INTERSPEECH 2017: Show & Tell Contribution
August 20–24, 2017, Stockholm, Sweden

A system for real time collaborative transcription correction

Peter Bell, Joachim Fainberg, Catherine Lai, Mark Sinclair

The Centre for Speech Technology Research, University of Edinburgh, EH8 9AB, UK
{peter.bell,j.fainberg,c.lai,mark.sinclair}@ed.ac.uk

Abstract

We present a system to enable efficient, collaborative human correction of ASR transcripts, designed to operate in real-time situations, for example, when post-editing live captions generated for news broadcasts. In the system, confusion networks derived from ASR lattices are used to highlight low-confident words and present alternatives to the user for quick correction. The system uses a client-server architecture, whereby information about each manual edit is posted to the server. Such information can be used to dynamically update the one-best ASR output for all utterances currently in the editing pipeline. We propose to make updates in three different ways; by finding a new one-best path through an existing ASR lattice consistent with the correction received; by identifying further instances of out-of-vocabulary terms entered by the user; and by adapting the language model on the fly. Updates are received asynchronously by the client.

Index Terms: speech recognition, speech transcription, language modelling

1. Introduction

News moves fast [1]. Media consumers are faced with an ongoing barrage of information involving new people, places, and events. Media providers face the challenge of tracking emerging stories and trends from around the world. The difficulties of managing constant updates are exacerbated by the fact that news items are increasingly distributed through audio and video media. Thus, transcription is required to access new content. While speech recognition technology has improved dramatically in recent times, automatic systems still make significant errors which require hand correction. In applications such as broadcast news captioning, such manual editing is highly time-critical.

An important factor in Automatic Speech Recognition (ASR) performance is how well the system matches its domain. While ASR technology can be customised to many applications, adaptation generally occurs too slowly to keep up with changes in events. Importantly, entities suddenly featuring in the news for the first time (e.g. 'White House chief of staff *Reince Priebus*') and portmanteaus (e.g. 'Brexit') are likely to be out-of-vocabulary (OOV) when first encountered. Moreover, OOV words are likely to mistranscribed in multiple ways depending on the context (e.g., 'Brecht's it', 'breaks it'). Additionally, utterances that are completely in-vocabulary may be incorrectly transcribed if the correct transcription is unlikely given the current language model.

To address these issues, we present a transcription system that can quickly learn from its mistakes given minimal manual intervention. The system identifies context types a manual correction occurred in, e.g. topics, allowing automatic application of the correction to similar contexts even when the transcription errors are of different form. To propagate corrections the system integrates recent news text to match incoming audio/video items with topic appropriate language models and generates pronunciations for OOV terms.

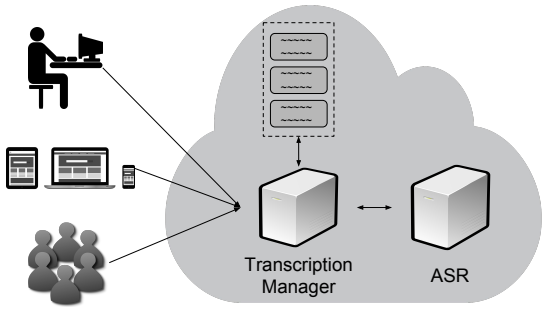

Figure 1: *All transcription and ASR services are in the cloud. Corrections can come from sources including professional transcribers, end-user communities and crowd-sourcing. Simple multi-platform applications allow fast and easy corrections.*

Figure 2: *Example user interface. The user can quickly and easily select alternative hypotheses, write a correction or delete.*

2. System architecture

Our overall system architecture is shown in Figure 1. End-users interact with automatic transcriptions via a web interface (Figure 1). Manual corrections are then integrated with our existing transcription tools running on a shared server in the cloud. Various ASR outputs are also saved, such as confusion networks, which are used to apply corrections through large collections. Through the interface the user can edit the transcripts by either selecting a suggestion from a confusion network, deleting the word, or typing a new alternative, as shown in Figure 2. Edits are pushed to the server, and the transcriptions are updated *asynchronously*. The system learns from the changes made and is dynamically updated for all users. This process is shown in Figure 3.

3. Automatic transcript correction

Our ASR and transcript correction systems are implemented in Kaldi [2], with base systems trained on multi-genre broadcast media data from the MGB Challenge [3]. Rapid correction and adaptation is achieved through a combination of on the fly lattice rescoring, OOV term identification, and topic-based language model adaptation.

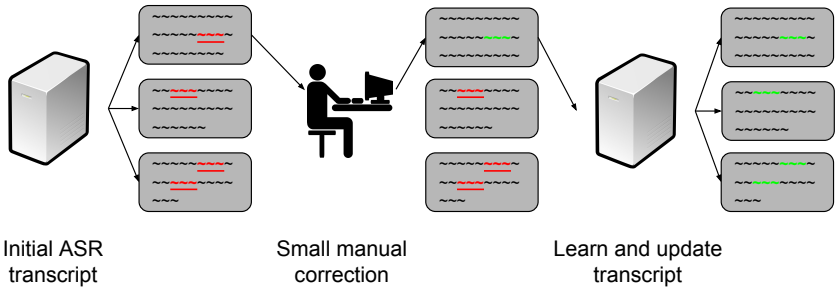

Figure 3: *An ASR system produces a transcription hypothesis that contains some errors. A human transcriber submits corrections for a small portion of the transcript. The system then learns from those corrections and updates its hypotheses for the whole transcript.*

3.1. Lattice rescoring on the fly

The system implements a simple method of updating the ASR transcript at the level of individual utterances. When an edit is received, it is used to update the ASR confusion network for the current utterance, expressed in the form of a finite-state transducer (FST), G'. Given a lattice for the current utterance also in FST form, U, it is very quick to compute a new one-best transcription consistent with the edits received by computing

$$W = ShortestPath(U \circ G') \quad (1)$$

3.2. Correction of out-of-vocabulary errors

A frequent problem in ASR for broadcast media is the occurrence of OOV terms. Although sometimes an OOV word may be identified in advance of automatic transcription and dynamically added to the vocabulary, it is common the the existence of an OOV term is discovered only during post-editing. Given that OOVs are frequently important entities in a given news story, it is useful to be able to quickly perform identification of further instances of the same OOV in the ASR output, once it has been received by the server. To achieve this, we recast the problem as a keyword search task: given an OOV term identified by a user edit, we search for other instances of the term in recently-processed media following the proxy-keywords method of [4], which again operates on the pre-existing lattice collection. Our implicit assumption is that specific OOV terms are highly likely to be clustered in time.

3.3. Topic-based language model adaptation

The system also performs topic-dependent language model adaption based on recent news. Topic distributions are learned over the last week's news using latent Dirichlet allocation (LDA) [5] using the MALLET toolkit [6]. Media items are assigned to clusters based on inferred topic distributions, and n-gram language models are trained for each topic cluster. The vocabulary for each model is made up of the union across all the topic-based language models as well as a large background n-gram model built from previously seen data. We interpolate each topic-based language model with the background model with a strong bias towards the current topic of interest. The system is also capable of adapting neural network based language models [7]. This is done by performing additional rounds of backpropagation using set of manual corrections received by the server, in a manner similar to [8].

Once the system identifies the topical context where a new manual correction occurs it can propagate that correction in other media items by updating topic specific language models. This allows corrections to be implicitly shared between users over large media collections.

4. Conclusion

We have proposed a transcription tool that adapts to recent news topics, is able to incorporate new OOV words, and that will dynamically and rapidly adapt to changes made by end-users. The server-client architecture allows multiple users to interact with the system at once, and to collaborate on the same transcriptions through asynchronous updates. Crucially, by rescoring the lattices given user edits, the new information is propagated downstream such that one edit may fix other, related errors.

5. Acknowledgements

This work was developed as part of a BBC News Labs *newsHACK*[1] event. We are grateful to the BBC for their support.

6. References

[1] K. Saltzis, "Breaking news online: How news stories are updated and maintained around-the-clock," *Journalism Practice*, vol. 6, no. 5-6, pp. 702–710, 2012.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[3] P. Bell, M. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester, and P. Woodland, "The MGB challenge: Evaluating multi-genre broadcast media recognition," in *Proc. ASRU*, 2015.

[4] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. ASRU*, 2013.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[7] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, vol. 2, 2010, p. 3.

[8] S. R. Gangireddy, P. Swietojanski, P. Bell, and S. Renals, "Unsupervised adaptation of recurrent neural network language models," in *Proc. Interspeech*, San Francisco, USA, Sep. 2016.

[1] `http://bbcnewslabs.co.uk/projects/news-hack`