

# Evaluating Spoken Dialogue Processing for Time-Offset Interaction

David Traum, Kallirroi Georgila, Ron Artstein, Anton Leuski

USC Institute for Creative Technologies

12015 Waterfront Drive, Playa Vista CA 90094-2536, USA

{traum|kgeorgila|artstein|leuski}@ict.usc.edu

## Abstract

This paper presents the first evaluation of a full automated prototype system for time-offset interaction, that is, conversation between a live person and recordings of someone who is not temporally co-present. Speech recognition reaches word error rates as low as 5% with general-purpose language models and 19% with domain-specific models, and language understanding can identify appropriate direct responses to 60–66% of user utterances while keeping errors to 10–16% (the remainder being indirect, or off-topic responses). This is sufficient to enable a natural flow and relatively open-ended conversations, with a collection of under 2000 recorded statements.

## 1 Introduction

*Time-offset interaction* allows real-time synchronous conversational interaction with a person who is not only physically absent, but also not engaged in the conversation at the same time. The basic premise of time-offset interaction is that when the topic of conversation is known, the participants' utterances are predictable to a large extent (Gandhe and Traum, 2010). Knowing what an interlocutor is likely to say, a speaker can record statements in advance; during conversation, a computer program selects recorded statements that are appropriate reactions to the interlocutor's utterances. The selection of statements can be done in a similar fashion to existing interactive systems with synthetic characters (Leuski and Traum, 2011).

In Artstein et al. (2014) we presented a proof of concept of time-offset interaction, which showed that given sufficiently interesting content, a reasonable interactive conversation could be demonstrated. However that system had a very small

amount of content, and would only really work if someone asked questions about a very limited set of topics. There is a big gap from this proof of concept to evidence that the technique can work more generally. One of the biggest questions is how much material needs to be recorded in order to support free-flowing conversation with naive interactors who don't know specifically what they can ask. This question was addressed, at least for one specific case, in Artstein et al. (2015). There we showed that an iterative development process involving two separated recording sessions, with Wizard of Oz testing in the middle, resulted in a body of material of around 2000 responses that could be used to answer over 95% of questions from the desired target audience. In contrast, the 1400 responses from the first recording session alone was sufficient to answer less than 70% of users' questions. Another question is whether current language processing technology is adequate to pick enough appropriate responses to carry on interesting and extended dialogues with a wide variety of interested interactors. The proof of concept worked extremely well, even when people phrased questions very differently from the training data. However, that system had very low perplexity, with fewer than 20 responses, rather than something two orders of magnitude bigger.

In this paper, we address the second question, of whether time-offset interaction can be automatically supported at a scale that can support interaction with people who know only the general topic of discussion, not what specific content is available. In the next section, we review related work that is similar in spirit to time-offset interaction. In Section 3 we review our materials, including the domain of interaction, the system architecture, dialogue policy, and collected training and test data. In Section 4, we describe our evaluation methodology, including evaluation of speech recognition and classifier. In Section 5, we present our results,

showing that over 70% of user utterances can be given a direct answer, and an even higher percentage can reach task success through a clarification process. We conclude with a discussion and future work in Section 6.

## 2 Related Work

The idea for time-offset interaction is not new. We see examples of this in science fiction and fantasy. For example, in the Hollywood movie “I, Robot”, Detective Spooner (Will Smith) interviews a computer-driven hologram of a recently deceased Dr. Lanning (James Cromwell).

The first computer-based dialogue system that we are aware of, that enabled a form of time-offset interactions with real people was installed at the Nixon Presidential Library in late 1980s (Chabot, 1990). The visitors were able to select one of over 280 predefined questions on a computer screen and observe a video of Nixon answering that question, taken from television interviews or filmed specifically for the project. This system did not allow Natural language input.

In the late 1990s Marinelli and Stevens came up with the idea of a “Synthetic Interview”, where users can interact with a historical persona that was composed using clips of an actor playing that historical character and answering questions from the user (Marinelli and Stevens, 1998). “Ben Franklin’s Ghost” is a system built on those ideas and was deployed in Philadelphia from 2005–2007 (Sloss and Watzman, 2005). This system had a book in which users could select questions, but, again, did not use unrestricted natural language input.

What we believe is novel with our New Dimensions in Testimony prototype is the ability to interact with a real person, not an actor playing a historical person, and also the evaluation of its ability to interact naturally, face to face, using speech.

## 3 Materials

### 3.1 Domain

Our initial domain for time-offset interaction is the experiences of a Holocaust survivor. Currently, an important aspect of Holocaust education in museums and classrooms is the opportunity to meet a survivor, hear their story firsthand, and interact with them. This direct contact and ability to ask questions literally brings the topic to life and motivates many toward further historical study and ap-

preciation and determination of tolerance for others. Unfortunately, due to the age of survivors, this opportunity will not be available far into the future. The New Dimensions in Testimony project (Maio et al., 2012) is an effort to preserve as much as possible of this kind of interaction.

The pilot subject is Pinchas Gutter, who has previously told his life story many times to diverse audiences. The most obvious topic of conversation is Pinchas’ experiences during World War II, including the Nazi invasion of Poland, his time in the Warsaw Ghetto, his experiences in the concentration camps, and his liberation. But there are many other topics that people bring up with Pinchas, including his pre- and post-war life and family, his outlook on life, and his favorite songs and pastimes.

### 3.2 System architecture

The automatic system is built on top of the components from the USC ICT Virtual Human Toolkit, which is publicly available.<sup>1</sup> Specifically, we use the AcquireSpeech tool for capturing the user’s speech, CMU PocketSphinx<sup>2</sup> and Google Chrome ASR<sup>3</sup> tools for converting the audio into text, NPCEditor (Leuski and Traum, 2011) for classifying the utterance text and selecting the appropriate response, and a video player to deliver the selected video response. The individual components run as separate applications on the user’s machine and are linked together by ActiveMQ messaging<sup>4</sup>: An instance of ActiveMQ broker runs on the machine, each component connects to the server and sends and receives messages to other components via the broker. The system setup also includes the JLogger component for recording the messages, and the Launcher tool that controls starting and stopping of individual tools. For example, the user can select between PocketSphinx and Google ASR engines by checking the appropriate buttons in the Launcher interface. Figure 1 shows the overall system architecture. We show the data flow through the system as black lines. Gray arrows indicate the control messages from the Launcher interface. Solid arrows represent messages passed via ActiveMQ and dotted lines represent data going over TCP/IP.

While most of the system components already

<sup>1</sup><http://vhtoolkit.ict.usc.edu>

<sup>2</sup><http://cmusphinx.sourceforge.net>

<sup>3</sup><https://www.google.com/intl/en/chrome/demos/speech.html>

<sup>4</sup><http://activemq.apache.org>

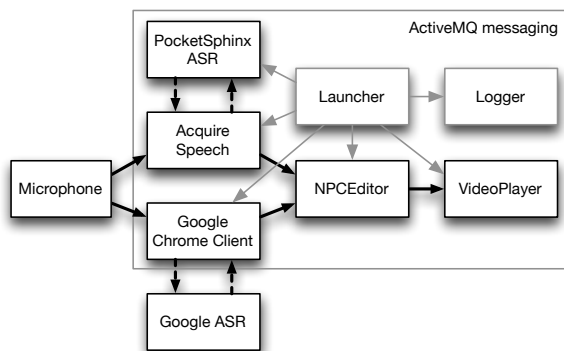


Figure 1: System architecture

existed before the start of this project, the Google Chrome ASR Client and VideoPlayer tools were developed in the course of this project. Google Chrome ASR client is a web application that takes advantage of the Google Speech API available in the Chrome browser. The tool provides push-to-talk interface control for acquiring user’s speech; it uses the API to send audio to Google ASR servers, collect the recognition result, and broadcast it over the ActiveMQ messaging. We developed the VideoPlayer tool so that we can control the response playback via the same ActiveMQ messaging. VideoPlayer also implements custom transition between clips. It has video adjustment controls so that we can modify the scale and position of the video image, and it automatically displays a loop of idle video clips while the system is in resting or listening states.

While the system was developed to be cross-platform so that it can run both on OS X and Windows, we conducted all our testing and experiments on OS X. The system is packaged as a single OS X application that starts the Launcher interface and the rest of the system. This significantly simplifies distribution and installation of the system on different computers.

### 3.3 Speech recognition

Currently the system can work with two speech recognition engines, CMU PocketSphinx and Google Chrome ASR. But for our experiments we also considered Apple Dictation.<sup>5</sup>

One major decision when selecting a speech recognizer is whether it allows for training domain-specific language models (LMs) or not.<sup>6</sup>

<sup>5</sup><https://support.apple.com/en-us/HT202584>

<sup>6</sup>While the acoustic models of a speech recognizer recognize individual sounds, the LM provides information about

Purely domain-specific LMs cannot recognize out-of-domain words or utterances. On the other hand, general-purpose LMs do not perform well with domain-specific words or utterances. Unlike PocketSphinx, which supports trainable LMs, both Google Chrome ASR and Apple Dictation come with their own out-of-the-box LMs that cannot be modified.

Table 1 shows example outputs of all three recognizers (PocketSphinx examples were obtained with a preliminary LM). As we can see, Google Chrome ASR and Apple Dictation with their general-purpose LMs perform well for utterances that are not domain-specific. On the other hand, PocketSphinx clearly is much better at recognizing domain-specific words, e.g., “Pinchas”, “Majdanek”, etc. but fails to recognize general-purpose utterances if they are not included in its LM. For example, the user input “what’s your favorite restaurant” is misrecognized as “what’s your favorite rest shot” because the word “restaurant” or the sequence “favorite restaurant” was not part of the LM’s training data. Similarly, the user input “did you serve in the army” is misrecognized as “did you certain the army” because the word “serve” or the sequence “serve in the army” was not included in the LM’s training data.

For training LMs for PocketSphinx we used the CMU Statistical Language Modeling toolkit (Clarkson and Rosenfeld, 1997) with back-off 3-grams. The CMU pronouncing dictionary v0.7a (Weide, 2008) was used as the main dictionary with the addition of domain-dependent words, such as names. We used the standard US English acoustic models that are included in PocketSphinx.

### 3.4 Dialogue policy

As mentioned in section 3.2, NPCEditor combines the functions of Natural Language Understanding (NLU) and Dialogue Management – understanding the utterance text and selecting an appropriate response. The NLU functionality is a classifier trained on linked question-response pairs, which identifies the most appropriate response to new (unseen) user input. The dialogue management logic is designed to deal with instances where the classifier cannot identify a good direct response. During training, NPCEditor calculates a response

what the recognizer should expect to listen to and recognize. If a word or a sequence of words is not included in the LM, they will never be recognized.

User Input	Google Chrome ASR Output	Apple Dictation Output	CMU Pocket Sphinx Output
hello pinchas where is lodz were you in majdanek were you in kristallnacht	hello pinterest where is lunch were you in my dannic were you and krystal knox	hello princess where is lunch were you in my donick where you went kristallnacht	hello pinchas where is lodz were you in majdanek where you when kristallnacht from
did you serve in the army have you ever lived in israel what's your favorite restaurant	did you serve in the army have you ever lived in israel what's your favorite restaurant	he served in the army that ever lived in israel what's your favorite restaurant	did you certain the army are you ever live in a israel what's your favorite rest shot

Table 1: Examples of speech recognition outputs

threshold based on the classifier’s confidence in the appropriateness of selected responses: this threshold finds an optimal balance between false positives (inappropriate responses above threshold) and false negatives (appropriate responses below threshold) in the training data. At runtime, if the confidence for a selected response falls below the predetermined threshold, that response is replaced with an “off-topic” utterance that asks the user to repeat the question or takes initiative and changes the topic (Leuski et al., 2006); such failure to return a direct response, also called non-understanding (Bohus and Rudnicky, 2005), is usually preferred over returning an inappropriate one (misunderstanding).

The current system uses a five-stage off-topic selection algorithm which is an extension of that presented in Artstein et al. (2009). The first time Pinchas fails to understand an utterance, he will assume this is a speech recognition error and ask the user to repeat it. If the misunderstanding persists, Pinchas will say that he doesn’t know (without asking for repetition), and the third time he will state that he cannot answer the user’s utterance. In a severe misunderstanding that persists beyond three exchanges, Pinchas will suggest a new topic in the fourth turn, and if even this fails to bring the user to ask a question that Pinchas can understand, then in the fifth turn Pinchas will give a quick segue and launch into a story of his choice. If at any point Pinchas hears an utterance that he can understand (that is, if the classifier finds a response above threshold), Pinchas will answer this directly, and the off-topic state will reset to zero.

A separate component of the dialogue policy is designed to avoid repetition. Normally, Pinchas responds with the top-ranked response if it is above the threshold. However, if the top-ranked response has been recently used (within a 4-turn window) and a lower ranked response

is also above the threshold, Pinchas will respond with the lower ranked response. If the only responses above threshold are among the recently used then Pinchas will choose one of them, since repetition is considered preferable to responding with an off-topic or inappropriate statement.

### 3.5 Data collection

The development process consisted of several stages: preliminary planning and question gathering, initial recording of survivor statements, Wizard of Oz studies using the recorded statements to identify gaps in the content, a second recording of survivor statements to address the gaps, assembly of an automated dialogue system, and continued testing with the automated system. The development process has been described in detail in Artstein et al. (2015); here we describe the data collected at the various stages of development, which constitute the training and test data for the automated system.

In the preliminary planning stages, potential user questions were collected from various sources, but these were not used directly as system training data. Instead, these questions formed the basis for an interview script that was used for eliciting the survivor statements during the recording sessions. The first training data include the actual utterances used during these elicitation interviews. The interviewer utterances were manually linked to the survivor responses; in the typical case, an utterance is linked to the response it elicited during the recording sessions, but the links were manually adjusted to remove instances when the response was not appropriate, and to add links to additional appropriate responses.

Additional training data were collected in the various stages of user testing – the Wizard of Oz testing between the first and second recording sessions, and fully automated system testing

Data source	Questions	Links
Elicitation	1546	2147
Wizard of Oz	1753	3329
System testing 2014	1825	1990
System testing 2015	1823	1959
Total	6947	9425

Table 2: Training data sets

following the second recording. Wizard of Oz testing took place in June and July 2014; participants sat in front of a screen that showed rough-cut video segments of Mr. Gutter’s statements, selected by human operators in response to user utterances in real time. Since the Wizard of Oz testing took place prior to the second recording, wizards were only able to choose statements from the first recording. The user utterances were recorded, transcribed, and analyzed to form the basis for the elicitation script for the second recording. Subsequent to the second recording, these utterances were reannotated to identify appropriate responses from all of the recorded statements, and these reannotated question-response links form the Wizard of Oz portion of the training data.

Testing with the automated system was carried out starting in October 2014, following the second recording of survivor statements. Users spoke to the automated system, and their utterances were recorded, transcribed, and annotated with appropriate responses. These data are partitioned into two – the testing that took place in late 2014 was mostly internal, with team members, other institute staff, and visitors, while the testing from early 2015 was mostly external, conducted over 3 days at a local museum. We thus have 4 portions of training data, summarized in Table 2.

Test data for evaluating the classifier performance were taken from the system testing in late 2014. We picked a set of 400 user utterances, collected during the last day of testing, which was conducted off-site and therefore consisted primarily of external test participants (these utterances are not counted in Table 2 above). We only included in-domain utterances for which an appropriate on-topic response was available. The evaluation therefore measures the ability of the system to identify an appropriate response when one is available, not its ability to identify instances where an on-topic response is unavailable. There

Code	Interpretation
4	Directly addresses the user question.
3	Indirectly addresses the user question, or contains additional irrelevant material.
2	Does not address the user question, but is on a related topic.
1	Irrelevant to the user question.

Table 3: Coherence rating for system responses

is some overlap in the test questions, so the 400 instances contain only 341 unique question types, with the most frequent question (*What is your name?*) occurring 5 times. We believe it is fair to include such overlap in the test set, since it gives higher weight to the more frequent questions. Also, while the text of overlapping questions is identical, each instance is associated with a unique audio file; these utterances may therefore yield different speech recognizer outputs, resulting in different outcomes.

The test set was specially annotated to serve as a test key. There is substantial overlap in content between the recorded survivor statements, so many user utterances can be addressed appropriately by more than one response. For training purposes it is sufficient to link each user utterance to some appropriate responses, but the test key must link each utterance to *all* appropriate responses. It is impractical to check each of the 400 test utterances against all 1726 possible responses, so instead we used the following procedure to identify responses that are likely to come up in response to specific test questions: we trained the system under different partitions of the training data and different training parameters, ran the test questions through each of the system versions, and from each system run we collected the responses that the system considered appropriate (that is, above threshold) for each question. This resulted in a set of 3737 utterance-response pairs, ranging from 3 to 19 responses per utterance, which represent likely system outputs for future training configurations. All the responses retrieved by the system were rated for coherence on a scale of 1–4 (Table 3). The responses rated 3 or 4 were deemed appropriate for inclusion in the test key, a total of 1838 utterance-response pairs, ranging from 1 to 10 responses per utterance.

## 4 Method

### 4.1 Speech recognition

As mentioned above, neither Google nor Apple ASRs allow for trainable LMs. But for PocketSphinx we experimented with different domain-specific LMs and below we report results on PocketSphinx performance with two different domain-specific LMs: one trained on Wizard of Oz and system testing data (approx. 5000 utterances) collected until December 2014 (LM-ds), and another one trained on additional data (approx. 6500 utterances) collected until January 2015 (LM-ds-add). The test set was the 400 utterances mentioned above. There was no overlap between the training and test data sets.

In order to evaluate the performance of the speech recognizers we use the standard word error rate (WER) metric:

$$\text{WER} = \frac{\text{Substitutions} + \text{Deletions} + \text{Insertions}}{\text{Length of transcription string}}$$

### 4.2 Classifier evaluation

Evaluation of the classifier is difficult, because it has to take into account the dialogue policy: the classifier typically returns the top-ranked response, but may return a lower-ranked response if it is above threshold and the higher-ranked responses were used recently. So while the classifier ranks all the available responses, anything below the top few will never be selected by the dialogue manager, rendering measures such as precision and recall quite irrelevant. An ideal evaluation should give highest weight to the correctness of the top-ranked response, with rapidly decreasing weight to the next several responses, but it is difficult to determine what weights are appropriate. We therefore focus on the top answer, since in most cases the top answer is what will get served to the user.

The top answer can be one of three outcomes: it can be appropriate (good), inappropriate (bad), or below threshold, in which case an off-topic response is served. A good response is better than an off-topic, which is in turn better than a bad response. This makes it difficult to compare systems with different off-topic rates: how do two systems compare if one gives more good and bad responses than the other, but fewer off-topics? We therefore compare systems using error return plots, which show the error rate across all possible return rates (Artstein, 2011): for each system we calculate the

number of errors at each return rate, and then plot the number of errors against the number of off-topics.

We used 6 combinations of the training data described in section 3.5. The baseline is trained with only the elicitation questions, and represents the performance we might expect if we were to build a dialogue system based on the recording sessions alone, without collecting user question data (except to the extent that user questions influenced the second recording session). To this baseline we successively added training data from the Wizard of Oz testing, system testing 2014, and system testing 2015. Our final training sets include the elicitation questions and system testing 2014 (without Wizard of Oz data), and the same with the system testing 2015 added.

All of the classifiers were trained in NPCEditor using the same options: text unigrams for the question language models, text unigrams plus IDs for the response language models, and F-score as the classifier scoring function during training. We used 3 versions of the test utterances: the transcribed text, the output of Google ASR, and the output of PocketSphinx, and ran each version through each of the 6 classifiers – a total of 18 configurations. For each testing configuration, we retrieved the top-ranked response for each utterance, together with the classifier confidence and a true/false indication of whether the response matched the answer key. The responses were ranked by the classifier confidence, and for each possible cutoff point (from returning zero off-topic responses to returning off-topic responses for all 400 utterances), we calculated the number of errors among the on-topic responses and plotted that against the number of off-topics. Each plot represents the error-return tradeoff for a particular testing configuration (see section 5.2).

## 5 Results

### 5.1 Speech recognition evaluation

Table 4 shows the WERs for the three different speech recognizers and the two different LMs.

Note that we also experimented with interpolating domain-specific with background LMs available from <http://keithv.com/software>. Interpolation did not help but this is still an issue under investigation. Interpolation helped with speakers who had low WERs (smooth easy to recognize speech) but hurt in cases of speakers with high

Speech Recognizer	Language Model		
	General	LM-ds	LM-ds-add
Google	5.07%	—	—
Apple	7.76%	—	—
PocketSphinx	—	22.04%	19.39%

Table 4: Speech recognition results (WER). General LM stands for general-purpose LM, LM-ds stands for domain-specific LM trained with data collected until December 2014, and LM-ds-add stands for domain-specific LM trained with additional data collected until January 2015.

WERs. In the latter cases, having a background model meant that there were more choices for the speech recognizer to choose from, which instead of helping caused confusion.

We also noticed that PocketSphinx was less tolerant of environmental noises, which most of the time resulted in insertions and substitutions. For example, as we can see in Table 1, the user input “have you ever lived in israel” was misrecognized by PocketSphinx as “are you ever live in a israel”. These misrecognitions do not necessarily confuse the classifier, but of course they often do.

## 5.2 Classifier evaluation

Classifier performance is best when training on all the data, and testing on transcriptions rather than speech recognizer output. Figure 2 shows the effect of the amount of training data on classifier performance when tested on transcribed text (a similar effect is observed when testing on speech recognizer output). Lower curves represent better performance. As expected, performance improves with additional training data – training on the full set of data cuts error rates by about a third compared to training on the elicitation questions alone. Additional training data (both new questions and question-response links) are likely to improve performance even further.

The effect of speech recognition on classifier performance is shown in Figure 3. Automatic speech recognition does impose a performance penalty compared to testing on transcriptions, but the penalty is not very large: classifier errors when testing with Google ASR are between 1 and 3 percentage points higher than with transcriptions, while PocketSphinx fares somewhat worse, with classifier errors about 5 to 8 percentage points

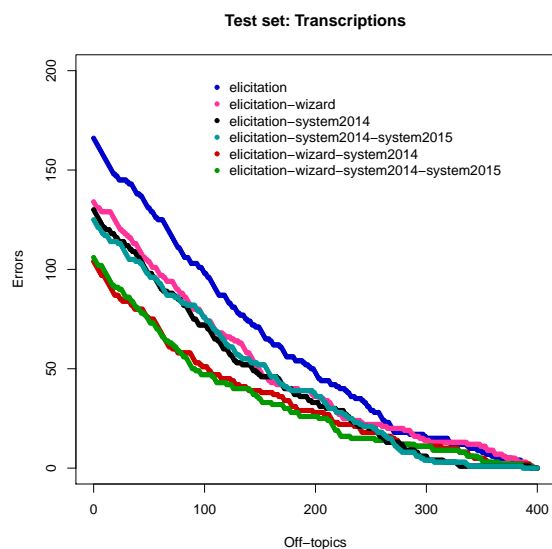


Figure 2: Tradeoff between errors and off-topics for various training sets (tested on transcribed text)

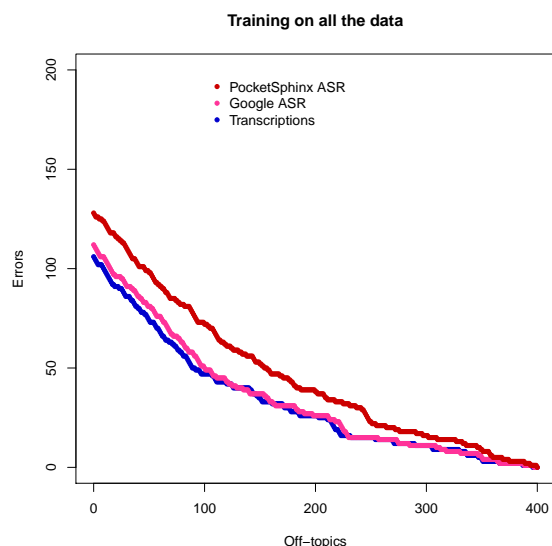


Figure 3: Tradeoff between errors and off-topics for different test sets (trained on the full data)

higher than with transcriptions. At a 20% off-topic rate, the response error rates are 14% for transcriptions and 16% for Google ASR, meaning that almost two thirds of user utterances receive a direct appropriate response. At 30% off-topics, errors drop to 10–11%, and direct appropriate responses drop to just shy of 60%. Informal impressions from current testing at a museum (section 6) suggests that these numbers are sufficient to enable a reasonable conversation flow.

## 6 Discussion

This paper has demonstrated that time-offset interaction with a real person is achievable with present day spoken language processing technology. Not only are we able to collect a sufficiently large and varied set of statements to address user utterances (Artstein et al., 2015), we are also able to use speech recognition and language understanding technology to identify appropriate responses frequently enough to enable a natural interaction flow. Future work is needed in three areas: investigating the interaction quality of the dialogue system, improving the language processing, and generalizing the process to additional situations.

To investigate the interaction quality, we need to look at dialogues in context rather than as isolated utterances, and to collect user feedback. We are presently engaged in a joint testing, demonstration, and data collection effort that is intended to address these issues. The time-offset interaction system has been temporarily installed at the Illinois Holocaust Museum and Education Center in Skokie, Illinois, where visitors interact with the system as part of their museum experience (Isaacs, 2015). The system is set up in an auditorium and users talk to Pinchas in groups, in a setting that is similar to in-person encounters with Holocaust survivors which also take place at the museum. Due to physical limitations of the exhibit space, interaction is mediated by museum docents: each user question is relayed by the docent into the microphone, and Pinchas responds to the docent's speech. An excerpt of museum interaction is in the Appendix. Data and feedback from the museum installation will be used to evaluate the interaction quality, including user feedback as to the naturalness of the interaction and user satisfaction.

The ongoing testing also serves the purpose of data collection for improving system performance: Figure 2 shows that errors diminish with additional training data, and it appears that we have not yet reached the point of diminishing returns with about 7000 training utterances. We hope to collect an average of 10 training utterances per response, that is about 17000 user utterances. Annotation is also incomplete: the test key has an average of 4.6 links per utterance, as opposed to an average of around 1.4 links per utterance in the training data. While complete linking is not necessary for classifier operation, improving the links will probably improve performance.

In addition to improving performance through improved data, there are also algorithmic improvements that can be made to the language processing components. One goal is to leverage the relative strengths of the general purpose and domain-specific ASRs, e.g., through the classifier: past work has shown that language understanding can be improved by allowing NLU to select from among several hypotheses provided by a single speech recognizer (Morbini et al., 2012), and we propose to try a similar method to utilize the outputs of separate speech recognizers. Another idea is to combine/align the outputs of the speech recognizers (before they are forwarded to the classifier) taking into account information from the recognition confidence scores and lattices. This will potentially help in cases where different recognizers succeed in correctly recognizing different parts of the utterance.

Time-offset interaction has a large potential impact on preservation and education – people in the future will be able to not only see and listen to historical figures, but also to interact with them in conversation. Future research into time-offset interaction will need to generalize the development process, in order to enable efficient use of resources by identifying common user questions that are specific to the person, ones that are specific to the dialogue context or conversation topic, and ones that are of more general application.

## Acknowledgments

This work was made possible by generous donations from private foundations and individuals. We are extremely grateful to The Pears Foundation, Louis F. Smith, and two anonymous donors for their support. The work was supported in part by the U.S. Army; statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. Heather Maio and Alesia Gainer spent long hours on data collection and system testing. The Los Angeles Museum of the Holocaust, the Museum of Tolerance, and New Roads School in Santa Monica offered their facilities for data collection. The USC Shoah Foundation provided financial and administrative support, and facilities. Finally, we owe special thanks to Pinchas Gutter for sharing his story, and for his tireless efforts to educate the world about the Holocaust.



## References

- Ron Artstein, Sudeep Gandhe, Jillian Gerten, Anton Leuski, and David Traum. 2009. Semi-formal evaluation of conversational characters. In Orna Grumberg, Michael Kaminski, Shmuel Katz, and Shuly Wintner, editors, *Languages: From Formal to Natural. Essays Dedicated to Nissim Francez on the Occasion of His 65th Birthday*, volume 5533 of *Lecture Notes in Computer Science*, pages 22–35. Springer, Heidelberg, May.
- Ron Artstein, David Traum, Oleg Alexander, Anton Leuski, Andrew Jones, Kallirroi Georgila, Paul Debevec, William Swartout, Heather Maio, and Stephen Smith. 2014. Time-offset interaction with a Holocaust survivor. In *Proceedings of IUI*, pages 163–168, Haifa, Israel, February.
- Ron Artstein, Anton Leuski, Heather Maio, Tomer Mor-Barak, Carla Gordon, and David Traum. 2015. How many utterances are needed to support time-offset interaction? In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference*, pages 144–149, Hollywood, Florida, May. AAAI Press.
- Ron Artstein. 2011. Error return plots. In *Proceedings of SIGDIAL*, pages 319–324, Portland, Oregon, June.
- Dan Bohus and Alexander I. Rudnicky. 2005. Sorry, I didn't catch that! – An investigation of non-understanding errors and recovery strategies. In *Proceedings of SIGDIAL*, pages 128–143, Lisbon, Portugal, September.
- Lucy Chabot. 1990. Nixon library technology lets visitors 'interview' him. *Los Angeles Times*, July 21.
- Philip Clarkson and Ronald Rosenfeld. 1997. Statistical language modeling using the CMU-Cambridge toolkit. In *Proc. of Eurospeech*, Rhodes, Greece, September.
- Sudeep Gandhe and David Traum. 2010. I've said it before, and I'll say it again: An empirical investigation of the upper bound of the selection approach to dialogue. In *Proceedings of SIGDIAL*, pages 245–248, Tokyo, September.
- Mike Isaacs. 2015. Holocaust Museum: Pilot program aims to preserve survivor voices for future generations. *Chicago Tribune*, May 19.
- Anton Leuski and David Traum. 2011. NPCEditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32(2):42–56.
- Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proceedings of SIGDIAL*, Sydney, Australia, July.
- Heather Maio, David Traum, and Paul Debevec. 2012. New dimensions in testimony. *PastForward*, Summer:22–26.
- Donald Marinelli and Scott Stevens. 1998. Synthetic interviews: The art of creating a 'dyad' between humans and machine-based characters. In *Proceedings of the Sixth ACM International Conference on Multimedia: Technologies for Interactive Movies (MULTIMEDIA)*, pages 11–16, New York, NY, USA. ACM.
- Fabrizio Morbini, Kartik Audhkhasi, Ron Artstein, Maarten Van Segbroeck, Kenji Sagae, Panayiotis Georgiou, David R. Traum, and Shri Narayanan. 2012. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *Proceedings of the Fourth IEEE Workshop on Spoken Language Technology (SLT)*, pages 49–54, Miami Beach, Florida, December.
- Eric Sloss and Anne Watzman. 2005. Carnegie Mellon's Entertainment Technology Center conjures up Benjamin Franklin's ghost. Press release, Carnegie Mellon Media Relations, June 28. [http://www.cmu.edu/PR/releases05/050628\\_etc.html](http://www.cmu.edu/PR/releases05/050628_etc.html).
- R.L. Weide. 2008. The CMU pronouncing dictionary.

## A Dialogue sample

The following dialogue excerpt is taken from the beta testing in Illinois (see section 6). Speech recognition output is shown in italics when it departs substantially from the spoken text.

**User** Hello Pinchas, how are you?  
*Las Vegas how are you*

**Pinchas** Can you just repeat that?

**User** Can you hear me OK?

**Pinchas** I can see you and I can hear you.

**User** Hello Pinchas, can you hear me?  
*how thick is can you hear me*

**Pinchas** I can hear you, yeah.

**User** Pinchas, can you tell me how old you are?  
*Vegas can you tell me how old you are*

**Pinchas** I was born in nineteen thirty two and we are now in two thousand and fourteen and you can make your own reckoning.

**User** Can you tell me a little bit about your childhood?

**Pinchas** My life before the war, I was born in Lodz in Poland, and I was eight years old when the war started. So the first eight years I lived in a home, a very religious one, a Hasidic home. I had a twin sister, my...

**User** Have you ever met Hitler?  
*mermaid Hitler*

**Pinchas** Please repeat that.

**User** Did you ever meet Hitler?

**Pinchas** Did I ever meet Hitler? Well, Hitler was a Führer of Germany and I was born in Poland and I was a little Jewish boy, so the chances of somebody like me meeting Hitler was impossible, because Hitler was...

**User** What would you do if you met Hitler?

**Pinchas** Geyn drerd arein, that's Yiddish and what it means go and bury yourself.