Enhancing Speech Recognition in Fast-Paced Educational Games using Contextual Cues

Carrie J. Cai, Robert C. Miller, Stephanie Seneff

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, Massachusetts 02139, USA
`{cjcai,rcm}@mit.edu, seneff@csail.mit.edu`

Abstract

Arcade-style games like Tetris and Pacman are often difficult to adapt for educational purposes because their fast-paced intensity and keystroke-heavy nature leave little room for simultaneous practice of other skills. Incorporating spoken language technology could make it possible for players to learn as they play, keeping up with game speed through multimodal interaction. To date, however, it remains exceedingly difficult to augment fast-paced games with speech interaction because the frustrating effect of recognition errors highly compromises entertainment. In this paper, we design a modified version of Tetris with speech recognition to help students practice and remember word-picture mappings. Using utterances collected from learners interacting with the speech-enabled Tetris game, we present and evaluate several techniques for leveraging contextual cues to increase recognition accuracy in fast-paced game environments.

Index Terms: speech recognition, education, serious games, user interfaces

1. Introduction

The pervasive spread of computer games has made a significant impact on game-based learning as a serious topic in the field of education. Research evidence has shown that fun and enjoyment are central to the process of learning because they increase learners' intrinsic motivation [2,9]. Good games can motivate players to learn through repeatedly doing the game itself until they have virtually automatized the new skill [4].

Although the highly engaging, repetitive nature of existing arcade-style games makes them natural settings for embedding learning through rehearsal, most adaptations of existing games emerge from turn-based frameworks like card games [10] or from complex virtual environments [14], perhaps due to less time pressure on learners and greater amenability to structural changes. However, arcade-style games such as Tetris and Pacman are advantageous in that they are much simpler to manipulate by developers, have open source code bases, and allow a wider range of time commitment from players. Just as flashcards enable students to review vocabulary on the run, arcade games allow players to either indulge in short spurts or stay indefinitely.

Augmenting games with speech interaction offers multiple advantages for adapting such games for learning. Not only does speech production strengthen memory by providing learners with phonological input back to the mind [8], but speech is also a typically unused input channel during traditional arcade gameplay. It could therefore enable users to keep up with the original game speed more so than text input. Previous work has further indicated that embedding motivations for *retrieval practice*, the act of repeatedly attempting recall from memory, could improve long-term retention in a speech-augmented game environment [3]. However, fast-paced games offer an unusual challenge in that their motivational effectiveness depends heavily on the rhythm and flow of the game, along with clear accountability for progress [12]. The thrill of playing a fast-paced game could be seriously dampened by the frustrating effect of speech recognition errors, a reason that perhaps explains the limited adoption of speech technology in this area.

Recent work has explored using dialogue context to enhance speech understanding, both in standard information-access systems [13][16] and in dialogue systems for second language learning [15]. However, less research is devoted to enhancing speech recognition systems in time-sensitive settings for rapid gameplay. Fast-paced arcade style games may offer the advantage of providing even more fine-tuned contextual information, due to simpler game logic, fewer possible states, and a more granular trial-by-trial structure.

In this paper, we investigate useful techniques for enhancing speech recognition performance by using in-game context to provide additional information to the recognizer. We use Tetris as a prototypical example for evaluating these approaches. Tetris is classic arcade-style video game in which players prevent falling blocks from stacking to the top by rotating and maneuvering the blocks to form rows.

2. System Design

Building on an existing open source web implementation of Tetris[1], we modified traditional Tetris rules to offer an incentive for learning any set of associations, such as capitals and countries or names and faces. Our specific implementation teaches word-picture associations to help users learn and remember the meaning of words.

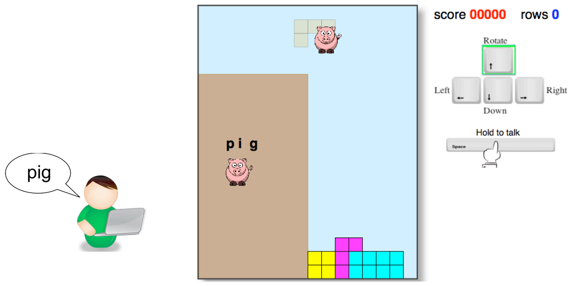

Figure 1: *Modified Tetris game interface. Saying the correct word unlocks block rotation.*

Each player sees a Tetris block attached to the picture and must correctly speak the word associated with the picture

[1] http://codeincomplete.com/posts/2011/10/10/javascript_tetris © Jake Gordon

before block rotation can be unlocked for the trial (Figure 1). As in traditional Tetris, a block can only be maneuvered while it is still falling. Once it has dropped, the next block with a new picture immediately appears. Although our specific implementation allows learners to rehearse word-picture associations, the framework is not limited to pictorial cues and can be applied to learning any set of paired associations, in either the first or second language. For example, learners could practice recalling historical events and the dates on which they occurred, or scientific terminology and their definitions.

The game can furthermore be configured in three different modes: 1) In study mode, the word associated with the picture is presented each time the picture appears. 2) In retrieval mode, learners see the word-picture pair only the first time it appears, and in subsequent trials only see the picture displayed. The word is revealed if the learner says nothing after four seconds, or as soon as the learner records a response regardless of correctness. 3) Multiple choice mode is similar to retrieval mode, except in subsequent trials learners are aided by the display of two word options to choose between rather than having to exercise free recall. In all three modes, the learner hears the pronunciation of the word when the word-picture pair is first introduced.

To recognize speech input, we used the WAMI (Web-Accessible Multimodal Interface) software [6], a framework that allows audio to be captured at the web page and transmitted to the SUMMIT speech recognizer [5] running remotely. To enhance user input efficiency, we implement the voice recording functionality via a spring-loaded hold-to-talk spacebar (Figure 1) rather than the more traditional two step process of push-to-record followed by push-to-stop.

3. Speech Data Collection

We collected speech on Amazon Mechanical Turk by inviting remote participants to play the fully speech-enabled Tetris game multiple times, in different modes. Due to poor quality microphone hardware in many older computers, only participants who passed a pre-qualifier microphone test were allowed to complete the tasks. Within each game, learners were first introduced to a word-picture pair and then rehearsed the mapping four times, totaling 35 trials for the seven words per game.

In real-life situations, a learner may wish to learn or review words that may be missing from the recognizer's existing vocabulary, such as scientific terminology or proper nouns like *peroxisome* or *Nowocin*. To model these situations, our game presented an artificial vocabulary rather than existing words in the English language. The novel word-picture mappings also precluded any user from having a learning advantage due to prior exposure. We pre-generated the artificial vocabulary using a probabilistic model[1] on English phonemes. The final vocabulary consisted of 28 English-like words (Table 1) mapped to pictures of familiar animals and household objects. The lexicon for speech recognition used an English letter-to-sound model [1].

During gameplay, each user's utterances and game activity were logged to a database. In total, we collected 2584 utterances, at a sample rate of 8 kHz, from 16 users (12 male, 4 female) between the ages of 21 and 51 (mean=31.6). All participants were native English speakers located in the United States. Data for two sessions were not evaluated due to technical difficulties expressed in the user comments in a follow-up questionnaire. We thus perform evaluation on a total of 2351 utterances.

Vocabulary Words	
wug	blicket
speff	dax
pimwit	zigant
nanose	gazzer
tusket	toma
intess	fendle
priole	moffer
unty	illo
rint	del
mata	blas
pos	omma
tranco	atter
musker	corros
henne	barnel

Table 1: *The artificial vocabulary that users learned while playing the speech-enabled Tetris game. These words were randomly mapped to common animals and household objects.*

4. Evaluation of Static Recognizer

The recognizer's performance depends critically on its letter to sound (L2S) model used to generate lexical pronunciations for each out-of-vocabulary word. To evaluate the robustness of our L2S model, we utilized different pronunciation models ranging from one to twenty-best pronunciation hypotheses.

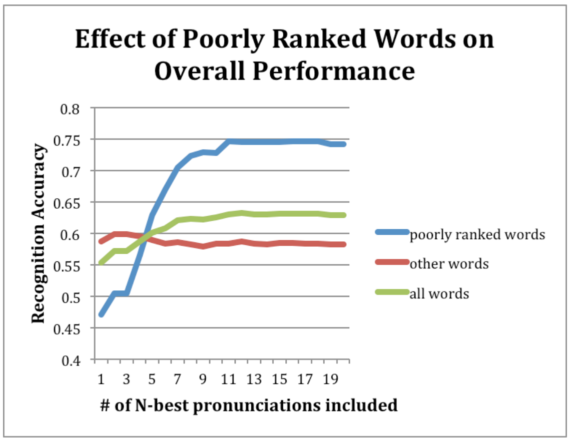

Figure 2: *Poorly ranked words (9 of 28) account for increased recognition accuracy when more pronunciations are included in the L2S.*

These N-best pronunciations were produced from the SUMMIT L2S model applied to the 28 artificial words. We configured a static recognizer with the full 28-word vocabulary and evaluated it on all utterances in which the speaker had produced any one of the 28 vocabulary words. When only one pronunciation per word was included in the L2S, recognizer performance was surprisingly low at 55%, but accuracy increased to 63% when 20 pronunciations were

[1] ibbly.com/Pseudo-words.html

included per word. Although performance for the majority of words peaked at a small number of included pronunciations, for 9 of the 28 words the most common pronunciation was ranked very low, causing overall performance on the 28 words to suffer in lexicons using only a limited number of L2S pronunciations (Figure 2). Hence, the total corpus benefited from an expansion of the lexicon to include more N-best pronunciations. The high risk of missing a key pronunciation commonly produced by users thus appears to outweigh the diluting effect of including greater pronunciation variety.

We also examined the extent to which performance could be enhanced by including L2S confidence scores for each pronunciation (Figure 3). Confidence scores [7] are used to weigh pronunciations based on their likelihood of being correct. For a benchmark comparison, we also evaluated the same corpus on a lexicon built using 1-best pronunciations manually created by an expert. Regardless of the number of pronunciations included, the expert lexicon performed better than an L2S lexicon with no confidence scoring, illustrating the disadvantage of poor pronunciations in the lexicon. However, the inclusion of L2S confidence scores produced a recognizer whose performance surpassed expert lexicon performance when the L2S model included at least ten-best pronunciations, illustrating some tangible benefit to including pronunciation variety on untrained words, particularly if confidence scores are available to down-weigh less likely pronunciation occurrences. In line with this notion, letter-to-sound confidence scores kept performance relatively steady even at the inclusion of a high number of potentially irrelevant pronunciations – a point at which lexicon performance without confidence scores had begun to drop.

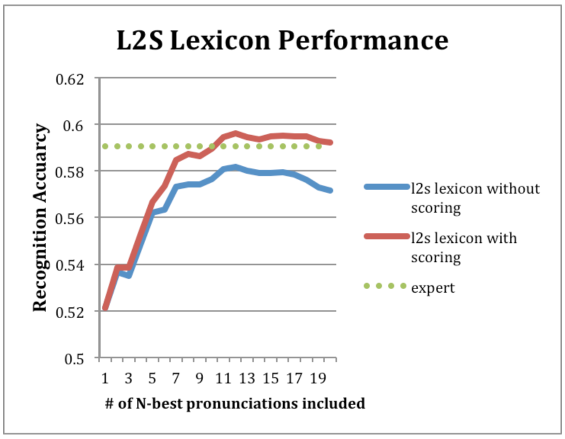

Figure 3: *Comparison of L2S performance with and without confidence scoring to an expert L2S.*

Although average recognition performance on a static 28-word recognizer was surprisingly low, recognition accuracy for the highest performing speaker was 94%, and it was above 85% for the top four speakers (Figure 4). As our user study was strictly a remote task, the remarkably wide spread among different speakers is partly due to substantial differences in microphone and hardware quality on different computers. To better understand the low average performance and high variance among speakers, we further categorized misrecognitions by false negative and false positive recognition errors. We found that the vast majority of errors were due to false negatives (85%), and only a small number were false positives (2%). The remaining errors (neither false positive nor false negative) were situations in which the learner produced the wrong utterance, but the recognizer hypothesized a third word that was neither the learner's utterance nor the target word.

Interestingly, the alarmingly high false negative rate was partially a function of in-game user behavior. Many users tended to repeat the same utterance multiple times upon experiencing a false negative error, in an attempt to resolve the recognizer's mistake. These repeated false negatives widened the performance gap between speakers because a single false negative error would almost always be exacerbated by an ensuing sequence of more false negative errors. This behavior may manifest particularly strongly in fast-paced game settings with short target utterances; the urgency associated with game incentives (i.e. Tetris blocks dropping) is complemented by the fact that one-word utterances are easy to repeat incessantly and thus worth the attempt. To discover the impact of repeated false negatives, we re-evaluated the corpus without false negatives that had been purely due to repetition, and found a 14% increase in overall recognition performance.

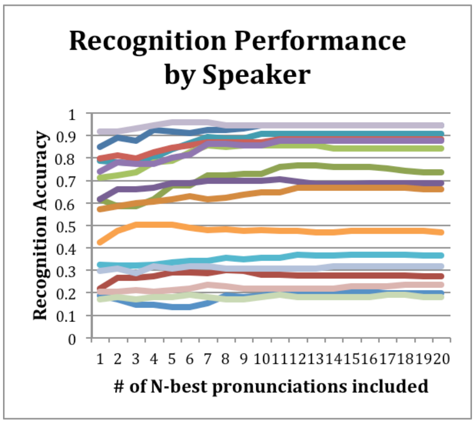

Figure 4: *Comparing recognizer performance across the 16 different speakers.*

False negative speech recognition errors also appeared to have an asymmetric impact on user enjoyment. In a post-study questionnaire on Mechanical Turk, some users reported that false negative errors inhibited their enjoyment of the game. For example, one user wrote that false negatives "made me less engaged, because I felt like [the game] was counting off for something I knew." On the other hand, false positive errors seemed to have a less detrimental effect on user enjoyment. Observations from local pilot testing revealed that false positive errors were more rare because users tended to speak only when they had some confidence or inkling of the correct answer. Moreover, because the target answer was revealed whenever the user succeeded, users often appeared amused rather than misdirected by the small number of false positives that they experienced.

The combination of time pressure and playful exploration inherent in gameplay may also have contributed to more anomalous utterances, which further increased the number of recognition errors. Anomalous utterances (Figure 5) accounted for 15% of the speech corpus and 10% of all recognition errors. For example, because we had changed the input method to be spring-loaded to optimize efficiency, some recordings were partially cut-off due to the player releasing the record button prematurely. At other times, recordings were silent because the user hesitated to speak or accidentally pressed the record button. On occasion, game sounds such as row-completion ringing tones were also captured in the recording, even though they were designed not to overlap temporally with recorded speech. Furthermore, some users uttered nonsense phrases or English labels for the pictures, perhaps in a playful attempt to test the recognizer or in order to trigger the display of a hint, which is designed to appear once the user has attempted any utterance in a trial. More rarely, users conflated two vocabulary words and spoke a hybrid of two words.

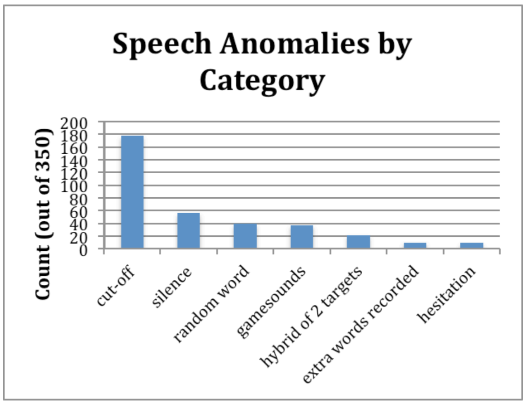

Figure 5: *Count of anomalous utterances by category.*

Overall, the most common anomalous cases were cut-off words and silent recordings (51% and 16% of anomalies, respectively). Cut-off recordings could be addressed by having the system constantly listen for speech and pad recorded utterances with extra time on both ends before sending them to the recognizer. Silent recordings could be better handled by incorporating silence into the recognizer's language model such that silence is a competing hypothesis in addition to the existing vocabulary words. In cases where the recognizer hypothesizes silence, the game interface can give feedback to the user to try again or speak louder. We leave these improvements for future work and instead focus on improving overall performance regardless of anomalies.

5. Strategies to Improve Performance

The disheartening effect of false negative recognition errors on user enjoyment suggests that relaxing the constraints of speech recognition to be more lenient could benefit engagement. The difficulties inherent in optimizing a letter-to-sound model for out-of-vocabulary words might also be alleviated by training lexicons on user-produced pronunciations mid-game that are detected to be likely correct. To this end, game-based constraints could be leveraged to provide strong contextual clues for maintaining high recognition accuracy in the face of greater leniency. To explore the viability of this approach, we identify several potential techniques for modifying the speech recognizer and re-evaluate the collected speech corpus on alternative recognizer configurations.

5.1. Dynamic vs. Static Vocabulary

Effective educational approaches tend to focus the learner's attention on only a few words or concepts at a time until their meanings have been internalized by the learner through repeated practice. In an intense and time-sensitive game setting, the gradual introduction of small sets of words is also critical for reducing the learner's cognitive load imposed by existing simultaneous interactions. Unlike typical speech interactions in which the set of possible user utterances may be large and uncertain, speech interactions amidst a learning game have implicit constraints that can be leveraged for enhancing speech recognition. Specifically, the game environment enables us to both constrain the recognizer vocabulary size and dynamically add additional words to the vocabulary as they are introduced to the learner. Constraining the vocabulary size can hopefully decrease the likelihood of false negative errors by preventing the recognizer from hypothesizing a word that the learner is unlikely to produce.

To determine the potential impact of this approach, we compare recognition accuracy between a static vocabulary of 28 words and a dynamic vocabulary (Figure 6), at varying numbers of pronunciations included in the lexicon. In the dynamic condition, we add a new word to the vocabulary only once it has appeared in the game, and constrain the maximum vocabulary size to only the words that the learner has seen within any particular game session (seven words maximum). The dynamic vocabulary demonstrated a 27% increase in accuracy over the static vocabulary when 10-best L2S pronunciations were included, and this benefit appeared fairly consistent across different numbers of N-best pronunciations included. The benefits were largely due to the substantial reduction in false negative errors, which were the source of most recognition errors.

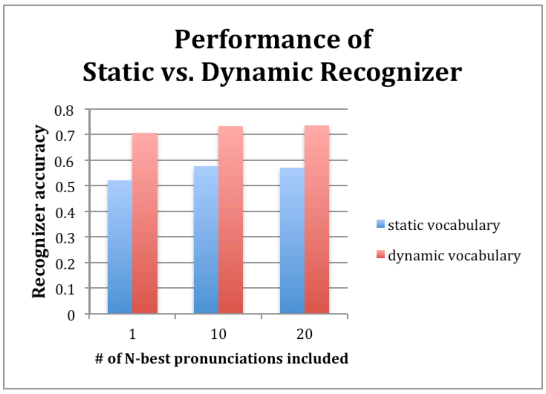

Figure 6: *Performance of static vs. dynamic recognizer at 1, 10, and 20 included pronunciations. The advantage of the dynamic recognizer remains fairly consistent across different numbers of N-best pronunciations.*

5.2. Deepening N-best Hypotheses

Game-based settings also provide strong contextual information about the target item on a trial-by-trial basis. Because the game keeps state of which target item is being presented to the user at every turn, a more lenient system could deem the learner correct if the target word appears in *any* of the top-N recognition hypotheses. This approach assumes that the recognizer has some room for error and that, because the learner is likely to have spoken the target word, it is safer to check the top few hypotheses for the correct response before deeming the utterance incorrect. Figure 7 illustrates a substantial increase in overall word accuracy simply by expanding the N-best depth from one (59%) to four (73%), all with a static vocabulary of 28 words. In practice, even though recognition accuracy can be further boosted with more hypotheses accepted, it would be preferable to set a limit on this number so that the user does not assume that the recognizer will accept any response.

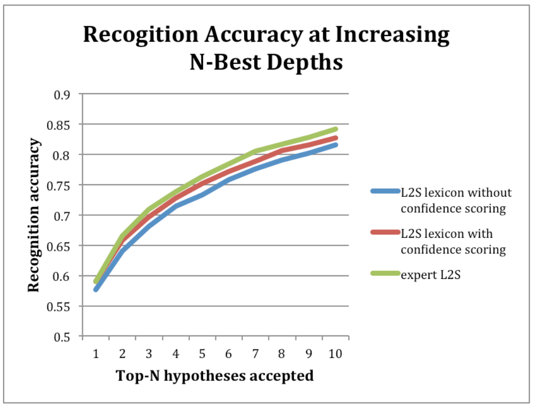

Figure 7: *Recognition performance, varying the number of N-best hypotheses accepted. Utterance is deemed correct when any top-N hypothesis matches the target word. Uses 10-best L2S pronunciations.*

A primary concern surrounding N-best depth expansion is the increased risk of false positive recognition errors. In the case of false positive errors, learners may mistakenly believe they have correctly recalled the word for a particular picture, with the consequence of strengthening an incorrect mapping. Hence, a trade-off may exist between decreasing frustration due to false negatives and increasing incorrectly learned mappings due to excessive leniency.

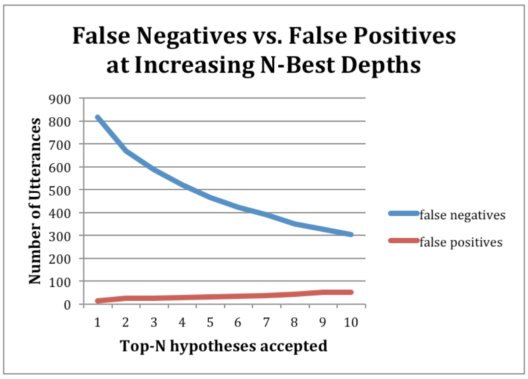

Figure 8: *Comparing the number of false negative and false positive utterances at an increasing number of N-best hypotheses accepted.*

To examine this potential trade-off, we measure the number of misrecognized utterances due to false negative and false positive errors at increasing N-best depths. Figure 8 shows that, as the number of accepted hypotheses increases, the number of false negative errors decreases dramatically, with only a minor increase in false positives. The significant decrease in false negatives is magnified by the elimination of repeated false negative errors due to learners re-attempting the same utterance after experiencing a false negative. Nevertheless, we find a very similar trend even after removing such repetitions from the dataset.

We further analyze false negatives and false positives among anomalous utterances, and find that anomalous recordings account for a substantial 80% of all false positive errors, compared to only 24% of all false negative errors. Because the majority of false positives are anomalies, and because a sizeable number of those are due to users producing random utterances, learners may find false positives more transparent and potentially less impenetrable than false negatives. In general, false positives are also less frustrating because they do not unfairly hinder the player's in-game progress. After a false positive, the player immediately focuses his or her attention on block rotation rather than being forced to re-attempt the utterance, making those experiences potentially more forgettable. These patterns lend support to the notion of adapting in-game speech recognition systems to be more lenient.

5.3. Training on high confidence user utterances

Lastly, out-of-vocabulary terminology can be detrimental to recognition accuracy and game enjoyment. Unlike acoustic and language models that learn the values of their parameters from training data, word pronunciations in a recognizer's lexicon are typically specified manually, often by an expert. Hence, a user wishing to review out-of-vocabulary words might encounter frequent recognition errors due to a letter-to-sound model that has been trained using only existing lexicons.

Recent work on pronunciation mixture models (PMM) has made it possible for experts to specify a set of pronunciations, but leave the weighting of these pronunciations to the PMM using speech data collected on the fly [11]. Yet, in a game-based learning context, it is unclear how unlabeled utterances can be used for training a PMM live, due to a chicken or egg problem of learners being unreliable agents for speaking the correct target item.

Nonetheless, we make a key insight that players are first introduced to the word-picture pair before the word is withheld for memorization practice. Because the learner sees both the word and cue on the first trial by way of introduction, the first utterance the player produces for any word has a high likelihood of being correct. In the Tetris game we have designed, the learner also hears the word pronounced out loud when it is first introduced, making it more likely that the learner will speak the target word correctly, particularly in the case of second language learning. On the other hand, first utterances may also be riskier for training since they could contain more anomalies such as hesitation and silence due to the user's unfamiliarity with the new item.

We thus evaluate speech recognition using pronunciations obtained by training a pronunciation mixture model solely on the user's first utterance of each word as a replacement lexicon (Figure 9). As a benchmark, we compare these results against lexicons produced using the letter-to-sound model. Because the test set for the PMM condition does not include any of a user's first utterances, we similarly remove all first utterances when evaluating recognition on the normal letter-to-sound lexicons.

Remarkably, the PMM trained purely on the users' first utterances demonstrated a 3% improvement over the L2S lexicon (averaged over results from one to twenty pronunciations included), despite having no ground-truth labeling of any first-trial utterances. A PMM trained on other learners' first utterances produced no significant advantage over the L2S lexicons, suggesting that speaker-dependent characteristics may be critical to effective recognition.

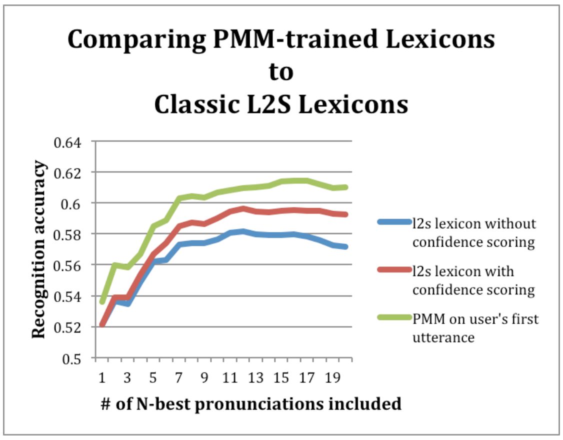

Figure 9: *Comparing performance of a PMM trained on the first utterance of each word to that of normal L2S lexicons.*

The promising speech recognition enhancement obtained by training only a small number of high confidence user utterances suggests further exploration of opportunities to perform user-specific PMM training using high confidence in-game scenarios. For example, starting a game in study mode before transitioning to retrieval mode could not only give the learner more time to develop familiarity with new items, but also offer an advantage for speech recognition enhancement. One could imagine collecting utterances during the study phase to produce a true mixture of multiple utterances produced by the same user for each word.

6. Conclusion

Our work has shown that a speech recognizer designed for traditional purposes may be unnecessarily strict when placed in a fast-paced game context, particularly because false negative recognition errors are both self-perpetuating and detrimental to learner enjoyment. We have proposed several techniques for improving performance, such as using a small and dynamic recognizer vocabulary, expanding the set of N-best accepted hypotheses, and using high confidence in-game utterances to retrain out-of-vocabulary words. Although a more lenient recognizer may run the risk of accepting learner errors, we found these occurrences to be surprisingly rare, and well worth the trade-off of decreasing the significant frustration associated with false negatives. It would be worthwhile to evaluate whether first utterances remain advantageous for PMM training in a second language learning context, despite learner inexperience in the target language.

While speech recognition has experienced limited adoption in fast-paced educational games compared to alternatives such as adventure style games, our results suggest that tailoring the recognizer to the unique needs of time-sensitive game environments could be key to increasing adoption. Future work should explore methods for handling speech anomalies specific to learning amidst rapid gameplay, such as using voice activity detection or time padding to prevent cut-off speech, and a silence model to handle accidental or hesitant recordings. Finally, automatic detection of words that are likely to be poorly ranked by the recognizer's letter-to-sound model, perhaps by comparing PMM scores to default L2S rankings of out-of-vocabulary items, would be a worthwhile venture for future research.

7. Acknowledgements

This research was funded by MIT Lincoln Laboratory. Special thanks to Ian McGraw for his valuable input and mentorship.

8. References

[1] Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. Speech Communication 50(5), 434-451.

[2] Bisson, C. and Luckner, J. (1996). Fun in learning: The pedagogical role of fun in adventure education. Journal of Experiential Education 19(2), 108-12.

[3] Cai, C. (2013) Adapting arcade games for learning. Proc. CHI 2013, 2665-2670.

[4] Gee, J. (2003). What video games have to teach us about learning and literacy. Computers in Entertainment 1(1), 20-20.

[5] Glass, J. (2003) A probabilistic framework for segment-based speech recognition. Computer Speech and Language 17(2), 137-152.

[6] Gruenstein, A., McGraw, I., and Badr, I. (2008). The WAMI Toolkit for Developing, Deploying, and Evaluating Web-Accessible Multimodal Interfaces. Proc. ICMI, 141-148.

[7] Hazen, T. J., Seneff, S., and Polifroni, J. (2002). Recognition confidence scoring and its use in speech understanding systems. Computer Speech and Language, 16(1), 49-67.

[8] Kumar, A., Reddy P., Tewari, A., Agrawal, R., and Kam A. (2012). Improving literacy in developing countries using speech recognition-supported games on mobile devices. Proc. CHI 2012, 1149-1158.

[9] Malone, T. (1980). What makes things fun to learn? A study of intrinsically motivating computer games. Pipeline 6(2), 50-51.

[10] McGraw, I., Yoshimoto, B. and Seneff, S. (2009). Speech-enabled card games for incidental vocabulary acquisition in a foreign language. Speech Communication 51(10), 1006-1023.

[11] McGraw, I., Badr, I., and Glass, J. (2013). Learning Lexicons from Speech Using a Pronunciation Mixture Model. IEEE Transactions on Audio, Speech & Language Processing 21(2): 357-366.

[12] Nakamura, J. And Csikszentmihalyi, M. (2009). Flow theory and research. In C. R. Snyder & S. J. Lopez (Eds.), Handbook of positive psychology, 195-206.

[13] Seneff, S., Adler, M., Glass, J. Sherry, B., Hazen, T., Wang, C., & Wu, T. (2007). Exploiting Context Information in Spoken Dialogue Interaction with Mobile Devices. Proc. Intl Workshop on Improved Mobile User Experience, Toronto, Canada.

[14] Van der Spek, E.D. Wouters, P., & Van Oostendorp, H. (2009). Code Red: Triage. Or, Cognition-based Design Rules Enhancing Decisionmaking Training in a Game Environment. In Games and Virtual Worlds for Serious Applications, 2009. VS-GAMES'09. Conference in (pp. 166-169). IEEE.

[15] Xu, Y. and Seneff, S. (2012). Improving Nonnative Speech Understanding Using Context and N-Best Meaning Fusion. Proc. ICASSP, 4977-4980. IEEE.

[16] Xue, W. and Rudnicky, I. (2000). Language Modeling for Dialog System. Proc. ICSLP.