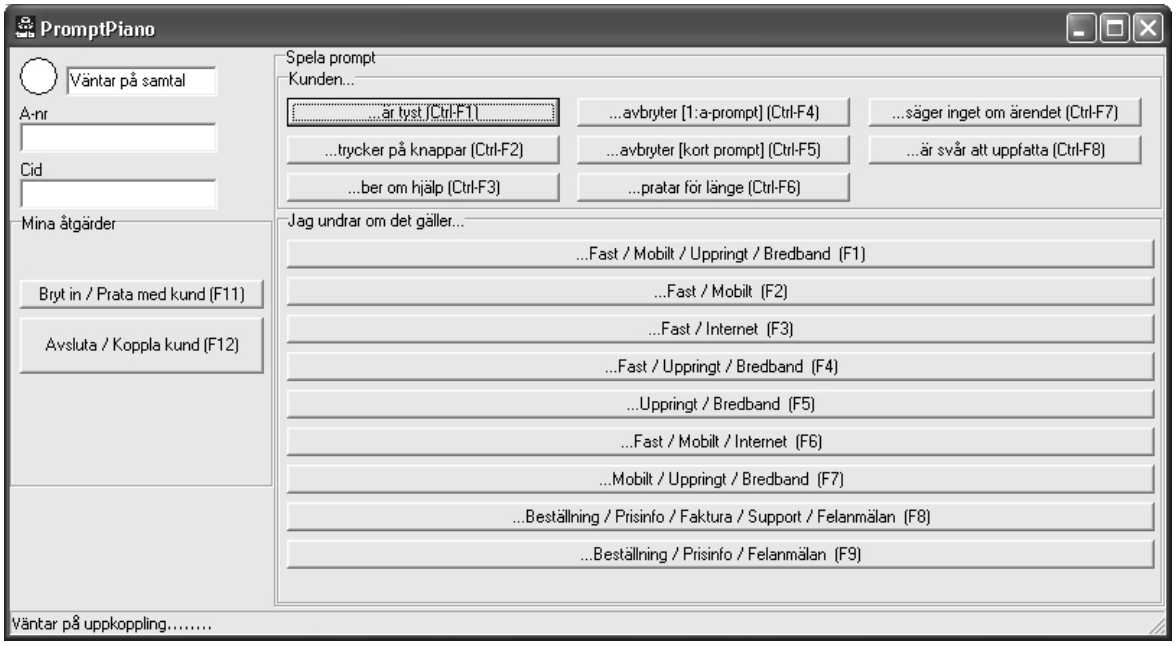

PLATE 1: The Prompt Piano Client interface as configured towards the end of the data collection. The interface is divided into three fields with buttons. *I:* The leftmost field provides caller information, like **A-nr** (the phone number the customer is calling from) and **Cid** (the phone number the customer provides as reason for the call). The wizard has two option buttons, **Mina åtgärder** ('my actions'), at hand: the button **Bryt in / Prata med kund** ('barge-in/talk to client') which is used for entering into the call, and the button **Avsluta / Koppla kund** ('end/route customer') which is used to terminate the recording prior to routing the call to the appropriate destination. (Both of these options are associated with prompts being played.) *II:* The second field, **Kunden…** ('the customer…'), contains buttons corresponding to renewed open prompts for the purpose of error-handling, **…är tyst** ('… is silent'), **…trycker på knappar** ('uses the touch-tone keypad'), **…ber om hjälp** ('asks for help'), **…avbryter** ('interrupts'), **…pratar för länge** ('talks for too long'), **…säger inget om ärendet** ('doesn't say anything about the reason for the call'), **…är svår att uppfatta** ('is hard to understand'). *III:* The third field, **Jag undrar om det gäller…** ('I would like to know if it is about…'), contains buttons corresponding to disambiguation prompts asking for additional information, e.g. whether the customer's reason for the call is about fixed ('fast') or mobile ('mobilt') telephony, broadband ('bredband') or something else. All buttons also have hot-key possibilities for agents who prefer this over point-and-click.

With the exception of the initial open prompt, the wizards have full control over when and in what order prompts are played and actions are executed. Thus, whereas an automated system will start playing the next prompt after an end-of-speech timeout typically within the range of 0.75–1.5 seconds, a wizard may decide to impose longer delays if she considers that the caller has not yet yielded the turn. On the other hand, the wizard may also respond more rapidly. Thus, the problem of response delays, which has sometimes had distorting impact in Wizard-of-Oz simulations, does not appear in our application (cf. Oviatt et al. 1992).

The PPS application was developed in the Envox graphical scripting language, which makes it possible to write event-driven applications controlled from an external source such as the PPC, and also supports Nuance call logging (for recording customer utterances) and Dialogic transaction recording (for recording entire conversations between two parties, in this case the customer and the PPS, or the customer and the wizard).[5] Design, implementation and testing of the Prompt Piano (PPC and PPS) took four person weeks.

The agents/wizards were involved in the development from the very start to ensure that the application (and in particular the GUI) was

[5] VXML was not used since it appeared that real-time control of an IVR from an external source would then have been more difficult to implement. Furthermore, VXML browsers generally have no support for features such as transaction recording during tromboned transfer and delayed invocation of the ECT supplementary service in conjunction with call transfer. Hence, in a VXML framework, additional components would be required to solve these tasks.

optimized according to their needs and wishes. The Prompt Piano GUI was reconfigured several times during the course of the data collection, both for the purpose of carrying out prompt-design experiments and in response to (individual or group) requests for changes by the agents/wizards.

5 Data collection

5.1 Overview

The purpose of the data collection was twofold: to obtain speech data that could be used for initial training of the speech recognizer, and to obtain data that could be used to guide dialogue design of the intended application. Thus, whereas the former only involved caller responses to open prompts, the latter required access to complete call-routing dialogues, including error-handling and disambiguation.

Organization. Ten wizards were used for the data collection. Initially, one week was used for training of the wizards and basic tuning of the prompts. This process required four person weeks (not all wizards were present all the time). After a break of three weeks, the data collection then went on for five weeks in a row, with the ten wizards acquiring around 42,000 call-routing dialogues. (This figure includes around 2,000 useable dialogues that were collected during the initial week.) This was more than had been anticipated, and much more than the 25,000 that had been projected as a minimum for training, tuning and evaluation of the speech recognizer. Thus, although 50 person weeks were used by the wizards for the actual collection, 32 person weeks would actually have been sufficient to reach the minimum of 25,000 dialogues. On average, 195 dialogues were collected per working day per wizard (mean values ranging from 117 dialogues per day to 317 dialogues per day; record for a wizard on a single day was 477).

Barge-in. Initially, barge-in was allowed for all prompts. However, it turned out to be useful to have one very short prompt with barge-in disabled, just asking the caller to state the reason for the call. The main usage of this was in cases where callers were repeatedly barging in on the system to the extent that the system could not get its message through.

Utterance fragments. As a consequence of, on the one hand, wizards having full control over when and whether to start playing a prompt and, on the other hand, the speech recognizer having a fixed end-of-speech timeout, it would sometimes happen that more than one sound file would be recorded between two prompts in the Nuance call logs. An example of this would be: “Eeh, I... I’m wondering whether... can you tell me the pricing of broadband subscriptions?”, where both of the two silent pauses would trigger the end-of-speech timeout. Although this constitutes a mismatch between data collection and final system, in practice this caused no problem: on the contrary, the sound files were simply treated as separate utterances for the purpose of training the speech recognizer, which means that the most informative fragment, typically at the end, was not lost. In addition, these data are potentially very valuable for research on turn-taking (in effect, intelligent end-of-speech detection).

Wizards entering into calls. The event of wizards taking over calls in order to sort out problematic dialogues occurred on average in 5% of the calls. The figure was initially a bit higher, presumably because the wizards were less skillful in using the prompts available, and because the prompts were less well-developed. As a side-effect of this, we have obtained potentially very valuable data for error-handling, with both human–machine and human–human data for the same callers and issues (compare Walker et al., 2000).

Post-experimental interviews. We also used the facility of letting wizards take over calls as a way of conducting post-experimental interviews. This was achieved by having wizards route the calls to themselves and then handle the issue, whereupon the wizard would ask the caller if they would accept being interviewed. In this way, we were able to assess customer satisfaction on the fly with respect to the intended system and even getting user feedback on specific design features already during the data collection.

5.2 Experiments

Several design experiments were run during the data collection. Here, we shall only very briefly describe one of them, in which we compared two styles of disambiguation prompts, one completely open and one more directed. As can be seen in TABLE 1, utterances following the open disambiguation prompt are on average 3.6 times longer than utterances following the directed prompt.

Prompt	Utterances and Words			Disfluency			Concepts					
	Utts	Words	Words /Utts	Disfl	Disfl /Utts	Disfl /Words	Concepts In	Concepts Out	DIFFs Total	DIFFs Change	DIFFS /Utts	DIFFS /Words
Directed	118	216	1.8	19	0.16	0.09	136	244	108	0	0.9	0.5
Open	121	791	6.5	72	0.6	0.09	144	248	122	18	1.01	0.15

TABLE 1. Summary statistics for the **directed prompt** ('I need some additional information about the reason for your call. Is it for example about an order, price information or support?'), and the **open prompt** ('Could you please tell me a little bit more about the reason for you call?') prompts. Totals and ratios are given for utterances/words, disfluencies and number of concepts acquired before the disambiguation prompt was played ("In") and after the customer had replied to the disambiguation prompt ("Out"). Also, ratios are given for number of concepts compared to number of utterances and words, as well as totals and ratios for the differences (DIFFs) between concepts in and concepts out, i.e., how many concepts you "win" by asking the disambiguation prompt.

Furthermore, in order to see to what extent these prompts also made callers provide more information, we manually tagged the transcribed utterances with semantic categories. Following the evaluation methodology suggested by Boye & Wirén (2007, Section 5), we then computed the difference with respect to "concepts" for utterances immediately following and preceding the two kinds of prompts.

Although the number of concepts gained is only slightly higher[6] for the open prompt (as a function of concepts per utterance), there are some palpable differences between the directed and the open prompt. One, shown in TABLE 1, is that there are no instances where an already instantiated concept (e.g. `fixedTelephony`) is changed to something else (e.g. `broadband`), while this happens 18 times following the open prompt. The other, not shown in TABLE 1, is that, following the directed prompt, one never "gains" more than one new concept, while there are 26 instances following the open prompt where the gain is two concepts, and even two instances where the gain is three concepts (which also means that one concept is changed).

Finally, when one analyses the syntactic characteristics following the two different types of prompts, there is an obvious shift from the telegraphic "noun-only" responses that amount to more than 70% of the directed prompt responses, to the responses following the open prompt, where 40% are complete sentences and 21% are noun phrases. Also, the syntax is more varied following the open prompt.[7]

6 Discussion

We claimed in Section 3.3 that by using an in-service Wizard-of-Oz data collection, we have been able to effectively overcome all problems of the alternative methods discussed there. A relevant question is then if there are any remaining, independent problems of the approach described here.

On the methodological side, there is clearly a certain amount of role playing left in the sense that service agents are acting as the system (albeit a system whose interaction metaphor is a service agent!). Interestingly, we noticed early on that the agents sometimes failed in properly simulating the intended system in one respect: Since they would often grasp what the caller wanted before he or she had finished speaking, they would start playing the next prompt so early that they were barging in on the caller. Thus, in their willingness to provide quick service, they were stepping outside of their assigned role. However, they soon learnt to avoid this, and it was never a problem except for the first few days.

Apart from this, the main disadvantage of Wizard-of-Oz collections clearly is the amount of work involved compared to the other methods. As we have seen, the Prompt Piano design and implementation took four person weeks, training of the wizards took another four person weeks, and collection of 25,000 dialogues required 32 person weeks—hence altogether 40 person weeks (although we actually used 50 person weeks, since we went on collecting more data). This could be compared with possibly a single person week required for the fully automated approach. The more

[6] However, the difference is not statistically significant, either using a *t* test (two-sampled, two-tailed: p=0.16 with equal variances assumed; p=0.158 equal variances not assumed) or Mann-Whitney *U* test (two-tailed: p=0.288).

[7] The distributions are, in descending order, for the **directed prompt**: Noun=85, Sentence=11, Yes/No=8, Noun Phrase=8, no response=3, Yes/No+Noun=2, Adverbial Phrase=1, Adjective Phrase=1; for the **open prompt**: Sentence=49, Noun Phrase=26, Noun=24, Verb Phrase=11, Adjective Phrase=5, Adverbial Phrase=2, no response=2, Yes/No=1, Interjection=1.

elaborate automated methods would come somewhere in between, also depending on whether a human agent is used for routing callers or not.

In the TeliaSonera case, the main desiderata favouring Wizard-of-Oz were highly representative data, no negative customer impact and need for early evaluation and design, particularly because this was the first deployment of natural-language call routing in Scandinavia. In other words, it was decided to accept a higher *initial* cost in return for reduced costs downstream, due to higher quality and less re-design of the implemented system.

It is impossible to quantify the downstream savings made by choosing Wizard-of-Oz since we have no baseline. However, one indication of the quality of the data is the initial performance of the classifier of the deployed system. (By "initial", we mean the period during which no data from the live system had yet been used for training or updating of the system.) In our case, the initial accuracy was 75%, using 113 application categories. We regard this as a high figure, also considering that it was achieved in spite of several new products having been introduced in the meantime that were not covered by the speech recognizer. The initial training of the speech recognizer and classifier used 25,000 utterances. As a comparison, when an additional 33,000 utterances (mostly from the live system) had been used for training, the accuracy increased to 85%.

Acknowledgements

Many colleagues have provided invaluable help and support throughout this project. Here we can only mention some of them: Johan Boye, Joakim Gustafson, Linda Bell, Fredrik Byström, Robert Sandberg, Erik Näslund, Erik Demmelmaier, Viktoria Ahlbom, Inger Thall and Marco Petroni. Last but not least we thank our skilled wizards: Christina Carlson, Marie Hagdorn, Gunilla Johannisson, Ana-Maria Loriente, Maria Mellgren, Linda Norberg, Anne Tärk, Mikael Wikner, Eva Wintse and Jeanette Öberg.

References

Allwood, Jens & Björn Haglund. 1992. Communicative Activity Analysis of a Wizard of Oz Experiment. Internal Report, PLUS ESPRIT project P5254.

Ammicht, Egbert, Allen Gorin & Tirso Alonso. 1999. Knowledge Collection For Natural Language Spoken Dialog Systems. *Proc. Eurospeech*, Budapest, Hungary, Volume 3, pp. 1375–1378.

Boye, Johan & Mats Wirén. 2007. Multi-slot semantics for natural-language call routing systems. *Proc. Bridging the Gap: Academic and Industrial Research in Dialog Technology. NAACL Workshop*, Rochester, New York, USA.

Dahlbäck, Nils, Arne Jönsson & Lars Ahrenberg, Wizard of Oz Studies — Why and How. 1993. *Knowledge-Based Systems*, vol. 6, no. 4, pp. 258–266. Also in: Mark Maybury & Wolfgang Wahlster (eds.). 1998. *Readings in Intelligent User Interfaces*, Morgan Kaufmann.

Di Fabbrizio, Giuseppe, Gokhan Tur & Dilek Hakkani-Tür. 2005. Automated Wizard-of-Oz for Spoken Dialogue Systems. *Proc. Interspeech*, Lisbon, Portugal, pp. 1857–1860.

Fraser, Norman M. & G. Nigel Gilbert. Simulating speech systems. 1991. *Computer Speech and Language*, vol. 5, pp. 81–99.

Gorin, A. L., G. Riccardi & J. H. Wright. 1997. How may I help you? *Speech Communication*, vol. 23, pp. 113–127.

von Hahn, Walther. 1986. Pragmatic considerations in man–machine discourse. *Proc. COLING*, Bonn, Germany, pp. 520–526.

Hirschman, L., M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky & E. Tzoukermann. 1993. Multi-Site Data Collection and Evaluation in Spoken Language Understanding. *Proc. ARPA Human Language Technology*, Princeton, New Jersey, USA, pp. 19–24 .

Oviatt, Sharon, Philip Cohen, Martin Fong & Michael Frank. 1992. A rapid semi-automatic simulation technique for investigating interactive speech and handwriting. *Proc. ICSLP*, Banff, Alberta, Canada, pp. 1351–1354.

Walker, Marilyn, Irene Langkilde, Jerry Wright, Allen Gorin & Diane Litman. 2000. Learning to Predict Problematic Situations in a Spoken Dialogue System: Experiments with How May I Help You? *Proc. North American Meeting of the Association for Computational Linguistics* (*NAACL*), pp. 210–217.

Zue, Victor, Nancy Daly, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni, Stephanie Seneff & Michael Soclof. 1989. The Collection and Preliminary Analysis of a Spontaneous Speech Database. *Proc. DARPA Speech and Natural Language Workshop*, pp. 126–134.