Speed vs. Accuracy: Designing an Optimal ASR System for Spontaneous Non-Native Speech in a Real-Time Application

Alexei V. Ivanov[†], Patrick L. Lange[†], David Suendermann-Oeft[†], Vikram Ramanarayanan[†], Yao Qian[†], Zhou Yu[‡] and Jidong Tao[*]

Abstract Automatic dialog interaction with remote interlocutors is a difficult application area for speech recognition technology because of the limited acoustic context, poor signal representation, high variability of spontaneous speech and limited time available to do the recognition of noncanonical spoken production. We present the speech recognition system for the non-native dialog applications that we are currently developing. We find that our system broadly matches human performance; that minimum Bayes risk decoding improves accuracy, and that the posterior probabilities have good power towards predicting errors. We also explore the temporal distribution of errors made by the recognizer with online speaker adaptation, the frequency of errors among auto-semantic and function words, as well as the distribution of error rates among the heterogeneous speaker population. Our findings motivate further development directions for dialog speech recognition systems.

1 Introduction

Applications that require real-time voice-enabled interactions, such as spoken dialog systems (SDSs), present several interesting design challenges. Foremost among these is optimizing the tradeoff between speed and accuracy. In dialog environments, the automatic speech recognizer (ASR) needs to propagate as quickly as possible the hypothesis about what was just said by the human interlocutor. At the same time, this hypothesis must be a close match to what the speaker has actually said.

We are mainly interested in real-time, speech-enabled educational learning and assessment applications for non-native speakers. An example for such applications is *Subarashii*, an interactive dialog system for learning Japanese [2, 4]. Subarashii's ASR component was built using the HTK speech recognizer [30] with both native and non-native acoustic models. In general, the performance of the system after spoken language understanding (SLU) was good for in-domain utterances, but not for out-of-domain utterances. Two other examples, Robot Assisted Language Learning [3] and computer-assisted language learning applications for Korean-speaking learners of English [26], demonstrated that acoustic models trained on the Wall

[†] Educational Testing Service R&D, 90 New Montgomery St, # 1500, San Francisco, CA
e-mail: {aivanou, plange, suendermann-oeft, vramanarayanan, yqian
[‡] Carnegie Mellon University, Pittsburgh, PA e-mail: {zhouyu}@cs.cmu.edu
[*] Educational Testing Service R&D, 600 Rosedale Road, Princeton, NJ e-mail: {jtao}@ets.org

Street Journal corpus with an additional 17 hours of Korean children's English speech for adaptation produced a word error rate (WER) of as low as 22.8% across multiple domains.

In our recent work [13, 12], we have investigated the online and offline performance of a Kaldi-based large vocabulary continuous speech recognition system in conjunction with the open-source and distributed HALEF spoken dialog system [27]. Already existing APIs for speech recognition e.g. Google's Speech API[1] [18] or Microsoft's Speech API[2] are not suitable as ASR system in our SDS. They are closed source and give no insights about the algorithms and models they use. Furthermore, these services do not come with the flexibility to use specialized acoustic and language models, for example, for better recognition of non-native speakers in a particular language or domain. There may also be a concern about data privacy when using services not hosted on one's own premises. Finally, fairness among test takers in an assessment application cannot be guaranteed without access to these models.

The WAMI Toolkit [9] is an attempt to resolve the lack of open-source services and provides tools to develop, deploy and evaluate web-accessible, multi-modal interfaces including speech recognition. It allows for an easy integration of speech related services such as speech to text or text to speech conversion into web applications. However, it makes heavy use of web technologies such as Adobe Flash and AJAX, thus, it cannot be easily integrated inside a stand-alone SDS, which is connected to the regular telephony network (PSTN), uses VoIP, or WebRTC-based streamed audio.

With that motivation, our goal within this paper is to present a cloud-based, highly adjustable and open-source ASR server. After laying out the operational requirements and constraints for such a system, we present our ASR architecture. We perform an analysis of speed constraints on accuracy. We analyze in detail the error distribution over content and function phrases, over speakers as well as over time. Finally, we evaluate the predictive power of our confidence scores in our application scenario.

2 System Description, Architecture and Statistical Modeling

An ASR system used inside an SDS has to fulfill more requirements than an isolated ASR system whose accuracy is traditionally the only measure of interest. In dialog applications, however, another important priority is recognition speed. The SDS needs to respond in a timely manner, ideally within three seconds after the human input, or the interaction becomes overly tiresome for the human interlocutor [6, 28, 25] and may also severely influence the naturalness of the conversation. The The standard metric used to quantify the speed of an ASR system is the real-time factor (xRT), defined as the ratio between the time it takes to process the input and the duration of the input. If the real-time factor is 1 or below 1 the input is considered to be processed in real-time. In our applications, we expect human utterances to be

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

[2] https://www.projectoxford.ai/speech

up to 10 seconds long. As ASR is the most computationally expensive step in the dialog cycle, an approximate upper bound of recognition speed is $xRT \approx 1.3$, given that the recognition starts simultaneously with the speech.

Furthermore, the ASR system needs to integrate well into the architecture of the SDS. To achieve compatibility with most systems, we require it to use open communication standards. Moreover, the ASR system has to be able to perform concurrent recognitions because in a multi-user scenario it is not feasible to instantiate arbitrarily many copies of the ASR system due to the large amount of memory, required for each of the loaded models. Finally, reasonable accuracy is still required. SDSs can recover from some recognition errors because most dialog applications rely on the SLU instead of the ASR result. Furthermore, the dialog flow can be altered (e.g. inserting re-prompts, clarification questions and confirmation requests) if the confidence in the recognition hypothesis is too low. The latter, however, requires a reliable prediction of errors. Application–specific recognition accuracy may be increased by using application–specific models. Language models adapted to the application domain have been shown to outperform general models [1]. Custom acoustic models can be used to support new languages or optimize performance for particular acoustic environments (e.g. noise) and speaker groups (e.g. dialects). In one of our applications for instance, we employ an acoustic model optimized for telephony speech and non-native speakers.

Our ASR is based on the Kaldi Toolkit [24]. This choice is motivated by a recent study [7] that found that Kaldi significantly outperformed other open-source recognizers on German Verbmobil and English Wall Street Journal corpora. The Kaldi online ASR was also shown to outperform the Google ASR API [18] when integrated into the Czech ALEX spoken dialog framework [22]. A recent study comparing several popular ASRs such as Kaldi [24], Pocketsphinx [11] and cloud-based APIs from Apple[3], Google and AT&T[4] in terms of their suitability for use in SDSs, [21] found no particular consensus on the best ASR, but observed that Kaldi performed well in comparison with the other closed-source industry-based APIs. Furthermore, the Kaldi Toolkit provides the necessary tools to train acoustic and language models and low level features are accessible.

To ensure we fulfill the speed constraints outlined in Section 2, we need to start the recognition ideally at the same time the speech starts and process it with similar pace as it progresses. Therefore, we have implemented a streaming web-based ASR service. A dialog application connects via a WebSocket connection to the dedicated remote ASR server that starts speech processing as soon as the audio becomes available. To handle multi-client recognition, the ASR server has been implemented as a multithreaded process: a unique listener thread that is responsible for the communication and audio chunk en-queuing; a collection of ASR threads that do the actual processing and hypothesis generation while serving several simultaneous clients; and a timer thread that sets the pace of the recognition critical cycle (see Figure 1 for a graphical representation of this architecture).

[3] Apple's Dictation is an OS level feature in both Mac OS X and iOS.

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

Data communication is organized through a shared memory queue. Constant ASR-related objects, such as the acoustic model and the decoding weighted finite state transducer, are globally visible throughout all of the threads, while the local utterance-related context is specific to the instance of the ASR thread that processes the given utterance.

Speaker and channel adaptation has to be performed online, while concurrently recognizing the incoming speech, i.e. the entire ASR system must be able to produce a hypothesis after a single pass through the data stream. No additional passes are allowed unless they require only a minor additional delay. A system based on i-vectors [17] as a method of adapting the deep neural network (DNN) based acoustic model to the speaker that satisfies that requirement has been proposed in [32].

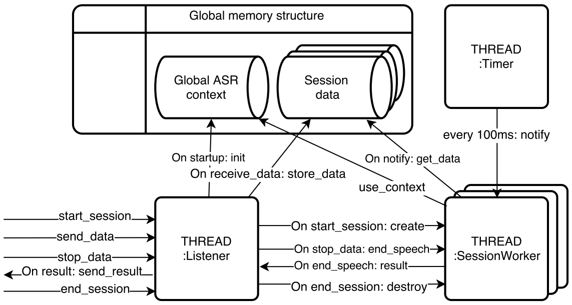

Fig. 1 Multithreaded ASR Architecture

Our acoustic model was trained using a standard Kaldi model generation pipeline based on approximately 760 hours of spontaneous vocal productions obtained from language learners in the scope of large-scale internet-based assessments of academic English. Although the dataset is not strictly collected from our prospective application domain, we believe it serves as a reasonable engineering approximation at the current stage of model development. We are working on collection of more application–specific data that will serve as our training set in the future versions of our system. In order to comply with the SDS design requirements, the data was down-sampled to 8 kHz. We used standard 13-dimensional MFCCs with deltas and delta-deltas and 10ms shift. The final acoustic model is a p-norm DNN [32] with 4 hidden layers, a dimensionality of the input/output layer of 2000/250 and was trained in 8 epochs. The system's phonetic alphabet is comprised of 42 basic tokens combining 39 "true" phonemes, and tokens for "silence", "spoken noise" and "noise". Additionally, the final phonological tokens have word position-specific modifiers for internal, singleton, word-beginning and word-ending positions.

The language model was estimated on the manual transcriptions of the same training corpus consisting of ≈ 5.8 million tokens and finally was represented as a tri-gram language model with ≈ 525 thousand tri-grams and ≈ 605 thousand bi-grams over a lexicon of ≈ 23 thousand words which included entries for the most frequent partially produced words. Ultimately, the final decoding graph was compiled having approximately 5.5 million states and 14 million arcs.

3 Experiments

We test our recognizer on a physical computer, that has an Intel(R) Core(TM) i7-4930K CPU running at 3.40GHz. This CPU is built as a six-core processor with the Ivy Bridge-E architecture. The computer has 16 GB of RAM and works under the Ubuntu v14.04 operating system. The choice of the hardware is essential to ensure that our *xRT* measurements are performed with a state-of-the-art computing system.

For evaluation we have used two sets coming from 100 different speakers and exceeding 9 hours of audio each: The development set (DEV), containing 593 utterances (68329 tokens, 3575 singletons, 0% OOV rate) and the test set (TST), that contains 599 utterances (68112 tokens, 3709 singletons, 0.18% OOV rate). Utterances in the corpus are quasi-spontaneous monologs responding to six different test questions covering two different speaking tasks: 1) providing an opinion based on personal experience and 2) summarizing or discussing material provided in a reading and/or listening passage. Maximum utterance duration is one minute. The average speaking rate is about 2 words per second. Every speaker produces up to six such utterances. Speakers had a brief time to familiarize themselves with the task and prepare an approximate production plan.

3.1 Speed–Accuracy Trade-off

Depending on the chosen parameter set (width of the pruning beam in decoding and lattice generation; the maximum number of concurrent hypotheses), the recognizer is capable of operating with various accuracy–processing speed tradeoffs. Usually systems with wider pruning beams are slower and more accurate. If one continues to increase the pruning beam width, the recognizer's accuracy saturates at some point and attains its maximum. Further increase in the beam width slows the system down even more without significant improvements in accuracy. We have experimentally found for our speech recognizer to be realtime-able it has to be moved significantly away from the highest accuracy point (see Table 1 for details). Essentially at the optimal operating point, the recognizer accuracy is bounded by the inability to consider all possible hypotheses, rather than modelling imperfections. This might be due to the inherent confusability of the alternatives, i.e. compared to the native speech case, recognition of non-native, heavily accented speech, is a more difficult task with more inherent confusion. Or it may also be that the complexity of our statistical models was not adequately chosen.

We have observed that taking the first best hypothesis from the confusion network constructed from the output word lattice provides a small but very consistent improvement in word error rate regardless of the specific system configuration. In constructing a confusion network [19] we follow the minimum Bayes risk decoding approach implemented in Kaldi [8, 29].

For reference, in Table 1 we also put the performance of a baseline DNN-based multi-pass speaker-adapting recipe of Kaldi, prepared on the same training data. As it is evident from the given comparison, the i-vector system is better in both accuracy and processing speed.

Set	Adaptation	Prunning Beam	Hypotheses Number	WER	xRT
DEV	fMLLR	various	various	22.27%	3.44
DEV	Online	50	40 K	21.58%	$>$ 25
DEV	Online	11	7 K	21.95%	1.26
TST	Online	11	7 K	23.05%	1.28

Table 1 Accuracy and speed of various recognizers.("Online" – the i-vector based online adaptation or "fMLLR" - the best operating point of a DNN-based standard multi-pass speaker adapting recipe of Kaldi for WSJ corpus).

3.2 Comparison with Human Performance

With the TST set WER of about 23.05% our proposed system has reached the level of broadly defined average human accuracy in the task of non-native speech transcription. In fact, experts have average WER around 15% [31] while crowd-sourcing workers perform significantly worse at around 30% WER [5]. Besides being accurate our system is capable to achieve that performance in real-time (xRT $\approx$ 1.28).

Matching human performance in the dialog context is a task that requires designing individual sub-systems (speech recognition, natural language understanding, dialog management) with a clear understanding of application–specific constraints and exploiting inherent possibilities. We see a possibility to further improve performance of our SDS in general and ASR in particular specifically by a) mimicking human strategies to handle ambiguous semantic context in the dialog; b) developing statistical models for other existing knowledge sources (e.g. grammar, semantics and pragmatics) and incorporating those into the process of hypothesis refinement; and c) exploring rapid topic domain adaptation. The remainder of the paper is devoted to the error analysis that motivates these improvements.

3.3 Error Distribution over Speakers

The speech recognition system cannot provide identical recognition accuracy to all potential interlocutors. There is an inherent variability in proficiency levels among language learners. The acoustic environment is not always constant either. ASR accuracy has to be studied as a distribution that is estimated on a broad target speaker population. Sensitivity of the WER to the interlocutor's proficiency is yet another quality measure for the ASR in language assessment applications.

Results of such an analysis are presented in Figure 2. The shape of the distribution (a skewed Gaussian) implies that there exists a systematic limiting factor precluding our ASR from sometimes showing low WERs. The estimated standard deviations are 12.35% and 8.59% for DEV and TST sets respectively.

For the system to be fair, a stratification over any of the social groupings, e.g. race, gender, geographical location, native language, etc., shall not lead to a statistically significant alternation of the distribution in Figure 2. That is true under the assumption of conditional independence of speaker proficiency over the above mentioned properties.

To assess fairness of the system to language learners coming from different geographical locations, we have stratified the joint collection of the DEV and TST sets into groups, specific to broad geographical regions. The stratification's coarse granularity was dictated by the necessity to retain sufficient and approximately equal statistics for each of the geographical subgroups.

With the help of the dual Kolmogorov-Smirnov non-parametric test [14] we have estimated the probability that each of the regional sub-group WERs is distributed in a similar way to the joint set. As is evident from Table 2, our current ASR system is not fair towards some of the sub-groups, e.g. Chinese English is recognized differently with statistical significance ($p < 0.05$). More specifically, our ASR system tends to produce less errors than in general when subjected with English utterances of language learners from China. We explain that fact with overly large proportion of the Chinese language learners in the training corpus. A better job needs to be done to properly select training material for the speech recognition system.

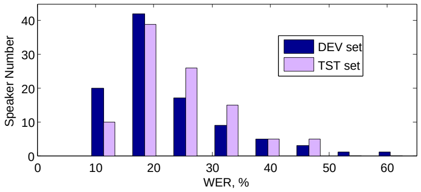

Fig. 2 WER distribution across different speakers.

Region	Speakers	p-value
Africa	10	0.84
South-East Asia	27	0.78
India	17	0.78
Americas	20	0.74
Europe and Central Asia	36	0.56
Middle East	28	0.31
Korea	30	0.13
China	27	0.02

Table 2 Regional ASR bias ("p-value" - significance level of the hypothesis that regional and global WER samples are drawn from the same distribution).

3.4 Error Distribution Over Word Type

Importance of an individual recognition error towards the general understanding of the interlocutor's input is not constant. Traditionally English words are divided into two broad classes: content (or auto-semantic) words that entail a distinct semantic concept; and function words that have little or ambiguous lexical meaning, but instead serve to express grammatical relationships of other words within a sentence. To adapt this classification to spontaneous spoken language we augment the above classification with a joint group of common lexicalized interjections (e.g. "yeah", "boo", "oops", etc.) and fillers (e.g. "um", "ah"). The system's lexicon, thus, contains 319 distinct function words and 24 interjections and fillers. The remaining 22800 lexicon entries are content words.

Table 3 reflects word-class and error occurrence statistics within the test set. While being an extremely small lexical set, function words are more frequent than content words in natural language. Content word recognition is a more difficult problem in information-theoretical terms (e.g. the task of choosing 1 out of 22800 requires significantly more information than 1 out of 319). The apparent raw rates of mis-recognition (substitutions and deletions combined) of content and function words are similar. It suggests that some of the function word errors can be recovered by applying a content-conditioned re-scoring model that encapsulates grammatical rules of the language. We explain the reduced content word insertion rate by the fact that English content words follow the minimal word constraint [20] and generally have larger phonetic support compared to function words. Exclusion of function words, fillers and interjections reduces the mis-recognition rate by half and makes the insertion rate five times smaller.

TST Set	**Total Words**	**Content Words**	**Function Words**	**Fillers+Interject.**
Reference token count	67864	24596	37522	5746
Insertion count	2836	575	1649	612
Insertions, %	4.18%	**0.85%**	2.43%	0.90%
Mis-recocgnition count	12809	6740	5357	712
Mis-recocgnitions, %	18.87%	**9.93%**	7.89%	1.05%

Table 3 Error distribution among different types of words for minimum Bayes risk decoding system evaluated on the test data.

The WER within the content word set can serve as a baseline for SLU model development and evaluation. This baseline assumes existence of a one-to-one mapping between content words and concepts. The mapping is also assumed to have no contextual dependency.

3.5 Error Distribution Over Time

It is interesting to observe the difference in error distribution through time for the systems with online and offline speaker adaptation. The overall balance of errors is presented in Table 4. A more detailed picture of error distribution through time can be found in Figure 3. The figure contains an estimate of the probability of the error to occur in the vicinity of a certain time instance ($P(error|T+dt)$ in Figure 3). As timing of a deletion is inherently ambiguous, only substitution and insertion statistics are used in this figure. We limit the figure to the first 40 seconds of the utterance, where the amount of the test material is maximal and the collected statistics are sufficient for a reliable estimate of the probability.

The system with online speaker adaptation has a higher probability to make an error in general (Table 4) and this probability is specifically higher in the beginning of the utterance (circled in Figure 3). This observation can be explained by the fact that initially the online speaker adaptation procedure has very little data to work with. Online speaker adaptation performs worse than offline adaptation during the first 15 seconds of an utterance. This is larger than the expected duration of a typical spoken response in our SDS. The design option to hide an increased speaker adaptation analysis window behind the response latency is, thus, ruled out. In order to achieve optimal speaker adaptation performance, there is a need to maintain the speaker adaptation profile through the whole dialog interaction.

Until the ASR gathers enough data for adaptation the dialog complexity has to be controlled by the SDS. E.g. the human interlocutor dialog act should be elicited by the dialog system in such a way that there is a possibility to interpret it with a low-perplexity model; the response of the dialog system has to be pragmatically correct regardless of the human input.

The figure also shows that the probability to make an error is not constant over time for both systems. It remains to be seen if this behavior is an indication that the system's error rate depends on the rhetorical structure of the utterance or is due to other particular properties of the experimental material. Both language and acoustic modelling (LM and AM) can result in the WER dependency on the rhetorical structure. E.g. the context is less certain in the sentence beginning, which will result in decreased performance of the history-based n-gram LM. We have also seen in past experiments that acoustic properties of oral production chunks depend on their salience [15]. That observation corroborates the effort code of Gussenhoven [10, 23]. The AM may be failing for non-salient speech chunks.

Set	DEV	DEV
Adaptation	**Online**	Offline
Word Error Rate,%	**21.90**	21.74
Substitutions, %	**12.41**	12.25
Deletions, %	**5.93**	5.96
Insertions, %	**3.56**	3.54
xRT	**1.26**	1.15

Table 4 Overall accuracy of the systems used in the comparative error timing analysis.

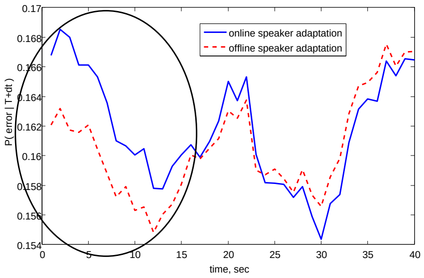

Fig. 3 Error timing analysis

3.6 Error Detection

The ability to predict its own errors is essential for an SDS ASR system. It helps in operation, e.g. to trigger a re-prompt or confirmation for content that was not recognized as sufficiently unambiguous, as well as during training to reduce dependency of the system development on human supervision. This ability is equivalent to the possibility of accurately estimating the probability of being correct under a selected hypothesis, i.e. to estimate the statistical confidence level [16]. If the confidence estimate is sufficiently accurate, we may set a boundary threshold parameter for a rejection subsystem that will deem all recognitions falling below the threshold as unreliable, while asserting those above the threshold as correct.

The most accessible form of the confidence measure in our present system is in the estimates of the posterior probabilities of word alternatives in the confusion network that our recognizer generates. It is interesting to learn how much of the error detection predictive power is contained in this unsophisticated statistical confidence measure applied to the task of spontaneous non-native speech recognition. Figure 4 presents a detection error trade-off (DET) curve for an error detection (rejection) system, that takes a posterior of the word hypothesis as an input. The colored line in this plot represents a plurality of individual operation points. Each point is a trade-off between making two types of error: calling an error to be a correct recognition (false acceptance) and discarding a valid hypothesis as an error (false rejection).

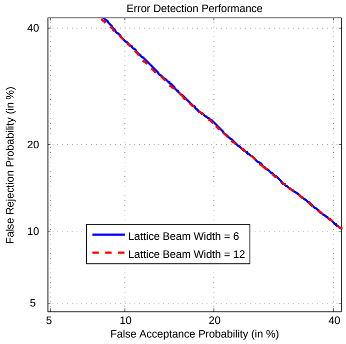

Fig. 4 Recognition error rejection DET curve

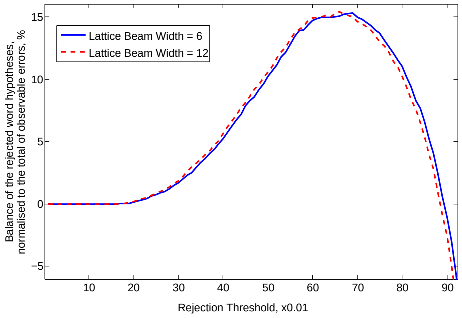

Fig. 5 Error rejection operating point selection

For instance, according to this figure, our system is capable of rejecting $\approx 60\%$ of errors at the cost of falsely rejecting $\approx 10\%$ of correct recognitions. Each individual operation point is optimal for a certain loss function.

If we choose our loss function to be proportional to the total number of errors, false acceptances and false rejections, assuming that the error rates during the operation are going to be the same as we have observed in our development data, we may specify a particular optimal operating point for the rejection system. Figure 5 depicts a balance between the correctly and incorrectly rejected hypotheses. In this figure, each correct rejection increments the balance while false rejection decrements it. The balance is finally scaled relative to the total amount of errors, observable by the recognition system (substitutions and insertions). The balance attains the extremum at $\approx 15\%$ of the error count with the threshold in the range $0.65-0.68$. At this point the system rejects $\approx 43.5\%$ of true errors and $\approx 5.8\%$ of correct recognitions. The recorded level of rejection performance is good to serve as a baseline in our further studies. If the confidence estimate is not good enough the balance curve might not have an extremum at all. It might be monotonously decreasing if the number of false rejections is always larger than the number of correct rejections for any value of the threshold. With the same set of operating parameters on the TST set the system rejects $\approx 44.11\%$ of true errors and $\approx 6.38\%$ of correct recognitions.

In both figures (Figure 4 and 5), we show the results for two systems: one with the pruning beam during the confusion network generation twice as large as that of the other. The system with the larger beam is more computationally complex and produces larger lattices with a bigger number of alternative transcriptions. Although the large beam system is much slower, the DET curves of each system are barely distinguishable. The balance curve for the larger beam case is slightly shifted towards the zero threshold value and there is a tiny increase in the balance maximum. The shift can be explained by the fact that the larger beam results in more populated lattices, i.e. a larger expected number of alternative transcriptions. The larger expected number of alternative transcriptions in turn reduces the expected posterior probability estimate in probability mass re-normalization during the confusion network construction. The absence of the significant increase in the balance allows us to conclude that a more accurate posterior estimation does not lead to increased rejection performance and the narrower, more computationally efficient beam is sufficient for our purposes.

4 Conclusions

We have seen that building a fast and accurate dialog speech recognition system for interacting with distant non-native interlocutors is possible. Our near real-time system performs better than non-specialist human transcribers and not far from the human expert performance level.

The DNN with i-vector-based speaker adaptation for acoustic modelling allows us to reach the state-of-the-art acoustic decoding accuracy with single-pass processing. However, due to the lack of observation statistics, the online speaker adaptation is not efficient during the initial 15 seconds of the interaction.

Word posterior probabilities in confusion networks have been observed to have good power towards predicting erroneous recognitions. The rejection model is capable of correctly predicting $\approx 44.11\%$ observable recognition errors at the cost of falsely rejecting $\approx 6.38\%$ of correctly recognized words. The reported rejection performance is measured in the system that satisfies the real-time requirements.

The analysis of error distribution across auto-semantic and function words roughly estimates the upper bound of the improvement in WER that can be achieved with the gramatical re-scoring model. The main impact of such model should be on the recognition of function words and potentially can be as big as 40% of the total WER.

Studying the WER distribution across different speaker populations, we find that a better job needs to be done in collecting the training data to ensure fairness of the resulting system towards various possible target user sub-groups.

References

1. Bellegarda, J.R.: Statistical language model adaptation: review and perspectives. Speech communication **42**(1) (2004)
2. Bernstein, J., Najm, A., Ehsani, F.: Subarashii: Encounters in japanese spoken language education. CALICO journal **16**(3), 361–384 (1999)
3. Dong-Hoon, A., Chung, M.: One-pass semi-dynamic network decoding using a subnetwork caching model for large vocabulary continuous speech recongnition. IEICE Transactions on Information and Systems **87**(5), 1164–1174 (2004)
4. Ehsani, F., Bernstein, J., Najmi, A.: An interactive dialog system for learning Japanese. Speech Communication **30**(2), 167–177 (2000)
5. Evanini, K., Higgins, D., Zechner, K.: Using Amazon mechanical turk for transcription of non-native speech. In: Proc. of the NAACL HLT. Los Angeles, CA, USA (2010)
6. Fried, J., Edmondson, R.: How Customer Perceived Latency Measures Success In Voice Self-Service. Business Communications Review **36**(3) (2006)
7. Gaida, C., Lange, P.L., Petrick, R., Proba, P., Malatawy, A., Suendermann-Oeft, D.: Comparing Open-Source Speech Recognition Toolkits. Tech. rep., DHBW Stuttgart, Stuttgart, Germany (2014)
8. Goel, V., Byrne, W.J.: Minimum Bayes-risk automatic speech recognition. Computer Speech & Language **14**(2) (2000)
9. Gruenstein, A., McGraw, I., Badr, I.: The WAMI toolkit for developing, deploying, and evaluating web-accessible multimodal interfaces. In: Proc. of the ICMI. Chania, Greece (2008)

10. Gussenhoven, C.: Intonation and Interpretation: Phonetics and Phonology. In: Proc. of Speech Prosody. Aix-en-Provence, France (2002)
11. Huggins-Daines, D., Kumar, M., Chan, A., Black, A., Ravishankar, M., Rudnicky, A.: Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. In: Proc. of the ICASSP. Toulouse, France (2006)
12. Ivanov, A.: Speech Recognition on GPUs with Open-Source Models: Faster, Better, Cheaper. In: Proc. of the GTC. San Jose, USA (2015)
13. Ivanov, A., Ramanarayanan, V., Suendermann-Oeft, D., Lopez, M., Evanini, K., Tao, J.: Automated Speech Recognition Technology for Dialogue Interaction with Non-Native Interlocutors. In: Proc. of 16th Annual SIGdial Meeting on Discourse and Dialogue (SIGDial'2015). Prague, Czech Republic (2015)
14. Ivanov, A., Riccardi, G.: Kolmogorov-Smirnov Test for Feature Selection in Emotion Recognition from Speech. In: Proc. of the IEEE ICASSP, International Conference. Kyoto, Japan (2012)
15. Ivanov, A., Riccardi, G., Ghosh, S., Tonelli, S., Stepanov, E.: Acoustic Correlates of Meaning Structure in Conversational Speech. In: Proc. Interspeech'2010, International Conference. Makuhari, Japan (2010)
16. Jiang, H.: Condence measures for speech recognition: A survey. Speech Communication **45**(4) (2005)
17. Kenny, P.: A small footprint i-vector extractor. In: Proc. of the Odyssey. Singapore, Singapore (2012)
18. Lange, P.L., Suendermann-Oeft, D.: Tuning Sphinx to outperform Google's Speech API. In: Proc. of the ESSV. Dresden, Germany (2014)
19. Mangu, L., Brill, E., Stolcke, A.: Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Computer Speech & Language **14**(4) (2000)
20. McCarthy, J., Prince, A.: Prosodic morphology 1986. University of Massachusetts, Amherst, MA, USA (1996)
21. Morbini, F., Audhkhasi, K., Sagae, K., Artstein, R., Can, D., Georgiou, P., Narayanan, S., Leuski, A., Traum, D.: Which ASR should I choose for my dialogue system. In: Proc. of the SIGDIAL. Metz, France (2013)
22. Plátek, O., Jurčíček, F.: Integration of an On-line Kaldi Speech Recogniser to the Alex Dialogue Systems Framework. In: Proc. of the TSD. Brno, Czech Republic (2014)
23. Post, B., Brechtje, D., Gussenhoven, C.: Fine Phonetic Detail and Intonational Meaning. In: Proc. of 16th Int. Congress of Phonetic Sciences (ICPhS). Saarbrüken, Germany (2007)
24. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi Speech Recognition Toolkit. In: Proc. of the ASRU. Hawaii, USA (2011)
25. Shigemitsu, Y.: Different Interpretations of Pauses in Natural Conversation - Japanese, Chinese and Americans. Academic Report **27**(2) (2005)
26. S.Lee, Noh, H., Lee, J., Lee, K., G.Lee: POSTECH approaches for dialog-based english conversation tutoring. Proc. APSIPA ASC pp. 794–803 (2010)
27. Suendermann-Oeft, D., Ramanarayanan, V., Teckenbrock, M., Neutatz, F., Schmidt, D.: HALEF: An Open-Source Standard-Compliant Telephony-Based Modular Spoken Dialog System - A Review and an Outlook. In: Proc. of the IWSDS. Busan, South Korea (2015)
28. Wennerstrom, A., Siege, A.F.: Keeping the Floor in Multiparty Conversations: Intonation, Syntax, and Pause. Discourse Processes **36**(2) (2003)
29. Xu, H., Povey, D., Mangu, L., Zhu, J.: Minimum Bayes Risk Decoding and System Combination Based on a Recursion for Edit Distance. Computer Speech & Language **25**(4) (2011)
30. Young, S., Woodland, P., Byrne, W.: The HTK Book, Version 1.5. Cambridge University, Cambridge, UK (1993)
31. Zechner, K.: What did they actually say? Agreement and disagreement among transcribers of non-native spontaneous speech responses in an English proficiency test. In: Proc. of the ISCA SLaTE. Birmingham, UK (2009)
32. Zhang, X., Trmal, J., Povey, D., Khudanpur, S.: Improving Deep Neural Network Acoustic Models using Generalized Maxout Networks. In: Proc. of the ICASSP. Florence, Italy (2014)