

Dynamic Partial Transcription Recycling System (DPTRS)

Authors: Pakhomov, Serguei, Schoenwetter, Michael, Bachenko, Joan.

Affiliation: Lernhout & Houspie, Inc.

Address: 5221 Edina Industrial Blvd., Edina, MN 55439

E-mail: pakh0002@tc.umn.edu, mjs@linguistech.com, joan-b@linguistech.com

Abstract

Literal transcriptions of speech are widely used in automatic speech recognition and speech recognition research for training and testing language and acoustic models, evaluating performance of speech recognition systems, constructing dictionaries. Generating literal transcriptions is a painstaking and error prone task that usually requires large amounts of time and effort spent by trained and therefore expensive transcriptionists. Partial transcriptions are a commercial product of most transcription operations and are readily available in most cases. The difference between a partial transcription and a literal transcription is that the former does not include various disfluencies such as filled pauses, repairs and repetitions found in abundance in spontaneous speech and is edited for grammar and usage. In a lot of cases, partial transcriptions represent the actual speech rather poorly. We would like to present an ASR based system that enables the user to convert the partial transcriptions into semi-literal transcriptions. The system has been tested at a commercial medical transcription operation and is currently used to build language models and adapted acoustic models.

The goodness of the system has been determined in part by aligning partial, semi-literal and literal transcriptions using DP that yields alignment accuracy measurements analogous to those used for measuring recognition accuracy. 774 sets of literal, semi-literal and partial transcriptions (a total of ~270,000 words) have been used to evaluate the current method. The alignment between literal and semi-literal transcripts proves to be significantly better (4.4% absolute) than the alignments between literal and partial transcripts.

The method we are proposing uses partial transcriptions, a filled pause model and a background model to build utterance specific language models. The partial transcript is a single file that is used to build an augmented probabilistic finite state model. We use the ECRL Toolkit for building the LM; however, any other toolkit can be used instead.

The APFSM is built by linear interpolation of a finite state model with two other probabilistic models. The background model accounts for expressions that may have been spoken by the speaker but did not make it into the partial transcription. Such expressions include but are not limited to greetings, thanking, false starts and repairs. A list of such out-of-transcription (OOT) expressions is derived by comparing already existing literal transcriptions with their partial transcription counterparts and taking the difference.

The filled pause model represents the same partial transcription populated with filled pauses ("um's and ah's") using a stochastic FP model derived from a relatively large corpus of literal transcriptions (Pakhomov, 1999, Pakhomov and Savova, 1999).

Interpolation weights are established empirically by calculating the resulting model's perplexity against held out data. OOV items are handled provisionally by generating on-the-fly pronunciations based on the existing dictionary spelling-pronunciation alignments.

The new language model is used to recognize speech that is represented in the partial transcript. Because the model is very small, the recognition time is close to real time. By increasing the language model factor, the recognizer is forced to produce a transcript that is likely to be more true to what has actually been said than the commercial partial transcript.

Further refinement of the new semi-literal transcript is carried out by using dynamic programming alignment on the recognizer's output used as hypothesis (HYP) and the partial transcription used as

reference (REF). The alignment results in each HYP word being labeled as either a MATCH, a DELETION, a SUBSTITUTION or an INSERTION. Those words¹ present in the HYP stream that do not align with anything in the REF stream are labeled as insertions and are assumed to represent the OOT elements of the dictation. Those words that do align but do not match are labeled as substitutions. Finally, the words found in the REF stream that do not align with anything in the HYP stream are marked as deletions. The final semi-literal transcript is constructed differently depending on the intended purpose of the transcript. If the transcript will be used for acoustic modeling or adaptation, only MATCHES, the REF portion of SUBSTITUTIONS and the HYP portion of only those INSERTIONS that represent punctuation and filled pauses² make it into the final semi-literal transcript. It is important to filter out everything else because acoustic modeling is very sensitive to misalignment errors. Language modeling, on the other hand, is less sensitive to that; therefore, one can be less conservative with INSERTIONS and DELETIONS.

Adapted acoustic models were built using this method for 4 talkers. The models trained on literal data performed on average 20% better than models trained on partial transcriptions. The models trained on semi-literal transcriptions performed on average 6.3% better (absolute value) than the models trained on partial transcriptions data.

REFERENCES

- Pakhomov, S. (1999). Modeling Filled Pauses in Medical Dictations. In Proc. ACL'99, pp. 619-624.
Pakhomov, S. and Savova, G. (1999). Filled Pause Distribution and Modeling in Quasi-Spontaneous Speech. In Proc. Disfluency Workshop at ICPhIS' 99.

ACKNOWLEDGEMENTS

We would like to extend special thanks to Michael Moon, Ph.D. for providing very inspiring and helpful discussions. We would also like to thank the researchers and staff at the company formerly known as Linguistic Technologies, Inc.

¹ "Words" here is used in a broad sense that includes filled pauses and silences.

² The empirical evidence suggests that the particular recognizer we used performs relatively well on recognizing punctuation and filled pauses.