Using Amazon Mechanical Turk for Transcription of Non-Native Speech

Keelan Evanini, Derrick Higgins, and Klaus Zechner
Educational Testing Service
{KEvanini, DHiggins, KZechner}@ets.org

Abstract

This study investigates the use of Amazon Mechanical Turk for the transcription of non-native speech. Multiple transcriptions were obtained from several distinct MTurk workers and were combined to produce merged transcriptions that had higher levels of agreement with a gold standard transcription than the individual transcriptions. Three different methods for merging transcriptions were compared across two types of responses (spontaneous and read-aloud). The results show that the merged MTurk transcriptions are as accurate as an individual expert transcriber for the read-aloud responses, and are only slightly less accurate for the spontaneous responses.

1 Introduction

Orthographic transcription of large amounts of speech is necessary for improving speech recognition results. Transcription, however, is a time consuming and costly procedure. Typical transcription speeds for spontaneous, conversational speech are around 7 to 10 times real-time (Glenn and Strassel, 2008). The transcription of non-native speech is an even more difficult task–one study reports an average transcription time of 12 times real-time for spontaneous non-native speech (Zechner, 2009).

In addition to being more costly and time consuming, transcription of non-native speech results in a higher level of disagreement among transcribers in comparison to native speech. This is especially true when the speaker's proficiency is low and the speech contains large numbers of grammatical errors, incorrect collocations, and disfluencies. For example, one study involving highly predictable speech shows a decline in transcriber agreement (measured using Word Error Rate, WER) from 3.6% for native speech to 6.4% for non-native speech (Marge et al., to appear). Another study involving spontaneous non-native speech showed a range of WER between 15% and 20% (Zechner, 2009).

This study uses the Amazon Mechanical Turk (MTurk) resource to obtain multiple transcriptions for non-native speech. We then investigate several methods for combining these multiple sources of information from individual MTurk workers (turkers) in an attempt to obtain a final merged transcription that is more accurate than the individual transcriptions. This methodology results in transcriptions that approach the level of expert transcribers on this difficult task. Furthermore, a substantial savings in cost can be achieved.

2 Previous Work

Due to its ability to provide multiple sources of information for a given task in a cost-effective way, several recent studies have combined multiple MTurk outputs for NLP annotation tasks. For example, one study involving annotation of emotions in text used average scores from up to 10 turkers to show the minimum number of MTurk annotations required to achieve performance comparable to experts (Snow et al., 2008). Another study used preference voting to combine up to 5 MTurk rankings of machine translation quality and showed that the resulting judgments approached expert inter-annotator agreement (Callison-Burch, 2009). These tasks, however, are much simpler than transcription.

MTurk has been used extensively as a transcription provider, as is apparent from the success of a middleman site that act as an interface to MTurk for transcription tasks.[1] However, to our knowledge, only one previous study has systematically evaluated the quality of MTurk transcriptions (Marge et al., to appear). This recent study also combined multiple MTurk transcriptions using the ROVER method (Fiscus, 1997) to produce merged transcriptions that approached the accuracy of expert transcribers. Our study is similar to that study, except that the speech data used in our study is much more difficult to transcribe–the utterances used in that study were relatively predictable (providing route instructions for robots), and contained speech from native speakers and high-proficiency non-native speakers. Furthermore, we investigate two additional merging algorithms in an attempt to improve over the performance of ROVER.

3 Experimental Design

3.1 Audio

The audio files used in this experiment consist of responses to an assessment of English proficiency for non-native speakers. Two different types of responses are examined: spontaneous and read-aloud. In the spontaneous task, the speakers were asked to respond with their opinion about a topic described in the prompt. The speech in these responses is thus highly unpredictable. In the read-aloud task, on the other hand, the speakers were asked to read a paragraph out loud. For these responses, the speech is highly predictable; any deviations from the target script are due to reading errors or disfluencies.

For this experiment, one set of 10 spontaneous (SP) responses (30 seconds in duration) and two sets of 10 read-aloud (RA) responses (60 seconds in duration) were used. Table 1 displays the characteristics of the responses in the three batches.

3.2 Transcription Procedure

The tasks were submitted to the MTurk interface in batches of 10, and a turker was required to complete the entire batch in order to receive payment. Turkers received $3 for a complete batch of transcriptions ($0.30 per transcription).

Batch	Duration	# of Words (Mean)	# of Words (Std. Dev.)
SP	30 sec.	33	14
RA1	60 sec.	97	4
RA2	60 sec.	93	10

Table 1: Characteristics of the responses used in the study

Different interfaces were used for transcribing the two types of responses. For the spontaneous responses, the task was a standard transcription task: the turkers were instructed to enter the words that they heard in the audio file into a text box. For the read-aloud responses, on the other hand, they were provided with the target text of the prompt, one word per line. They were instructed to make annotations next to words in cases where the speaker deviated from the target text (indicating substitutions, deletions, and insertions). For both types of transcription task, the turkers were required to successfully complete a short training task before proceeding onto the batch of 10 responses.

4 Methods for Merging Transcriptions

4.1 ROVER

The ROVER method was originally developed for combining the results from multiple ASR systems to produce a more accurate hypothesis (Fiscus, 1997). This method iteratively aligns pairs of transcriptions to produce a word transition network. A voting procedure is then used to produce the merged transcription by selecting the most frequent word (including NULL) in each correspondence set; ties are broken by a random choice.

4.2 Longest Common Subsequence

In this method, the Longest Common Subsequence (LCS) among the set of transcriptions is found by first finding the LCS between two transcriptions, comparing this output with the next transcription to find their LCS, and iterating over all transcriptions in this manner. Then, each transcription is compared to the LCS, and any portions of the transcription that are missing between words of the LCS are tallied. Finally, words are interpolated into the LCS by se-

[1] http://castingwords.com/

lecting the most frequent missing sequence from the set of transcriptions (including the empty sequence); as with the ROVER method, ties are broken by a random choice among the most frequent candidates.

4.3 Lattice

In this method, a word lattice is formed from the individual transcriptions by iteratively adding transcriptions into the lattice to optimize the match between the transcription and the lattice. New nodes are only added to the graph when necessary. Then, to produce the merged transcription, the optimal path through the lattice is determined. Three different configurations for computing the optimal path through the lattice method were compared. In the first configuration, "Lattice (TW)," the weight of a path through the lattice is determined simply by adding up the total of the weights of each edge in the path. Note that this method tends to favor longer paths over shorter ones, assuming equal edge weights. In the next configuration, "Lattice (AEW)," a cost for each node based on the average edge weight is subtracted as each edge of the lattice is traversed, in order to ameliorate the preference for longer paths. Finally, in the third configuration, "Lattice (TWPN)," the weight of a path through the lattice is defined as the total path weight in the "Lattice (TW)" method, normalized by the number of nodes in the path (again, to offset the preference for longer paths).

4.4 WER calculation

All three of the methods for merging transcriptions are sensitive to the order in which the individual transcriptions are considered. Thus, in order to accurately evaluate the methods, for each number of transcriptions used to create the merged transcription, $N \in \{3, 4, 5\}$, all possible permutations of all possible combinations were considered. This resulted in a total of $\frac{5!}{(5-N)!}$ merged transcriptions to be evaluated. For each N, the overall WER was computed from this set of merged transcriptions.

5 Results

Tables 2 - 4 present the WER results for different merging algorithms for the two batches of read-aloud responses and the batch of spontaneous responses. In each table, the merging methods are ordered according to their performance when all transcriptions were used (N=5). In addition, the overall WER results for the individual turkers and an expert transcriber are provided for each set of responses. In each case, the WER is computed by comparison with a gold standard transcription that was created by having an expert transcriber edit the transcription of a different expert transcriber.

Method	N=3	N=4	N=5
Individual Turkers	7.0%		
Lattice (TWPN)	6.4%	6.4%	6.4%
Lattice (TW)	6.4%	6.4%	6.4%
LCS	6.0%	5.6%	5.6%
Lattice (AEW)	6.1%	6.0%	5.5%
ROVER	5.5%	5.2%	5.1%
Expert	4.7%		

Table 2: WER results 10 read-aloud responses (RA1)

Method	N=3	N=4	N=5
Individual Turkers	9.7%		
Lattice (TW)	9.5%	9.5%	9.4%
Lattice (TWPN)	8.3%	8.0%	8.0%
Lattice (AEW)	8.2%	7.4%	7.8%
ROVER	7.9%	7.9%	7.6%
LCS	8.3%	8.0%	7.5%
Expert	8.1%		

Table 3: WER results for 10 read-aloud responses (RA2)

In all cases, the merged transcriptions have a lower WER than the overall WER for the individual turkers. Furthermore, for all methods, the merged output using all 5 transcriptions has a lower (or equal) WER to the output using 3 transcriptions. For the first batch of read-aloud responses, the ROVER method performed best, and reduced the WER in the set of individual transcriptions by 27.1% (relative) to 5.1%. For the second batch of read-aloud responses, the LCS method performed best, and reduced the WER by 22.6% to 7.5%. Finally, for the batch of spontaneous responses, the Lattice (TW) method performed best, and reduced the WER by 25.6% to 22.1%.

Method	N=3	N=4	N=5
Individual Turkers	29.7%		
Lattice (TWPN)	29.1%	28.9%	28.3%
LCS	29.2%	28.4%	27.0%
Lattice (AEW)	28.1%	25.8%	25.1%
ROVER	25.4%	24.5%	24.9%
Lattice (TW)	25.5%	23.5%	22.1%
Expert	18.3%		

Table 4: WER results for 10 spontaneous responses

6 Conclusions

As is clear from the levels of disagreement between the expert transcriber and the gold standard transcription for all three tasks, these responses are much more difficult to transcribe accurately than native spontaneous speech. For native speech, expert transcribers can usually reach agreement levels over 95% (Deshmukh et al., 1996). For these responses, however, the WER for the expert transcriber was worse than this even for the read-aloud speech. These low levels of agreement can be attributed to the fact that the speech is drawn from a wide range of English proficiency levels among test-takers. Most of the responses contain disfluencies, grammatical errors, and mispronunciations, leading to increased transcriber uncertainty.

The results of merging multiple MTurk transcriptions of this non-native speech showed an improvement over the performance of the individual transcribers for all methods considered. For the read-aloud speech, the agreement level of the merged transcriptions approached that of the expert transcription when only three MTurk transcriptions were used. For the spontaneous responses, the performance of the best methods still lagged behind the expert transcription, even when five MTurk transcriptions were used. Due to the consistent increase in performance, and the low cost of adding additional transcribers (in this study the cost was $0.30 per audio minute for read-aloud speech and $0.60 per audio minute for spontaneous speech), the approach of combining multiple transcriptions should always be considered when MTurk is used for transcription. It is also possible that lower payments per task could be provided without a decrease in transcription quality, as demonstrated by Marge et al. (to appear). Additional experiments will address the practicality of producing more accurate merged transcriptions for an ASR system—simply collecting larger amounts of non-expert transcriptions may be a better investment than producing higher quality data (Novotney and Callison-Burch, 2010).

It is interesting that the Lattice (TW) method of merging transcriptions clearly outperformed all other methods for the spontaneous responses, but was less beneficial than the LCS and ROVER methods for read-aloud speech. It is likely that this is caused by the preference of the Lattice (TW) method for longer paths through the word lattice, since individual transcribers of spontaneous speech may mark different words as unitelligible, even though these words exist in the gold standard transcription. Further studies with a larger number of responses will be needed to test this hypothesis.

References

Chris Callison-Burch. 2009. Fast, cheap and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proc. EMNLP*.

Neeraj Deshmukh, Richard Jennings Duncan, Aravind Ganapathiraju, and Joseph Picone. 1996. Benchmarking human performance for continuous speech recognition. In *Proc. ICSLP*.

Jonathan G. Fiscus. 1997. A post-processing system to yield word error rates: Recognizer Ooutput Voting Error Reduction (ROVER). In *Proc. ASRU*.

Meghan Lammie Glenn and Stephanie Strassel. 2008. Shared linguistic resources for the meeting domain. In *Lecture Notes in Computer Science*, volume 4625, pages 401–413. Springer.

Matthew Marge, Satanjeev Banerjee, and Alexander I. Rudnicky. to appear. Using the Amazon Mechanical Turk for transcription of spoken language. In *Proc. ICASSP*.

Scott Novotney and Chris Callison-Burch. 2010. Cheap, fast, and good enough: Automatic speech recognition with non-expert transcription. In *Proc. NAACL*.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast – But is it good? Evaluating non-expert annotations for natural language tasks. In *Proc. EMNLP*.

Klaus Zechner. 2009. What did they actually say? Agreement and disagreement among transcribers of non-native spontaneous speech responses in an English proficiency test. In *Proc. ISCA-SLaTE*.