Which ASR should I choose for my dialogue system?

Fabrizio Morbini, Kartik Audhkhasi, Kenji Sagae, Ron Artstein,
Doğan Can, Panayiotis Georgiou, Shri Narayanan, Anton Leuski and David Traum
University of Southern California
Los Angeles, California, USA
{morbini,sagae,artstein,leuski,traum}@ict.usc.edu
{audhkhas,dogancan}@usc.edu {georgiou,shri}@sipi.usc.edu

Abstract

We present an analysis of several publicly available automatic speech recognizers (ASRs) in terms of their suitability for use in different types of dialogue systems. We focus in particular on cloud based ASRs that recently have become available to the community. We include features of ASR systems and desiderata and requirements for different dialogue systems, taking into account the genre of the system, type of user, and other features. We then present speech recognition results for six different dialogue systems. The most interesting result is that different ASR systems perform best on these different data sets. We also show that there is an improvement over a previous generation of recognizers on some of these data sets. We also investigate NLU performance on the ASR output, and explore the relationship between ASR and NLU performance.

1 Introduction

Dialogue system developers who are not also speech recognition experts are in a better position than ever before in terms of the ease of integrating existing speech recognizers in their systems. While there have been commercial solutions and toolkits for a number of years, there were a number of problems in getting these systems to work. For example, early toolkits relied on specific machine hardware, software, and firmware to function properly, often had a difficult installation process, and moreover often didn't work well for complex dialogue domains, or challenging acoustic environments. Fortunately the situation has greatly improved in recent years. Now there are a number of easy to use solutions, including open-source systems (like PocketSphinx), as well as cloud-based approaches.

While this increased choice of quality recognizers is of great benefit to dialogue system developers, it also creates a dilemma – which recognizer to use? Unfortunately, the answer is not simple – it depends on a number of issues, including the type of dialogue domain, availability and amount of training data, availability of internet connectivity for the runtime system, and speed of response needed. In this paper we assess several freely available speech recognition engines, and examine their suitability and performance in several dialogue systems. Here we extend the work done in Morbini et al. (2012) focusing in particular on cloud based freely available ASR systems. We include 2 local ASRs for reference, one of which was also used in the earlier work for easy comparison.

2 Speech Recognizer Features and Engines

The following are some of the major criteria for selection of a speech recognizer.

Customization Some of the available speech recognizers allow the users to tune the recognizer to the environment it will operate in, by providing a specialized **lexicon**, trained **language models** or **acoustic models**. Customization is especially important for dialogue systems whose input contains specialized vocabulary (see section 4).

Output options A basic recognizer will output a string of text, representing its best hypothesis about the transcription of the speech input. Some recognizers offer additional outputs which are useful for dialogue systems: ranked **n-best hypotheses** allow later processing to use context for disambiguation, and **incremental results** allow the system to react while the user is still speaking.

Performance characteristics Dialogue systems differ in their requirements for **response speed**; a

System	Customization	Output options		Open Source	Performance	
		N-best	Incremental		Speed	Installation
Pocketsphinx	Full	Yes	Yes	Yes	realtime	Local
Apple	No	No[a]	No	No	network	Cloud
Google	No	Yes	Yes[e]	No	network	Cloud
AT&T	Partial[b]	Yes	No	No	network	Cloud
Otosense-Kaldi	Full	Yes	No	Yes[d]	variable[c]	Local

[a] Single output annotated with alternative hypotheses [b] Custom language models, fixed set of acoustic models
[c] User can control the trade-off between speed and output quality. [d] Release scheduled for Fall 2013.
[e] Only for web-delivered applications when used in a Google Chrome browser.

Table 1: Speech recognizer features important for use in dialogue systems

speech recognizer that **runs locally** can help by avoiding network latencies.

Output quality Typically, a dialogue system would want the best **recognition accuracy** possible given the constraints. Ultimately, dialogue systems want the output that would yield the best performance for **Natural Language Understanding** and other downstream processes. As a rule, better speech recognition leads to better language understanding, though this is not necessarily the case for specific applications (see section 5).

We evaluated 5 freely available speech recognizers. Their features are summarized in Table 1. We did not include the MIT WAMI toolkit[1] as we are focused on speech services that can directly be used by stand alone applications as opposed to web delivered ones.

Pocketsphinx Pocketsphinx (Huggins-Daines et al., 2006) is a version of the CMU Sphinx ASR system optimized to run also on embedded systems. Pocketsphinx is fast, runs locally, and requires relatively modest computational resources. It provides n-best lists and lattices, and supports incremental output. It also provides a voice activity detection functionality for continuous ASR. This ASR is fully customizable and trainable, but users are expected to provide language models suitable for their applications. A few acoustic models are provided, and can be adapted using the CMUSphinx tools.[2]

Apple Dictation Apple's Dictation is the OS level feature in both MacOSX and iOS.[3] It is integrated into the text input system pipeline so a user can replace her keyboard with a microphone for entering text in any application. Dictation is often associated with the Siri personal assistant feature of iOS. While it is likely that Dictation and Siri share the same ASR technology, Dictation only does speech recognition. Apple states that Dictation learns the characteristics of the user's voice and adapts to her accent (Apple Inc, 2012). Dictation requires an internet connection to send recorded user speech to Apple's servers and receive ASR results. Processing starts as soon as the user starts speaking so the delay of getting the recognition results after the user finishes speaking is minimal.

To integrate Dictation into a dialogue system, a system designer needs to include any system defined text input control into her application and use the control APIs to observe text changes. The user would need to press a key when starting to speak and push the key again once she is done speaking. The ASR result is a text string annotated with alternative interpretations of individual words or phrases in the text. There is an API for extracting those interpretations from the result. While the Dictation feature is reasonably fast and easy to integrate, dialogue system developers have no control over the ASR process, which must be treated as a black box. Apple dictation is limited in that no customization is possible, no partial recognition results are provided, and there is an unspecified limit on the number of utterances dictated for a period of time, which is not a problem for interaction between a single user and a dialogue system, but may be an issue in dialogue systems that support multiple concurrent users.

[1] http://wami.csail.mit.edu/

[2] http://cmusphinx.sourceforge.net/wiki/tutorialadapt

[3] Dictation was introduced in iOS 5.0 and MacOSX 10.8.

Google Speech API The Google speech API[4] provides supports for the HTML 5 speech input feature. It is a cloud based service in which a user submits audio data using a HTML POST request and receives as reply the ASR output in the form of an n-best list. The audio data is limited to roughly 10 seconds in length, longer clips are rejected and return no ASR results. The user can (1) customize the number of hypotheses returned by the ASR, (2) specify which language the audio file contains and (3) enable a filter to remove profanities from the output text. As is the case with Apple Dictation, ASR must be treated as a black box, and no task customization is possible for dialogue system developers. Users cannot specify or provide custom language models or acoustic models. The service returns only the final hypothesis, there is no incremental output[5]. In addition, results for the same inputs may change unpredictably, since Google may update or otherwise change its service and models, and models may be adapted using specific audio data supplied by users. In our experiments, we observed accuracy improvements when submitting the same audio files over repeated trials over two weeks.

AT&T Watson Watson is the AT&T ASR engine available through the AT&T Speech Mashup service.[6] It is a cloud based service that can be accessed through HTML POST requests, like the Google Speech API. AT&T Watson is designed to support the demands of online spoken dialogue systems, and can be customized with data specific to a dialogue system. Additionally, in our tests we did not observe any limitation in the maximum length of the input audio data. However, AT&T does not provide a default general-purpose language model, and application-specific models must be built within the Speech Mashup service using user-provided text data. The acoustic model must be selected from a list provided by the AT&T service, and acoustic models can be further customized within the Speech Mashup service. The ASR returns an n-best list of hypothesis but does not provide incremental output.

Otosense-Kaldi Another ASR we employed was the Kaldi-based OtoSense-Kaldi engine developed at SAIL.[7] OtoSense-Kaldi[8] is an on-line, multi-threaded architecture based on the Kaldi toolkit (Povey et al., 2011) that allows for dynamically configurable and distributed ASR.

3 Dialogue Systems, Users, and Data

All spoken dialogue systems are similar in some respects, in that there is speech by a user (or users) that needs to be recognized, and this speech is punctuated by speech from the system. Moreover, the speech is not fully independent, but utterances are connected to other utterances, e.g. answers to questions, or clarifications. There are, however many ways in which systems can differ, that have implications for which speech recognizers are most appropriate. Some of the dimensions to consider are:

Type of microphone(s) One of the biggest impacts on ASR is the acoustic environment. Will the audio be clean, coming from a close-talking head or lapel-mounted microphone, or will it need to be picked up from a broader directional microphone or microphone array?

Number of speakers/microphones Will there be one designated microphone per person, or will speaker identification need to be performed? Will audio from the system confuse the ASR?

Push to talk or continuous speech Will the user clearly identify the start and end of speech, or will the system need to detect speech acoustically?

Type of Users Will there be designated long-term users, where user-training or system model adaptation is feasible, or will there be many unknown users, where training is not feasible? See also section 3.1 for more on user types.

Genre What kinds of things will people be saying to the system? Is it mostly commands or short answers to questions, or more open-ended conversation? See section 3.2 for more on genre issues.

Training Data Is within-domain training data available, and if so how much?

[4] https://www.google.com/speech-api/v1/recognize

[5] The demo page available at http://www.google.com/intl/en/chrome/demos/speech.html shows continuous speech understanding with incremental results but require Google Chrome to run and is specific to web delivered applications. A more in depth inspection of Google Chrome's source code may reveal how to enable that feature also in stand alone applications.

[6] https://service.research.att.com/smm

[7] http://sail.usc.edu

[8] OtoSense-Kaldi will be released (BSD license) in 2013.

3.1 Types of Users

The type of user is important for the overall design of the system and has implication for ASR performance as well. One important aspect is the broad physical differences among speakers, such as male vs female, adult vs child (e.g. Bell and Gustafson, 2003), or language proficiency/accent, that will have implications for the acoustics of what is said, and ASR results. Other aspects of users have implications for what will be said, and how successful the interface may be, overall. Many (e.g. Hassel and Hagen, 2006; Jokinen and Kanto, 2004) have looked at the differences between novice and expert users. Ai et al. (2007a) also points out a difference between real users and recruited subjects. Real users also come in many different flavors, depending on their purposes. E.g. are they interacting with the system for fun, to do a specific task that they need to get done, to learn something (specific or general), or with some other purpose in mind?

We considered the following classes of users regarding overall system design/performance, order in terms of easiest to hardest to get to acceptable performance and robustness levels:

Demonstrator: generally the easiest for a system to understand – a demonstrator is trained in use of the system, knows what can and can't be said, is motivated toward success, and is generally interested in showing off the most impressive/successful aspects of the system to an audience rather than using it for its own sake.

Trained/Expert User: similar to the demonstrator, but is using the system to achieve specific results rather than just to show off its capabilities. This means that the user may be forced down lines that are not ideal for the system, if these are necessary to accomplish the task.

Motivated User: does not have the training of the expert user, and may say many things that the system can not handle as opposed to equivalent expressions that could be handled. However the user does want the system to succeed, and is in general willing to do whatever he or she thinks is necessary to improve system performance. Unlike the expert user, the motivated user might be incorrect about what will help the system (e.g. hyperarticulation in response to system misunderstanding).

Casual User: interested in finding out what the system can do, but does not have particular motivations to help or hinder the system. Casual Users may also leave in the middle of an interaction, if it is not engaging enough.

Red Team: out to try to test or "break" the system, or show it as not-competent, and may try to do things the system can't understand or react well to, even when an alternative formulation is known to work.

3.2 Types of Dialogue System Genres

Dialogue Genres can be distinguished along many lines, e.g. the number and relationship of participants, specific conversational rules, purposes of the participants, etc. We distinguish here four genres of dialogue system that have been in use at the Institute for Creative Technologies. Each genre has implications for the internal representations and system architectures needed to engage in that genre of dialogue.

Simple Question-answering This genre involves strong user-initiative and weak global dialogue coherence. The user can ask any question to the system at any time, and the system should respond, with an appropriate answer if able, or with some other reply indicating either inability or unwillingness to provide the answer. This genre allows modeling dialogue at a surface-text level (Gandhe, 2013), without internal semantic representations of the input, and where the result of "understanding" input is the system's expected output. The NCPEditor[9] (Leuski and Traum, 2011) is a toolkit that provides an authoring environment, classification, and dialogue capability for simple question-answering characters. The SGT Blackwell, SGT Star, and Twins systems described below are all systems in this genre.

Advanced Question-answering This genre is similar to the simple question-answering characters, in that the main task of the user is to elicit information from the system character. The difference is that there is more intermediate dialogue coherence, in that questions can be answered several utterances after they have been asked, there can be intervening sub-dialogues, and characters sometimes take the initiative to pursue their own

[9] Available free for academic research purposes from https://confluence.ict.usc.edu/display/VHTK/Home

goals rather than just responding to the user. Because of the requirements for somewhat deeper understanding, and relation of input to context and character goals and policies, there is a need of at least a shallow semantic representation and representation of the dialogue information state, and the character must distinguish understanding of the input from the character output (since the latter will depend on the dialogue policy and information state, not just the understanding of input). The tactical questioning architecture (Gandhe et al., 2009) provides authoring and run-time support for advanced question-answering characters, and has been used to build over a dozen characters for purposes such as training tactical questioning, training culture, and psychology experiments (Gandhe et al., 2011). The Amani character described below is in this genre.

Slot-filling Probably the most common type of dialogue system (at least in the research community) is slot-filling. Here the dialogue is fairly structured, with an initial greeting phase, then one or more tasks, which all start with user selecting the task, and the system taking over initiative to "fill" and possibly confirm the needed slots, before retrieving some information from a database, or performing a simple service. Mixed-initiative versions of this genre exist, where the user can also provide unsolicited information, which reduces the number of system queries needed. This genre also requires a semantic representation, at least of the slots and acceptable values. Generally, the set of possible values is large enough, that some form of NLG is needed (at least template filling), rather than authoring of all full sentences. There are a number of toolkits and development frameworks that are well suited to slot-filling systems, e.g. Ravenclaw (Bohus and Rudnicky, 2003) or Trindikit (Larsson and Traum, 2000). The Radiobots system, described below is in this genre.

Negotiation and Planning In this genre, the system is more of an equal partner with the user, than a servant, as in the slot-filling systems. The system must not merely understand user requests, but must also evaluate whether they meet the system goals, what the consequences and preconditions of requests are, and whether there are better alternatives. For this kind of inference, a more detailed semantic representation is required than just filling in slots. While we are not aware of publicly available software that makes this kind of system easy to construct, there have been several built using an information-state approach, or the soar cognitive architecture. The TRIPS system (Allen et al., 2001) also has many similarities.

3.3 ICT Dialogue Systems Tested

We tested the recognizers described in section 2 on data sets collected from six different dialogue domains. Five are the same ones tested in Yao et al. (2010), to which we added the Twins set.

SGT Blackwell was created as a virtual human technology demonstration for the 2004 Army Science Conference. This is a question-answering character, with no internal semantic representation and the primary NLU task merged with Dialogue management as selecting the best response.

The original users were ICT demonstrators. However, there were also some experiments with recruited participants (Leuski et al., 2006a; Leuski et al., 2006b). Later SGT Blackwell became a part of the "best design in America" triennial at the Cooper-Hewitt Museum in New York City, and the data set here is from visitors to the museum, who are mostly casual users, but range from expert to red-team. Users spoke into a mounted directional microphone (see Robinson et al., 2008 for more details).

SGT STAR (Artstein et al., 2009a) is a question-answering character similar to SGT Blackwell, although designed to talk about Army careers rather than general knowledge. The users are Army personnel who went to job fairs and visited schools in the mobile Army adventure vans, speaking using headset microphones, and performing for an audience. The users are somewhere between demonstrators and expert users. They are speaking to SGT STAR for the benefit of an audience, but their primary purpose is to convey information to the audience in a memorable way (through dialogue with SGT STAR) rather than to show off the highlights of the character.

The Twins are two life-size virtual characters who serve as guides at the Museum of Science in Boston (Swartout et al., 2010). The characters promote interest in Science, Technology, Engineering and Mathematics (STEM) in children between the ages of 7 and 14. They are question-answering characters, but unlike SGTs Blackwell and Star, the response is a whole dialogue sequence, potentially involving interchange from both characters, rather than a single character turn.

There are two types of users for the Twins: demonstrators, who are museum staff members, using head-mounted microphones, and museum visitors, who use a Shure 522 table-top mounted microphone (Traum et al., 2012). More on analysis of the museum data can be found in (Aggarwal et al., 2012). We also investigated speech recognition and NLU performance in this domain in (Morbini et al., 2012).

Amani (Artstein et al., 2009b; Artstein et al., 2011) is an advanced question-answering character used as a prototype for systems meant to train soldiers to perform tactical questioning. The users are in between real users and test subjects: they were cadets at the U.S. Military Academy in April 2009, who interacted with Amani as a university course exercise on negotiation techniques. They were interacting in order to practice negotiation skills learned in the class, however. They used head-mounted microphones to talk with Amani.

Radiobots (Roque et al., 2006) is a training prototype that responds to military calls for artillery fire in a virtual reality urban combat environment. This is a domain in the slot-filling genre, where there is a preferred protocol for the order in which information is provided and confirmed. Users are generally trainees, learning how to do calls for fire, they are motivated users with some training. The semantic processing involved tagging each word with the dialogue act and parameter that it was associated with (Ai et al., 2007b).

This data set was collected during the development of the system in 2006 at Fort Sill, Oklahoma, during two evaluation sessions from recruited volunteer trainees who performed calls for specific missions (Robinson et al., 2006). These subjects used head-mounted microphones rather than the ASTI simulated radios from later data collection.

SASO-EN (Traum et al., 2008) is a negotiation training prototype in which two virtual characters negotiate with a human "trainee" about moving a medical clinic. The genre is negotiation and planning, where the human participant must try to form a coalition, and the characters reason about utilities of different proposals, as well as causes and effects. The output of NLU is a frame representation including both semantic elements, like thematic argument structure, and pragmatic elements, such as addressee and referring expressions. Further contextual interpretation is performed by each of the virtual characters to match the (possibly partial) representation to actions and states in their task model, resolve other referring expressions, and determine a full set of dialogue acts (Traum, 2003). Speech was collected at the USC Institute for Creative Technologies (ICT) during 2006–2009, mostly from visitors and new hires, who acted as test subjects.

4 ASR Performance

We tested each of the Datasets described in Section 3.3 with some of the recognizers described in Section 2. All recognizers were tested on the Amani, SASO-EN, and Twins domains, and we also tested a natural language understanding component on these data sets (Section 5). For SGT Blackwell, SGT STAR, and Radiobots, we report the performance on the same development set used in (Yao et al., 2010). For Amani and SASO-EN (where we also report the NLU performance), we run a 10-fold cross-validation in which 9 folds where used to train the NLU and ASR language model and the 10^{th} was used for testing. For the Twins dialogue system, we used the same partition into training, development and testing reported in Morbini et al. (2012) and the results reported here are from the development set. The different experimental setup used do not affect the information provided as the performance of different ASR systems should be compared only within each dialogue domain and not across different dialogue domains.

Table 2 summarizes the performance of the various ASR engines on the evaluation data sets. Performance is measured as Word Error Rate and was obtained using the NIST SCLITE tool.[10]

Note that only Otosense-Kaldi in the Twins domain had adapted acoustic models. In the remaining cases only the language model was adapted. Looking at the results on the development set reported in (Yao et al., 2010), we have improvements in 3 out of 5 domains: Amani (-11.8% Google), SASO-EN (-11.7% AT&T) and SGT Blackwell (-13% Google). In Radiobots and SGT Star the performance achieved with just language model adaptation, when permitted, is worse: +4.8% and +1.7% respectively.

We find that there is no single best performing

[10] http://www.itl.nist.gov/iad/mig/tools/

Speech recognizer	Evaluation data set					
	Amani	Radiobots	SASO-EN	SGT Blackwell	SGT Star	Twins
Pocketsphinx	39.7	**11.8**	28.4	51	28.6	81
Apple	28	—	30.9	—	—	29
AT&T	29	12.1	**16.3**	27.3	**21.7**	28.8
Google	**23.8**	36.3	20	**18**	26	20.6
Otosense-Kaldi	33.7	—	22.1	—	—	**18.7**

Table 2: Word Error Rates (%) for the various dialogue systems and ASR systems tested.

speech recognizer: results vary greatly between the evaluation test sets. In 4 of the 6 datasets overall, and 2 of the 3 datatests tested with Otosense-Kaldi, the best performer is a cloud-based service (Google or AT&T). There are two datasets for which a local, fully customizable recognizer performs better than the cloud-based services. Radiobots, consisting of military calls for artillery fire, has a fairly limited and very specialized vocabulary, and indeed the two recognizers with custom language models (Pocketsphinx and AT&T) perform much better than the non-customizable recognizer (Google).

The Twins dataset is unique in that for the Otosense-Kaldi system we custom-trained acoustic and language models, while standard WSJ acoustic models and adapted language models were used for the other dialogue systems. In both cases the models were triphone based with an Linear Discriminant Analysis (LDA) front end, and Maximum Likelihood Linear Transformation (MLLT) and Maximum Mutual Information (MMI) training. This reflects on the very good performance in the Twins domain, decent performance on the SASO-EN domain (reasonable mismatch of WSJ and SASO-EN) and very degraded performance in Amani (highly mismatched Amani and WSJ domains). The observed degradation in performance is accentuated by the MMI discriminative training on the mismatched-WSJ data. As with PocketSphinx and Watson, and unlike with Apple Dictation and Google Speech API, with Kaldi we fully control experimental conditions and can guarantee no contamination of the train-test data.

In summary, our evaluation shows that customizable recognizers are useful when the expected speech is highly specialized, or when substantial resources are available for tuning the recognizer.

5 NLU Accuracy & Relation between ASR and NLU

While the different genres of system have different types of output for NLU: response text, dialogue act and parameter tags, speech acts, or semantic frames, many of them can be coerced into a selection task, in which the NLU selects the right output from a set of possible outputs. This allows any multiclass classification algorithm to be used for NLU. A possible drawback is that for some inputs, the right output might not be available in the set considered by the training data, even if it might easily be constructed from known parts using a generative approach.

A second issue is that even though we can cast the problem as multi-class classification, classification accuracy is not always the most appropriate metric of NLU quality. For question-answering characters, getting an appropriate and relevant reply is more important than picking the exact reply selected by a human domain designer or annotator: there might be multiple good answers, or even the best available answer might not be very good. For that reason, the question-answering characters allow an “off-topic” answer and Error-return plots (Artstein, 2011) might be necessary to choose an optimal threshold. For the SASO-EN system, slot-filler metrics such as precision, recall, and f-score are more appropriate than frame accuracy, because some frames may have many slots in common and few that are different (e.g. just a different addressee). Nonetheless, we begin our analysis within this common framework. For simplicity, we start with just three domains: Twins, Amani, and SASO-EN. SGT STAR and Blackwell are very similar to Twins in terms of NLU. Radiobots is more challenging to coerce to multiclass classification.

Conventional wisdom in the speech and lan-

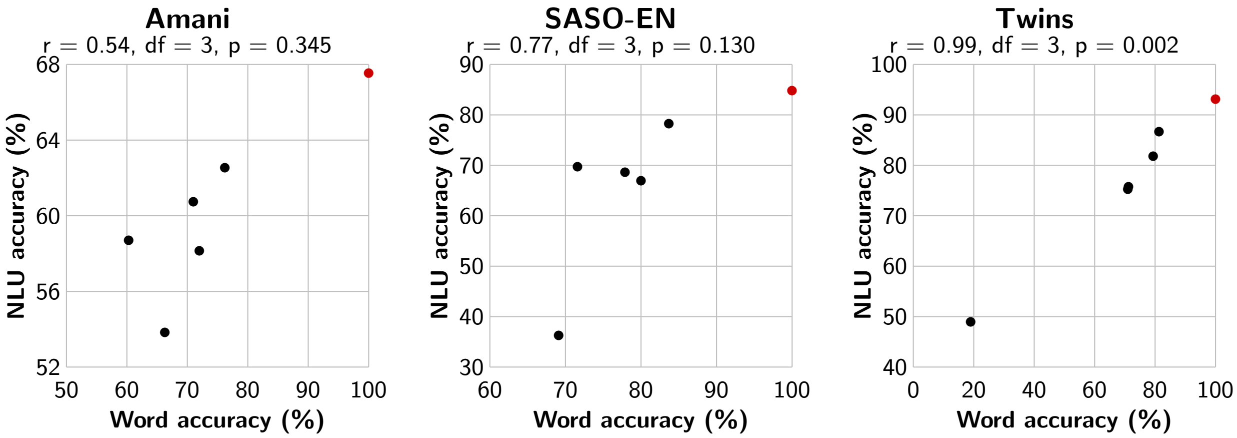

Figure 1: Relation between ASR and NLU performance (red dots are manual transcriptions)

guage processing community is that performance of ASR and NLU are closely tied: improved speech recognition leads to better language understanding, while deficiencies in speech recognition cause difficulty in understanding. This conventional wisdom is borne out by decades of experience with speech and dialogue systems, though we are not aware of attempts to systematically demonstrate it. The present study shows that the expected relation between speech recognition and language understanding holds for the systems we tested.

Accepted assumptions about the relation between speech recognition and language understanding have been repeatedly challenged. Direct challenges are typically limited to specific applications. Wang et al. (2003) show that for a slot-filling NLU, ASR can be specifically tuned to recognize those words that are relevant to the slot-filling task, resulting in improved understanding despite a decrease in performance on overall word recognition. Alshawi (2003) and Huang and Cox (2006) show that in call-routing applications the word level can be dispensed with altogether and calls routed based on phonetic information alone without noticeable loss in performance. These challenges suggest that the speech-language divide is not as clean as the theory suggests.

To investigate the relation between ASR and NLU, we ran each ASR output from each of the 5 recognizers through an understanding component to obtain an NLU output (each dataset had a separate NLU component, which was held constant for all speech recognizers). ASR and NLU performance are conventionally measured on scales of opposite polarity: better performance shows up as lower word error rates but higher NLU accuracies. For the correlations we invert the conventional ASR scale and use word accuracy, so that higher numbers signify better performance on both scales.[11]

Figure 1 shows the results obtained in the 3 dialogue systems by the various ASR systems. The figures plot ASR performance against NLU performance; NLU results on manual transcriptions are included for comparison. There are too few data points for the correlations between ASR and NLU performance to be significant, but the trends are positive, as expected.

Our experiments lend supporting evidence to the claim that in general, ASR performance is positively linked to NLU performance (special cases notwithstanding). The 3 datasets exhibit strong correlations between speech recognition and language understanding performance. Thus, we claim that the basis of the conventional wisdom is sound: speech recognition directly affects language understanding. This conclusion holds *when the speech recognizer has been optimized to produce the most accurate transcript,* rather than for a specific NLU.

6 Conclusion and Future Work

We have extended here the ASR system evaluation published in Yao et al. (2010) including some new cloud based ASR services that achieve very good performance showing an improvement of around 12%. We also showed that ASR and NLU performance are correlated.

One possible avenue of future work is to extract importance weights for each word from the learnt NLU models and use these weights to try

[11] We define “accuracy” as 1 minus WER, so this number can in principle dip below zero if there are more errors than words.

to explain those cases that diverge from the correlation between ASR and NLU performance. This may also give us a better measure than WER for assessing ASR performance in dialogue systems. Another avenue of future work involves examining different types of NLU engines, and different metrics for the different dialogue system genres, which, again, may lead to a more relevant assessment of ASR performance.

Acknowledgments

The effort described here has been sponsored by the U.S. Army. Any opinions, content or information presented does not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Priti Aggarwal, Ron Artstein, Jillian Gerten, Athanasios Katsamanis, Shrikanth Narayanan, Angela Nazarian, and David Traum. 2012. The Twins corpus of museum visitor questions. In *LREC-2012*, Istanbul, Turkey, May.

Hua Ai, Antoine Raux, Dan Bohus, Maxine Eskenazi, and Diane Litman. 2007a. Comparing spoken dialog corpora collected with recruited subjects versus real users. In *The 8th SIGdial Workshop on Discourse and Dialogue*.

Hua Ai, Antonio Roque, Anton Leuski, and David Traum. 2007b. Using information state to improve dialogue move identification in a spoken dialogue system. In *Proceedings of the 10th Interspeech Conference*, Antwerp, Belgium, August.

James F. Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *IUI*, pages 1–8.

Hiyan Alshawi. 2003. Effective utterance classification with unsupervised phonotactic models. In *HLT-NAACL 2003*, pages 1–7, Edmonton, Alberta, May.

Apple Inc. 2012. Mac basics: Dictation (Technote HT5449), November.

R. Artstein, S. Gandhe, J. Gerten, A. Leuski, and D. Traum. 2009a. 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, Berlin.

Ron Artstein, Sudeep Gandhe, Michael Rushforth, and David R. Traum. 2009b. Viability of a simple dialogue act scheme for a tactical questioning dialogue system. In *DiaHolmia 2009: Proceedings of the 13th Workshop on the Semantics and Pragmatics of Dialogue*, page 43–50, Stockholm, Sweden, June.

Ron Artstein, Michael Rushforth, Sudeep Gandhe, David Traum, and MAJ Aram Donigian. 2011. Limits of simple dialogue acts for tactical questioning dialogues. In *7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Barcelona, Spain, July.

Ron Artstein. 2011. Error return plots. In *12th SIGdial Workshop on Discourse and Dialogue*, Portland, OR, June.

Linda Bell and Joakim Gustafson. 2003. Child and adult speaker adaptation during error resolution in a publicly available spoken dialogue system. In *INTERSPEECH* (ISC, 2003).

Dan Bohus and Alexander I. Rudnicky. 2003. Ravenclaw: dialog management using hierarchical task decomposition and an expectation agenda. In *INTERSPEECH* (ISC, 2003).

Sudeep Gandhe, Nicolle Whitman, David R. Traum, and Ron Artstein. 2009. An integrated authoring tool for tactical questioning dialogue systems. In *6th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Pasadena, California, July.

Sudeep Gandhe, Michael Rushforth, Priti Aggarwal, and David R. Traum. 2011. Evaluation of an integrated authoring tool for building advanced question-answering characters. In *Proceedings of Interspeech-11*, Florence, Italy, 08/2011.

Sudeep Gandhe. 2013. *Rapid prototyping and evaluation of dialogue systems for virtual humans*. Ph.D. thesis, University of Southern California.

Liza Hassel and Eli Hagen. 2006. Adaptation of an automotive dialogue system to users' expertise and evaluation of the system. *Language Resources and Evaluation*, 40(1):67–85.

Quiang Huang and Stephen Cox. 2006. Task-independent call-routing. *Speech Communication*, 48(3–4):374–389.

D. Huggins-Daines, M. Kumar, A. Chan, A.W. Black, M. Ravishankar, and A.I. Rudnicky. 2006. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I.

ISCA. 2003. *8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003, Geneva, Switzerland, September 1-4, 2003*.

Kristiina Jokinen and Kari Kanto. 2004. User expertise modeling and adaptivity in a speech-based e-mail system. In Donia Scott, Walter Daelemans, and Marilyn A. Walker, editors, *ACL*, pages 87–94. ACL.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340, September. Special Issue on Spoken Language Dialogue System Engineering.

Anton Leuski and David R. Traum. 2011. NPCEditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32:42–56.

Anton Leuski, Brandon Kennedy, Ronakkumar Patel, and David Traum. 2006a. Asking questions to limited domain virtual characters: How good does speech recognition have to be? In *25th Army Science Conference*.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006b. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27.

Fabrizio Morbini, Kartik Audhkhasi, Ron Artstein, Maarten Van Segbroeck, Kenji Sagae, Panayiotis S. Georgiou, David R. Traum, and Shrikanth S. Narayanan. 2012. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *SLT*, pages 49–54. IEEE.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, December.

S.M. Robinson, A. Roque, A. Vaswani, D. Traum, C. Hernandez, and B. Millspaugh. 2006. Evaluation of a spoken dialogue system for virtual reality call for fire training. In *25th Army Science Conference*, Orlando, Florida, USA.

S. Robinson, D. Traum, M. Ittycheriah, and J. Henderer. 2008. What would you ask a conversational agent? Observations of human-agent dialogues in a museum setting. In *Proc. of Sixth International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

A. Roque, A. Leuski, V. Rangarajan, S. Robinson, A. Vaswani, S. Narayanan, and D. Traum. 2006. Radiobot-CFF: A spoken dialogue system for military training. In *Proc. of Interspeech*, Pittsburgh, Pennsylvania, USA.

William Swartout, David Traum, Ron Artstein, Dan Noren, Paul Debevec, Kerry Bronnenkant, Josh Williams, Anton Leuski, Shrikanth Narayanan, Diane Piepol, Chad Lane, Jacquelyn Morie, Priti Aggarwal, Matt Liewer, Jen-Yuan Chiang, Jillian Gerten, Selina Chu, and Kyle White. 2010. Ada and Grace: Toward realistic and engaging virtual museum guides. In Jan Allbeck, Norman Badler, Timothy Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Intelligent Virtual Agents: 10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20–22, 2010 Proceedings*, volume 6356 of *Lecture Notes in Artificial Intelligence*, pages 286–300. Springer, Heidelberg.

David R. Traum, Stacy Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. 2008. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *IVA*, pages 117–130.

David Traum, Priti Aggarwal, Ron Artstein, Susan Foutz, Jillian Gerten, Athanasios Katsamanis, Anton Leuski, Dan Noren, and William Swartout. 2012. Ada and grace: Direct interaction with museum visitors. In *The 12th International Conference on Intelligent Virtual Agents (IVA)*, Santa Cruz, CA, September.

David Traum. 2003. Semantics and pragmatics of questions and answers for dialogue agents. In *proceedings of the International Workshop on Computational Semantics*, pages 380–394.

Ye-Yi Wang, A. Acero, and C. Chelba. 2003. Is word error rate a good indicator for spoken language understanding accuracy. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '03)*, pages 577–582.

Xuchen Yao, Pravin Bhutada, Kallirroi Georgila, Kenji Sagae, Ron Artstein, and David R. Traum. 2010. Practical evaluation of speech recognizers for virtual human dialogue systems. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *LREC*. European Language Resources Association.