

Enhancing accessibility through correction of speech recognition errors

John-Mark Bell

Learning Societies Lab, School of Electronics and Computer Science, University of Southampton, Southampton, UK.

jmb@ecs.soton.ac.uk

Abstract

The learning within lectures of hearing-impaired students can be hindered by errors in captions generated by speech recognition. My research intends to address this problem by investigating ways of correcting these captions. I summarise approaches to automatic error correction and describe the preliminary studies that have been conducted. These studies show that human editors set a tough benchmark for automatic correction to meet and indicate that automatic correction is feasible. Finally, I summarise my intention to develop a correction framework that will permit quantitative and qualitative testing of correction methods.

Introduction & motivation

University lectures are not an accessible environment for students with hearing difficulties. In lectures, information is mainly imparted through the words that the lecturer speaks. Although visual media (e.g. Powerpoint slides) are often used, these are usually a complement to what the lecturer says and not a complete representation of all that is taught in the lecture. Currently, a number of approaches are used to tackle this problem. These include employing a sign-language interpreter, a stenographer, or third-party note-taker to capture the information content of a lecture. Unfortunately, sign-language interpreters and stenographers are a limited resource. Thus it can be difficult to find suitably qualified interpreters and stenographers are often employed in more financially rewarding environments (such as court reporting). Third-party note-takers are often students who are taking the same course. The quality of the notes is variable and there is no guarantee that they capture the full information content of the lecture.

Existing research has investigated the use of real-time, automatic speech recognition (ASR) based, captioning of lectures [1]. It shows that ASR captioning can, in theory, make lectures accessible for hearing-impaired students by providing access to the words the lecturer speaks. The captions can either be displayed upon a screen at the front of the lecture theatre for all students to view or on a personal display client running on a student's laptop or PDA. Bain et al [1] also present anecdotal evidence that ASR transcription is of use to those without hearing impairment (e.g. those who have difficulty taking in-class notes). ASR-generated captions also pave the way to automatically generated lecture transcripts that can be placed online after the lecture for revision purposes.

However, the results of the research show that "nearly 40 percent of faculty participants reached the benchmark of at least 85 percent accuracy" (where "accuracy" is defined as the number of correct words in the ASR output divided by the total number of words spoken) and the mean accuracy rate across all participants was 77% with a standard deviation of

9.58. From this, it is clear that over 60% of lecturers did not reach the required accuracy rate in their lectures and that improvement is required.

Errors within lecture transcripts can be particularly problematic as they may distort the lecture content, either rendering it meaningless or, worse, imparting incorrect information. For example, it is not uncommon for ASR-generated captions to insert or remove “not” from a sentence, which clearly distorts the meaning of what was said.

The focus of my research is to improve the experience of hearing-impaired students within university lectures. As it is unlikely that significant improvements in the accuracy of ASR will be achieved in the short term, this will be approached through investigation into methods of correcting ASR output. The intention is to determine whether automatic post-processing of ASR output offers any significant improvement over the raw ASR output within the educational setting.

Related work

The impact of errors upon ASR usage is application dependent. For example, an ASR-based airline ticket booking system might be more susceptible to certain types of recognition error than a lecture transcription system. This results in a number of error handling strategies. These can be divided into two broad categories; strategies for detecting the location of errors and mechanisms for correcting errors. It is important to note that this categorisation is entirely conceptual – in practice, the separation may be implicit within an error handling technique.

The initial problem is to detect reliably the identity of errors. Sarma et al [6] propose using contextual analysis to identify likely errors. This exploits the relationship between words in a given context, usually over a larger range than afforded by the trigram models used by an ASR engine. By generating statistics for the likelihood of a word occurring given the words around it, it becomes possible to predict whether a given word is in error.

Another approach to error detection is to apply Naive Bayes classifiers to features of the ASR output. One possible feature set would be the confidence scores associated with the output words, the alternate hypotheses for a given word and the difference in scores between output words, as discussed by Zhou et al [9]. However, as these approaches are entirely statistical, there is no guarantee that they will successfully identify all errors in the output. Nor do they avoid false positives – i.e. not all words flagged as being in error are truly errors.

After detecting the potential presence of errors, it becomes necessary to correct the errors (or mitigate their effects). One way of solving the error correcting problem is to employ humans to edit ASR output. This was investigated by Wald et al [8]. Their study shows that, on average, a human editor may correct 23.9% of the errors in ASR output in real time. To correct an error, the user has to detect, select, and then edit it.

Alternatively, a technological approach may be used. For example, previous research into error correction within dialogue systems (such as the airline booking system mentioned previously) has shown that statistical post-processing of ASR output can have a beneficial impact upon accuracy [5]. This is achieved through use of a fertility channel model to “translate” between the ASR output and what was actually said. Dialogue systems can also enter a corrective sub-dialogue where the human participant performs the correction through prompting by the dialogue system (e.g. [2]). In situations where keywords are of paramount importance, utterance verification may be used to detect and selectively correct errors [7].

Other correction approaches include using probabilistic techniques or knowledge bases to reorder candidate lists [4].

Solution & methodology

Existing ASR engines generally have some form of facility that permits editing and correction of the output by the end user. This is designed for use during dictation or in off-line editing. Therefore, it is not well suited to the requirements of use in a lecture. Usually, when corrections are made using this facility, the ASR engine's statistical model is updated to reflect the correction. This allows the ASR engine to learn words and phrases that are commonly used by the speaker. Additionally, the impact of a single correction upon the ASR engine's statistical model is likely to be small, due to the engine's model being the result of a large corpus of training data.

The approaches identified could lead to an automatic correction process operating on the output of an ASR engine (i.e. as a post-processing step). This would be more suited to a lecture scenario and could interface with the manual editing system described by Wald et al [8].

Domain-specific processing

Generalised ASR engines are trained using a large corpus of data, which is non-domain-specific in nature. Due to this non-specificity, it is likely that the engine will make mistakes when encountering domain-specific content, even if all words are in the engine's vocabulary. The intention is to produce word-based statistical models that would be able to compensate for the ASR engine's genericity by post-processing the output. These models would contain information about the kind of language found in lectures and specific knowledge about the subject matter. For example, techniques from the field of machine translation may be used to produce a model that "translates" between ASR output and what was actually said.

Linguistic analysis

ASR engines' linguistic knowledge is based upon statistical information generated from a large corpus of training data. An approach using linguistic analysis would use more explicit knowledge of language constructs to detect grammatical errors. This is likely to be particularly useful on a higher level than the statistical word sequence model used by ASR engines. As ASR engines' models are very narrow (limited to two or three words), higher-level concepts such as grammatical correctness are not tested for. The use of linguistic analysis is intended to introduce this higher-level consideration of the output.

Candidate list analysis

As a side effect of the statistical approach used by ASR engines, a list of alternative hypotheses is commonly generated. This list contains words or phrases that are considered to be the most likely alternatives for the output word. Therefore it is the ideal place to look for potential correct output. For example, if a system has access to content-related information (such as Powerpoint slides or lecture notes), it could use this to hypothesise the correct word in the candidate list.

Uses of phonemic information

Phoneme information may be used to mitigate the effect of segmentation errors within the ASR engine's acoustic modelling stage. Anecdotal evidence suggests that, in a reasonably

large number of cases, mis-segmentation is the likely cause of a misrecognition. Therefore, if the phonemic information is available on output, a phoneme-based re-segmentation technique might be used to attempt to correct this situation. Phonemic information may also be used to counteract out-of-vocabulary (OOV) errors through the use of a phoneme-sequence to word dictionary.

Current work

So far, some preliminary investigations have been undertaken into the impact of multiple, independent, human editors upon the output and into the feasibility of utilising candidate list information to automatically correct ASR output. A further study has investigated the feasibility of using a machine-translation style model to correct the ASR output automatically.

Effectiveness of multiple human editors

Raw data collected by Wald et al [8] were re-evaluated to gain some insight into the effect of multiple human editors working independently, thus giving a benchmark against which automatic correction techniques may be compared. In the original study, five users attempted correction of errors in sixteen separate test cases, using a prototype tool. Assuming the existence of a system capable of collating the output of their editing, I wished to determine what impact the addition of editors would have.

To do this, each combination of editors was examined and their output merged, to find the combination that produced least errors for each of 1, 2, 3, 4 and 5 editors. On average, a single editor was capable of correcting 24% of the errors in the transcript. In the best case, using more than two editors gains little extra benefit (two editors corrected 44% of the errors, on average). It is likely that the majority of editors are correcting the same (or very similar) errors, thus the remaining errors are less likely to be corrected by adding further editors.

Candidate list evaluation

An experiment was carried out into the improvement possible when considering the candidate list generated by an ASR engine. It was assumed possible to detect erroneous output reliably. A metric was used to obtain a theoretical upper bound on the achievable level of improvement to the output transcript's error rate. The metric treated as correct all erroneous words for which the correct transcription could be found in the candidate list, no matter where in the list the correct transcription occurred. The results are promising and indicate that, for a mean initial word error rate of 22%, an absolute reduction in error rate of 7% may be achieved with a standard deviation of 1.67. This corresponds to 32% of the errors in the ASR output being corrected.

Machine translation model

A preliminary investigation was conducted into the feasibility of using a machine translation model to correct errors in ASR output automatically. This was based upon the larger-scale experiments (from the natural language translation field) described by Casacuberta [3]. These previous experiments were designed to determine the better of two model training techniques. Essentially, it involved creating "extended symbols", which mapped from the input word to the output word sequence, then using these to create a finite-state transducer.

My experimentation was upon a reasonably small corpus, which was divided into four blocks. In the experimentation, three types of model were created – bigram, trigram and four-gram,

and two types of extended symbol were used (type-i and type-ii). This resulted in 6 models being built and tested, with the aim being to discover:

1. The usefulness of machine translation models in this situation
2. The percentage of errors in the ASR output that could be corrected
3. The most successful model

A cross-validation experiment was then carried out in which three blocks were used for training each model and the remaining block was used for testing. This was repeated four times (once for each permutation of the blocks). All words were in vocabulary to eliminate the potential for out-of-vocabulary errors.

The results of this investigation are promising. They show that the machine translation approach could result in a significant reduction in the number of errors in the ASR output. The type-ii extended symbol was clearly better than the type-i extended symbol, regardless of whether a bigram, trigram or four-gram model was built. No difference in improvement was observed between trigram and four-gram models, which is likely to be the result of the small size of the corpus. Both trigram and four-gram models performed better than the bigram models (for their respective symbol type). Overall, the best model (type-ii, tri/four-gram) was capable of correcting 27.7% of the errors in the raw ASR output.

Future Work

Future work will focus upon finding reliable methods for identifying, isolating and correcting erroneous sections of ASR output. This will consider the use of linguistic features (such as parts of speech) and confidence scores for error detection. It will also consider the use of domain-specific knowledge and statistical modelling to perform error correction. This will be followed by the development of a framework into which implementations of error detection and correction techniques may be plugged. The framework will permit quantitative testing of individual and combined techniques; thus allowing comparison of them. Further to this, qualitative testing involving users of the system will be conducted with the aim being to demonstrate its usefulness and highlight areas where extra improvement would be beneficial.

References

1. K. Bain, S. Basson, A. Faisman, and D. Kanevsky. Accessibility, transcription and access everywhere. *IBM Systems Journal*, 44(3):589–603, July 2005.
2. C. Bousquet-Vernhettes, R. Privat, and N. Vigouroux. Error handling in spoken dialogue systems: toward corrective dialogue. In *Proceedings of Workshop on Error Handling in Spoken Dialogue Systems*, 2003.
3. F. Casacuberta. Inference of finite-state transducers by using regular grammars and morphisms. In *Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 2000.
4. H. Lieberman, A. Faaborg, W. Daher, and J. Espinosa. How to wreck a nice beach you sing calm incense. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 278–280, New York, NY, 2005.
5. E. K. Ringger and J. F. Allen. Error correction via a post-processor for continuous speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 427–430, Atlanta, GA, 1996.
6. A. Sarma and D. D. Palmer. Context-based speech recognition error detection and correction. In *Proceedings of the HLT-NAACL Conference: Short Papers*, pages 85–88, Boston, MA, 2004.

7. A. Setlur, R. Sukkar, and J. Jacob. Correcting recognition errors via discriminative utterance verification. In *Proceedings., Fourth International Conference on Spoken Language*, volume 2, pages 602–605, October 1996.
8. M. Wald, J.-M. Bell, P. Boulain, K. Doody, and J. Gerrard. Correcting automatic speech recognition errors in real time. *International Journal of Speech Technology*, In Press.
9. L. Zhou, J. Feng, A. Sears, and Y. Shi. Applying the naive bayes classifier to assist users in detecting speech recognition errors. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 183b–183b, 2005.

About the author



John-Mark Bell is a PhD student within the Learning Societies Lab in the School of Electronics and Computer Science at the University of Southampton. His studentship, which began in October 2006, is funded by an EPSRC grant and is supervised by Dr Mike Wald and Dr Andrew Gravel. His undergraduate study was also at the University of Southampton, being awarded an MEng in Computer Science in 2006. John-Mark's research interests include accessibility and universal design, with a specific focus upon hearing impairment. His PhD is focussed upon improving the accessibility of university lectures for hearing-impaired students, especially through improvement of captioning