IMPROVING MULTIPLE-CROWD-SOURCED TRANSCRIPTIONS USING A SPEECH RECOGNISER

R. C. van Dalen, K. M. Knill, P. Tsiakoulis, M. J. F. Gales

University of Cambridge, Department of Engineering
Trumpington Street, Cambridge CB2 1PZ, United Kingdom

ABSTRACT

This paper introduces a method to produce high-quality transcriptions of speech data from only two crowd-sourced transcriptions. These transcriptions, produced cheaply by people on the Internet, for example through Amazon Mechanical Turk, are often of low quality. Often, multiple crowd-sourced transcriptions are combined to form one transcription of higher quality. However, the state of the art is to use essentially a form of majority voting, which requires at least three transcriptions for each utterance. This paper shows how to refine this approach to work with only two transcriptions. It then introduces a method that uses a speech recogniser (bootstrapped on a simple combination scheme) to combine transcriptions. When only two crowd-sourced transcriptions are available, on a noisy data set this improves the word error rate to gold-standard transcriptions by 21 % relative.

Index Terms— Automatic speech recognition, crowd-sourcing, transcription combination

1. INTRODUCTION

Speech processing often relies on human-annotated training data. In recent years, it has become possible to use crowd-sourcing, which allows non-experts to perform small tasks. For various transcription tasks, crowd-sourcing is cheaper than paying experts (for the transcriptions used in this work, by a factor of 10), and has produced results not much worse than experts do [1, 2].

For speech recognition, the most important annotation is transcriptions of audio data, to be used, for example, to train speech recognisers. Various pieces of work have looked into using crowd-sourced transcriptions of audio. It is possible to apply smart quality control [3]. It is also possible to use transcriptions in a way that is robust to errors, for example, for speech recognition adaptation [4].

This paper aims to produce high-quality transcriptions for speech recogniser training by combining as few crowd-sourced transcriptions as possible. The conceptual model is illustrated as a Venn diagram in Fig. 1. The ellipse in the middle, in purple, stands for the gold-standard transcriptions, which are unavailable. Anything outside of this ellipse is incorrect. Three different transcriptions, by two crowd-sourcers and one ASR system, overlap in great part with the gold-standard transcriptions, but also have errors. However, all three have different errors, and the speech recogniser in particular has few errors in common with the crowd-sourcers. The intersection between a few different transcriptions is of higher quality, i.e. closer to the gold-standard transcription, than any single transcription. Of particular concern, however, is that using a speech recogniser, which may make more errors than any crowd-sourcer, to find a combined transcription, could deteriorate the final transcription instead of improving it. This paper will take particular care to prevent that.

This paper reports on research supported by Cambridge English, University of Cambridge.

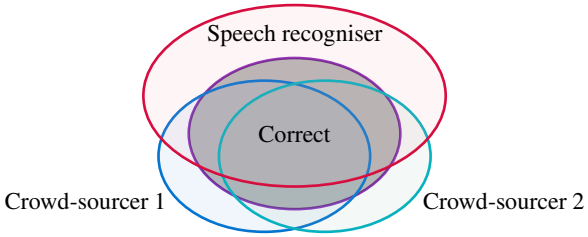


Fig. 1. Venn diagram of transcriptions: most errors are different between ASR and crowd-sourcers, and between crowd-sourcers.

The state of the art in combination of crowd-sourced transcriptions [5] is to acquire multiple transcriptions for each utterance and to combine them. This uses Rover [6], a combination method originally meant for combining speech recogniser output. A problem is that at least three transcriptions are necessary for breaking ties. Without a tie-breaking mechanism, the quality of a combination of two transcriptions will not be better than of a single transcription [7]. This paper will introduce a refinement to the Rover combination scheme so it prefers words over non-words, which makes it possible to get gains even from just two transcriptions.

None of these schemes use a speech recogniser as a knowledge source. A paper that does is [8], which incrementally acquires crowd-sourced transcriptions for each utterances and stops as soon as it is confident enough. However, this scheme requires two transcriptions to match exactly, and thus is only practical for short utterances.

Alternatively, it is possible to run a speech recogniser with a *biased language model* [9, 10]. This is a language model that has been trained on the errorful transcriptions, so that where the transcriptions are correct, they are likely be recognised correctly. This is especially useful where stretches of text are correct, and at times the audio deviates from the text for a number of words [11, 12, 13]. Here, however, errors in the transcriptions may happen more frequently, so this method may be inappropriate. It also introduces the risk that the speech recogniser adds errors instead of correcting them, since any errors from the red ellipse in Fig. 1 can be produced.

Therefore, this paper will introduce a novel method that combines transcriptions into a network as [6, 5, 7] do, and then uses speech recognition as a knowledge source to find one higher-quality transcription. It uses the word network resulting from the combination of transcriptions, and constrains the speech recogniser to choose the best path in such a network.

2. CHOOSING ONE TRANSCRIPTION

A simple general strategy of improving transcription quality is to acquire multiple transcriptions for each utterance and choose one. This strategy assumes that the variation in quality within one transcription is much less than between different transcriptions of one utterance. This may be the case when utterances are short, as in [8], or when each of the crowd-sourcers delivers work of consistent quality.

One strategy is to pick the longest of multiple transcriptions in terms of the number of characters or word tokens. The length of a transcription can be seen as a proxy for the amount of effort put into the transcription. Spontaneous speech from language learners often contains partially produced words and hesitations, and longer transcriptions should be more likely to represent those accurately.

Another strategy is to use agreement between multiple crowd-sourcers [8]. If a majority of transcribers produce the exact same transcription, then this is likely to be the correct one. This scheme lends itself to requesting transcriptions one by one, which in practice can focus effort, and money spent, on utterances that need it. However, for longer utterances with more noise, the sentence error rate would be too high. For example, for the professional transcriptions used in this work (see section 4.1), the inter-transcriber sentence error rate is around 80 %.

The last strategy is to use a trained recogniser, possibly trained using the longest transcription for each utterance. The recogniser assigns a likelihood to each complete transcription, and the most likely transcription is selected. This is equivalent to running forced alignment with each transcription and selecting the transcription that yields the highest likelihood. Because the recogniser only selects from complete crowd-sourced transcriptions, it is prevented from biasing the output. In terms of Fig. 1, errors are restricted to transcriptions in the union of the ellipses for crowd-sourcers.

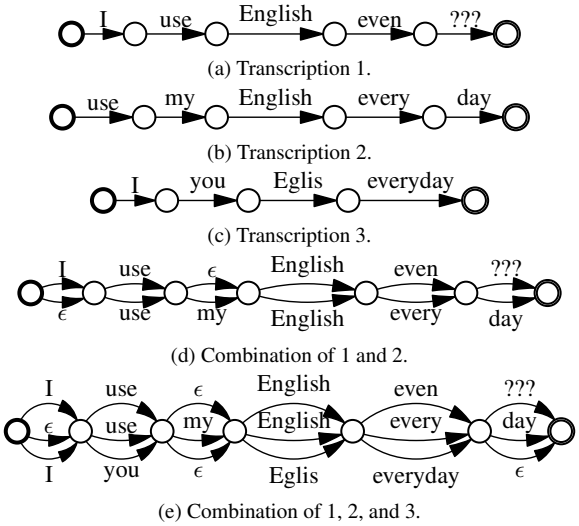
Selecting one transcription for a whole utterance, as each of these approaches do, does not allow mixing and matching of words from different transcriptions. The longer the utterances are, the more of a limitation this becomes. The next section will therefore discuss fine-grained combinations of words.

3. COMBINING TRANSCRIPTIONS

Multiple transcriptions by crowd-sourcers who are well-meaning but in a hurry can contain different mistakes. It is therefore useful to be able to mix and match parts of transcriptions.

One way of combining transcriptions is to align two or more, and represent them as a word network [6]. This process is illustrated in Fig. 2. The resulting network contains paths for each transcription, and paths made up from combining transcriptions. Where the transcriptions agree, the final network contains transitions with the same word; where they disagree, it contains arcs for each source transcription, with a word or possibly ϵ, the empty symbol, as a label. It is produced by aligning two transcriptions with a minimum edit distance algorithm (see e.g. [14] for a textbook description). This is a dynamic programming algorithm that finds an alignment that minimises the number of insertions, deletions, and substitutions to turn the one transcription into the other. To align two transcriptions of length m, the algorithm searches through a space of $O(m^2)$ states, which takes $O(m^2)$ time. In general, to align n transcriptions at once, the state space, and thus the time taken, would be $O(m^n)$.

Instead, an approximation can be used [6]. The longest transcriptions are aligned first, and then the other ones are added one by one. This uses a generalisation of the standard minimum edit distance algorithm that aligns a single transcription with a network.



(a) Transcription 1.
(b) Transcription 2.
(c) Transcription 3.
(d) Combination of 1 and 2.
(e) Combination of 1, 2, and 3.

Fig. 2. Combining multiple transcriptions into a word network.

The algorithm requires a cost function for adding arcs to a graph like Fig. 2. The cost of adding a transition with a symbol is 0 if there is already a transition with that symbol, and 1 otherwise. This corresponds to a "substitution" (when the symbol is a word) or a "deletion" (when the symbol is ϵ). The cost of an "insertion", here inserting a new state and therefore an arc, and ϵ-arcs for existing transcriptions, is also 1. Having defined the local costs, the same dynamic programming algorithm for the edit distance that is normally used on two sequences of words can then be used on a network and a sequence of words. Each step in the resulting lowest-cost path has a transition in the original network and a new transition. This indicates where to add transitions to the original network.

The following discusses the approaches for using this word network to acquire one combined transcription.

3.1. Majority voting

The current state of the art [5] is to perform Rover combination without weights [6] on a number of crowd-sourced transcriptions. This produces a word network as described above, and for each segment chooses the most frequently occurring word, or no word if ϵ occurs most frequently. This scheme requires at least three transcriptions [7], otherwise ties can be broken only randomly, which does not improve transcription quality.

However, even between two transcriptions it is often possible to choose by a simple heuristic. Assuming that crowd-sourcers are more likely to leave out words than to insert them, a tie between ϵ and a word can be broken in favour of the word. Additionally, if one transcription contains an indication that the transcriber could not hear the word (here indicated by "???"), and another does have a word, again the tie can be broken in favour of the word. This heuristic is not available in the original Rover tool but is straightforward to implement. This will be called "Rover+tie".

3.2. Rover combination with confidence scores

Standard Rover combination was proposed for combining speech recognition outputs, which can have confidence scores, often derived from posteriors. An obvious extension to the scheme above (to the speech recognition researcher) is therefore to produce a speech

recognition lattice and thence a confusion network, align it with the word network generated from the transcriptions, and copy confidence scores from segments with matching symbols to the transcription network. A problem with this scheme is the following.

To compute posteriors of words, in theory the likelihoods of all possible word sequences must be computed. Since this is impossible, speech recognisers normally prune the search paths and approximate the hypothesis spaces with a lattice. This is sufficient for computing confidence scores on hypotheses from the same speech recogniser: almost by definition these hypotheses are contained in the lattice. However, not all words from transcriptions will be in the lattice.

Another problem is the level of confidence assigned to the absence of a word. In standard use of Rover, the inputs are speech recognition hypotheses. The speech recognisers used for this will have been calibrated to give a consistent balance of insertions and deletions on a development set. For transcriptions by crowd-sourcers whose identities change between utterances, this type of consistency is unattainable.

In initial experiments, these two problems made this form of standard Rover perform less well than the alternatives.

3.3. Constrained speech recognition

The methods discussed in the previous section use confidence scores, which are essentially posteriors. They are not always equal to posteriors of the model (say, the HMM), because they often represent better estimates of the actual posteriors, with all word sequences before and after marginalised out. However, a sequence of high-posterior words does not in general form a consistent word sequence. For example, even when used on speech recogniser output Rover combination does not necessarily account for all the audio: it may combine word hypotheses that overlap in time, or have gaps in between them.

When combining speech recognition outputs, an advantage to posteriors over likelihoods is that they give a different view (because competing hypotheses are taken into account). In the case of interest here, however, which is to find the best crowd-sourced transcription, it is not at all clear that posteriors are preferable to likelihoods.

This paper therefore proposes the following straightforward method. It is to use a speech recogniser, constraining the hypotheses using the word network, like in Fig. 2e, as a word grammar. This forces the speech recogniser to find a consistent hypothesis, accounting for all of the audio. It uses the word likelihoods instead of the posteriors. To prevent any bias towards the language model, none is used.

3.4. Biased language model

A method to use errorful transcriptions (such as closed captions, or speeches that were given more free-style than they were on paper) is to use a recogniser with a biased language model [9, 10]. Such a language model is trained on the errorful transcriptions, and then interpolated with a generic background language model with a low interpolation weight. While recognising, N-grams that have been seen in the transcriptions are more likely to be recognised than other sequences of words than if the audio was recognised using a fair language model.

This method of using errorful transcriptions is particularly useful when they contain fragments of the actual speech larger than a few words. It has been used for lectures [12], where lecture notes were available, and audio books [11]. The requirement is sometimes set that a sequence in the recognition output of, say, at least three words matches the original transcription [10, 13]. This may not be appropriate for the case here, where the errors will be more spread throughout the transcription.

4. EXPERIMENTS

The audio data used to compare the approaches for obtaining high-quality transcriptions from crowd-sourced ones was made available by Cambridge English through the ALTA Institute[1]. It consists of recorded proficiency tests for English. The recording quality varies greatly. The language skills of the speakers similarly range from proficient (level C) to very poor (level A1).

Three subsets are defined. One, BLXXXman, is for evaluation. It contains 88 speakers in 10 hours of audio. All are Indian Gujarati native speakers. Each utterance has been transcribed by four crowd-sourcers. Additionally, each utterance has been transcribed by at least one professional transcription service. Some speakers were transcribed by more than one transcriber, for the purpose of evaluation.

The other two subsets, BLXXXtrn00 and BLXXXtrn01, are used for speech recogniser training. They contain 279 and 306 speakers in 31.5 and 34 hours, respectively. Again, all speakers are Indian Gujarati native speakers. Each of the utterances has been transcribed twice by crowd-sourcers. Each of the utterances is a reply to a single prompt. Some of the prompts elicit answers with multiple sentences. The average number of words per utterance (measured on the gold-standard transcriptions) is 28.

The initial speech recogniser used for some of the methods is trained on a set which contains 253 speakers in common with BLXXXtrn00, but leaves out 26 speakers (initially used for evaluation) and includes 55 non-Indian-Gujarati speakers. A GMM-HMM acoustic model is trained with maximum-likelihood estimation on PLP features. Then bottleneck features from a neural network, trained on the AMI meeting data corpus, are appended to the PLP features and a new system is built, applying minimum phone error training and speaker-adaptive training. The language model is a trigram model interpolated between one trained on broadcast news and one trained on the acoustic model training data.

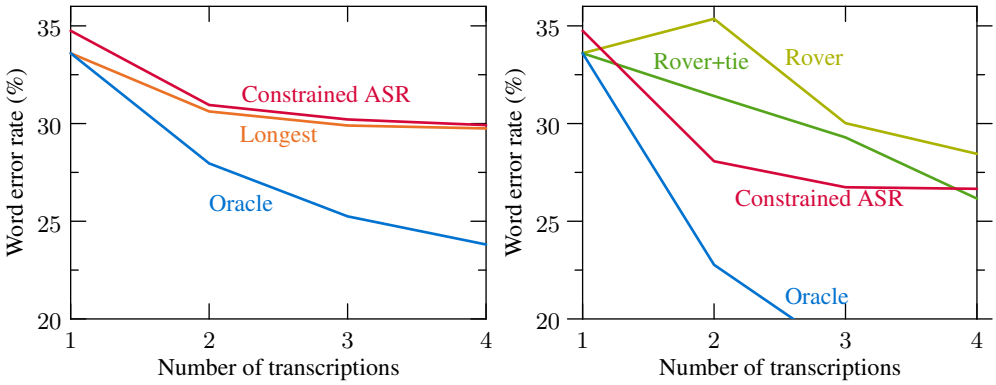
4.1. Transcription quality

To normalise some of the variation in transcriptions produced by crowd-sourcers, a preprocessing step is performed. A variety of hesitation markers are mapped to a special symbol. A list of markers, including some 100 spelling variants of "unintelligible" and "inaudible", are mapped to a special token for unknown words, and markers for non-English text are similarly mapped. The transcribers often try to be helpful and describe background noises in great detail; these descriptions are also removed. Non-alphabetical symbols are removed, and numerals written out. This preprocessing is performed the same way for crowd-sourced and professional transcriptions.

The error rates between each pair of the three professional transcription services on the utterances that were double-transcribed are 21.8, 26.8, and 25.4, with a weighted average of 23.5. This indicates how noisy and errorful the data is. All word error rates quoted in the rest of this paper will use as the reference transcription the longest transcription (in number of words) for each utterance. This should be a relatively conservative way of choosing the most reliable transcription.

The speech recogniser with an unbiased language model obtains a word error rate of 40.4 %.

[1] http://www.wiki.cl.cam.ac.uk/rowiki/NaturalLanguage/ALTA



(a) Picking one whole transcription.

(b) Combining transcriptions.

Fig. 3. Performance for combination schemes: choosing one transcription versus combining transcriptions.

4.2. Results

Fig. 3 shows word error rates against the professional transcriptions for various approaches. The left figure, 3a, shows performance of picking one whole transcription for each utterance as the number of crowd-sourced transcriptions (and therefore cost) increases. The right figure shows the performance of schemes that pick a path through a word network as in Fig. 2e. All lines start at the same point, 33.6 %, when there is only one transcription to choose from (the numbers for constrained ASR are slightly different because a few transcriptions failed to align at the pruning threshold).

The first thing to note is the cheating experiments, the oracle performance. In the left graph, this gives the performance if the best transcription is chosen for each utterance. The word error rate goes down to 23.8 % for four transcriptions to choose from. In the right graph, the best possible path through a word network is chosen. The best performance is off the graph, at 15.6 %, and much better than choosing one whole transcription. But even for two transcriptions the potential performance is 28.0 % for choosing one whole transcription versus 22.7 % for a word network. This makes clear that combination schemes have the potential to yield far more accurate transcriptions than picking one transcription.

The next thing to note is the two experiments in the left graph. The system indicated by "Longest" does not use a speech recogniser; it merely selects the longest transcription in terms of number of tokens. The "Constrained ASR" approach runs a speech recogniser, forcing it to choose between the transcriptions. The lines are very close together, showing that the heuristic of selecting the longest transcription, a proxy for effort and accuracy on part of the transcriber, is a good indicator of transcription quality.

The right graph, Fig 3b, shows the performance of two variants of Rover combination in shades of green. The difference between "Rover" and "Rover+tie" is that "Rover+tie" breaks ties in favour of words over the absence of a word or an unknown word (something the standard Rover tool does not support). Standard Rover breaks ties randomly; going from one to two transcriptions, therefore has essentially no impact on performance. However, again the heuristic of preferring longer sequences over shorter ones pays off; on two transcriptions "Rover+tie" performs better (by 4.0 % absolute) than standard Rover, and this advantage remains as the number of transcriptions goes up to 4. However, constraining the ASR not to whole transcriptions but to all combinations of them does yield a performance improvement.

With two transcriptions, most of the advantage of using the speech recogniser as a source of information has been used up. At four transcriptions, the strength of the speech recogniser as an arbiter gets beaten by human knowledge. As the number of transcriptions goes to infinity, the proportion of arcs with a specific word will tend to the real posterior probability that a human would pick that word. With four samples, the sampling error has a similarly-sized effect on performance to the modelling error of the speech recogniser.

If there are a large number of resources available, it is therefore worth combining a large number of crowd-sourced transcriptions using Rover, breaking ties in favour of words. However, if resources are limited, paying for two crowd-sourced transcriptions, combining them into a word network, and finding the most likely path, gives the largest improvement in performance. Compared to the state of the art (Rover without tie-breaking), at 35.4 %, this gives a word error rate of 28.1 %, a relative improvement of 21 %.

A final experiment is run with a biased language model. A trigram language model is trained on the transcriptions for all data sets acquired with the best combination scheme (at 28.1 % for BLXXXman). This is then interpolated with a weight of 0.9 with a language model trained on broadcast news with a weight of 0.1. Then, the speech recogniser is run with that language model. On BLXXXman, this yields a word error rate of 33.2 %. Unlike in earlier use cases for biased language models [9, 10, 11, 12, 13], here the errors are spread more evenly across the transcriptions. Where the transcription contains any errors within the N-gram window, the biased language model cannot help.

5. CONCLUSION

This paper has introduced a novel method to combine multiple crowd-sourced transcriptions of speech into a higher-quality transcription. Unlike previous work, the final transcription is constrained to be in a word network resulting from combining various transcriptions, but uses an automatic speech recogniser. The advantage of constraining the transcription is that the speech recogniser cannot unduly influence the final transcription. The advantage of using a speech recogniser is that only two crowd-sourced transcriptions are required. Compared with a combination scheme that is the state of the art, the method in this paper reduces the word error rate in the final transcription by 21 % relative.

6. REFERENCES

[1] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng, "Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks," in *Proceedings of EMNLP*, 2008.

[2] Gabriel Parent and Maxine Eskenazi, "Speaking to the crowd: Looking at past achievements in using crowdsourcing for speech and predicting future challenges," in *Proceedings of Interspeech*, 2011.

[3] Scott Novotney and Chris Callison-Burch, "Cheap, fast and good enough: Automatic speech recognition with non-expert transcription," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

[4] Kartik Audhkhasi, Panayiotis G. Georgiou, and Shrikanth S. Narayanan, "Analyzing quality of crowd-sourced speech transcriptions of noisy audio for acoustic model adaptation," in *Proceedings of ICASSP*, 2012.

[5] Keelan Evanini, Derrick Higgins, and Klaus Zechner, "Using Amazon Mechanical Turk for transcription of non-native speech," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

[6] Jonathan G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proceedings of ASRU*, 1997.

[7] Matthew Marge, Satanjeev Banerjee, and Alexander I. Rudnicky, "Using the Amazon Mechanical Turk for transcription of spoken language," in *Proceedings of ICASSP*, 2010.

[8] Jason D. Williams, I. Dan Melamed, Tirso Alonso, Barbara Hollister, and Jay Wilpon, "Crowd-sourcing for difficult transcription of speech," in *Proceedings of ASRU*, 2011.

[9] H. Y. Chan and P. C. Woodland, "Improving broadcast news transcription by lightly supervised discriminative training," in *Proceedings of ICASSP*, 2004.

[10] Long Nguyen and Bing Xiang, "Light supervision in acoustic model training," in *Proceedings of ICASSP*, 2004.

[11] Norbert Braunschweiler, M.J.F. Gales, and Sabine Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *Proceedings of Interspeech*, 2010.

[12] Y. Long, M. J. F. Gales, P. Lanchantin, X. Liu, M. S. Seigel, and P. C. Woodland, "Improving lightly supervised training for broadcast transcription," in *Proceedings of Interspeech*, 2013.

[13] Hank Liao, Erik McDermott, and Andrew Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription," in *Proceedings of ASRU*, 2013.

[14] Daniel Jurafsky and James H. Martin, *Speech and Language Processing*, Prentice-Hall, New Jersey, 2000.