IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

The Intelligent Voice ASR system for the Iberspeech 2018 Speech to Text Transcription Challenge

Nazim Dugan[1], *Cornelius Glackin*[1], *Gérard Chollet*[1], *Nigel Cannings*[1]

[1]Intelligent Voice Ltd, London, UK

nazim.dugan@intelligentvoice.com

Abstract

This paper describes the system developed by the Empathic team for the open set condition of the Iberspeech 2018 Speech to Text Transcription Challenge. A DNN-HMM hybrid acoustic model is developed, with MFCC's and iVectors as input features, using the Kaldi framework. The provided ground truth transcriptions for training and development are cleaned up using customized clean-up scripts and then realigned using a two-step alignment procedure which uses word lattice results coming from a previous ASR system. 261 hours of data is selected from *train* and *dev1* subsections of the provided data, by applying a selection criterion on the utterance level scoring results. The selected data is merged with the 91 hours of training data used to train the previous ASR system with a factor 3 times data augmentation by reverberation using a noise corpus on the total training data, resulting a total of 1057 hours of final training data. Selected text from the *train* and *dev1* subsections are also used for new pronunciation additions and language model (LM) adaptation of the LM from the previous ASR System. The resulting model is tested using data from the *dev2* subsection selected with the same procedure as the training data.

Index Terms: speech recognition, forced alignment, neural network

1. Introduction

Deep Neural Networks (DNNs) and Hidden Markov Model (HMM) based hybrid Automatic Speech Recognition (ASR) systems [1] are still widely used despite the recent rise of end-to-end speech recognition systems [2], based on recurrent neural networks (RNNs). An important part of the ASR acoustic model training procedure is the time alignment of the training audio with the transcriptions. Usually, the utterance start and end times of the training transcriptions are given to the ASR training system, and in the utterance, character or phone level alignments of the training text are computed by an iterative process, or with Connectionist Temporal Classification (CTC) [3]. If the given start and/or end time is wrong for an utterance it would be discarded by the ASR training system or if not, it will reduce the accuracy of the ASR training. Therefore, input data preparation with accurate utterance level time alignments is an important prerequisite for training an ASR system in either case.

Kaldi is a commonly used speech recognition framework which supports DNN-HMM hybrid systems with a couple of different DNN implementations. It also supports Gaussian Mixture Model (GMM) HMM hybrid systems [4] which are mostly used iteratively for the alignment of input transcripts with the input audio frames. State of the art Kaldi recipes use a phonetic description of the transcripts and the GMM-HMM iterative phone alignment methodology, even though there is experimental support for character-based training and CTC alignment in the Kaldi framework. These new experimental recipes mostly produce ASR systems with slightly lower accuracy compared to the state of the art recipes.

In this work, a DNN-HMM hybrid Kaldi recipe with GMM-HMM iterative phone alignment was used for training a European Spanish ASR system with 8kHz down-sampled input audio and transcriptions from the Iberspeech 2018 speech to text transcription challenge training and development data, and four well known European Spanish corpora. A previous ASR system for European Spanish language was used for the utterance level time alignments of the input transcriptions provided for the Iberspeech 2018 speech to text transcription challenge, training and development. These transcriptions are also used for the language model (LM) adaptation of the previous ASR model using the SRILM toolkit [6].

2. Data preparation

It was observed that the provided utterance level time alignments of the training and development transcripts were not accurate and some of the provided files do not have any time alignments. Word lattice results coming from a previous ASR system trained using four well known European Spanish corpora: Albayzin [7], Dihana [8], CORLEC-EHU [9], TC-STAR [10] and the text corpus El País were used for the re-alignment of the training and development transcripts. The ASR system used to generate the word lattice was developed with the same input data as the ASR model described in [5]. However, an improved recipe of the Kaldi framework [11] is used which utilizes a factor 3-times data augmentation with reverberation and noise addition, high resolution Mel Frequency Cepstral Coefficients (MFCCs), iVectors [12] and *nnet3 chain* implementation. This previous ASR model will be referred to as the *base model* throughout the remaining text.

The provided training and development audio files are sub-sampled to 8kHz before being used in the training and testing processes.

10.21437/IberSPEECH.2018-57

2.1. Two-step time alignment procedure

A two-step utterance level time alignment procedure is used which includes forced alignment of plain text transcripts and word level time alignments using the NIST *sclite* ASR scoring utility in Speech Recognition Scoring Toolkit (SCTK) [13].

2.1.1. Forced alignment of plain text transcripts

Time alignments in the *srt* type subtitle and *stm* type reference files of the *train*, *dev1* and *dev2* subsections of the provided training and development data are cleaned up to produce plain text transcriptions where utterances are separated by a newline character. An iterative forced alignment script [14] which accepts plain text transcripts and ASR word lattice results as input, is used to compute the utterance level time alignments. Punctuation cleanup is also applied on the plain text transcripts in order to improve the accuracy of the alignment procedure, since ASR word lattice results do not involve any punctuation. In each iteration, the alignment script uses confidence regions of the results of the previous iteration to narrow down the search space. The number of iterations is configured as three from previous experience.

2.1.2. Word level time alignment

The results from the forced alignment procedure described in the previous section have been used to generate *stm* type reference files to be used as the reference input to the NIST *sclite* ASR scoring utility. The ASR word lattice results are converted to *ctm* format which is accepted by the *sclite* utility as a hypothesis file with word alternative level time information. The NIST *sclite* utility was called with the "*-o sgml*" option in order to generate word level logs of correct words, substitutions, deletions and insertions as seen in Table 1. Information in these logs are used to find the word level timings of all the words in the reference files using linear time interpolation for the deletions. The computed word level timings are used to verify and update the utterance level start and end times computed with the forced alignment script described in previous section.

Table 1: *Formatted output example of NIST sclite ASR scoring utility with sgml type output. Possible* ***Type*** *values are correct* (C), *substition* (S), *deletion* (D) and *insertion* (I). ***Ref*** *column is for the words from stm reference file,* ***Hyp*** *column is for the words from ctm hyphotesis file.* ***Hyptime*** *is the word start time in hypothesis file and* ***Reftime*** *is the time for the reference word computed by the sclite utility.*

Type	Ref	Hyp	Reftime	Hyptime
C	españa	españa	10.640	10.700
I	-	un	11.240	11.310
S	entre	entra	11.560	11.640
D	y	-	12.120	12.140

Word level timings can be obtained with the *sclite* utility without using the utterance level start and end times coming from the forced alignment procedure described in the previous section. However, the computation time in this case is on the order of days for a typical file from the training or development set and the results are not as accurate as the results obtained by the two-step process discussed in this work.

2.2. Data selection for training and testing

Utterance level time alignment information computed with the two-step procedure described in the previous section is used to convert the plain text transcripts into *stm* reference format accepted by NIST *sclite* utility where each utterance was labeled as a different speaker in order to enable utterance level ratios of correct, substitute, deleted and inserted words. The ASR best path results are used as the hypothesis input in *ctm* format. With the observation that the ratio of insertions is high for wrongly aligned utterances, the two criteria below are applied in order to select the utterances with correct alignments:

1) %insertions < (%correct + %substitutions + %deletions)

2) %correct > 0

The provided ground truth transcriptions are assumed to be correct, and no analysis was carried out to test the correctness. 278 hours of training data is selected from *train*, *dev1* and *dev2* subsections of the provided data with the described data preparation and selection mechanism. 17 hours of data from *dev2* subsection is preserved for testing and 261 hours of data from *train* and *dev1* subsections are used for training the ASR model.

The selected training data is combined with 91 hours of other European Spanish ASR training data from databases: Albayzin, Dihana, CORLEC-EHU, TC-STAR [5] to obtain 352 hours of training data for ASR task.

3. ASR model training

The Kaldi framework [11] was used to train the acoustic model of the ASR system submitted to Iberspeech 2018 Text to Speech Transcription Challenge by the Empathic team. Component diagram for ASR system training and testing is given in Figure 1.

3.1. Kaldi Framework

Kaldi is an open source toolkit for automatic speech recognition, intended for use by speech recognition researchers and professionals. It is composed of C++ binaries, utilities written in Bash Script, Python and Perl scripting languages and ready to run training and testing recipes with data preparation steps for various languages and scenarios.

3.2. Lexicon update

The vocabulary and the rule based phonetic descriptions of the base model is expanded using the out of vocabulary words detected in the new data selected from the provided training and development data. The final vocabulary size used in the ASR model training is 110124 words. However, for the testing with a language model a subset of this vocabulary is used.

3.3. Acoustic model building

The Kaldi Aspire recipe [15] which is the recipe used by the John Hopkin's University team for the submission to the IARPA ASpIRE challenge was used for building the DNN-HMM acoustic model. High resolution MFCC's with 40 coefficients and iVectors are used as input features. Gaussian posteriors used for the iVector estimation are based on the input features with Cepstral Mean and Variance Normalization (CMVN). GMM-HMM iterative phone alignment was used before starting the neural network training of the DNN-HMM training stage where *nnet3 chain* implementation of the Kaldi framework was used with *frame-subsampling-factor* of 3, reducing number of output frames to 1/3 of the input frames. A 3-fold data augmentation is applied on the input acoustic features for the DNN-HMM training stage using the reverberation algorithm implemented in the Kaldi framework, by using noise databases RWCP, AIR and Reverb2014 in order to create multi-condition data of total 1057 hours.

A sub-sampled Time-delayed Deep Neural Network (TDNN) [16] with 6 layers and with ReLU and pnorm activation functions is used. Details of the neural network architecture and the stochastic gradient descent (SGD) based greedy layer-wise supervised training can be found in the system submission article for the Kaldi Aspire recipe [17].

TDNN training time for 2 epochs and 508 iterations is 13 hours 30 minutes with 3 NVidia GPUs (Quadro K6000, GeForce Titan X, GeForce Titan XP). Last iteration training and validation accuracies are 0.171631 and 0.199849 respectively. In order to avoid over-training, a selective system combination is carried out over all the iterations skipping the first 100 iterations, considering recorded accuracies for individual iteration results.

3.4. Language model (LM) adaptation

A 3-gram LM of the base model described in Section 2 is adapted using the selected training transcriptions of the provided data for the Iberspeech 2018 Speech to Text Transcription Challenge. Selected training transcriptions are parsed with the *ngram-count* command of the SRILM utility up to order of 3–grams and calculated probabilities are mixed with the previous probabilities of the base model using *ngram* command with mixing coefficient 0.1. Vocabulary size for the produced LM is about 67000.

3.5. Model testing

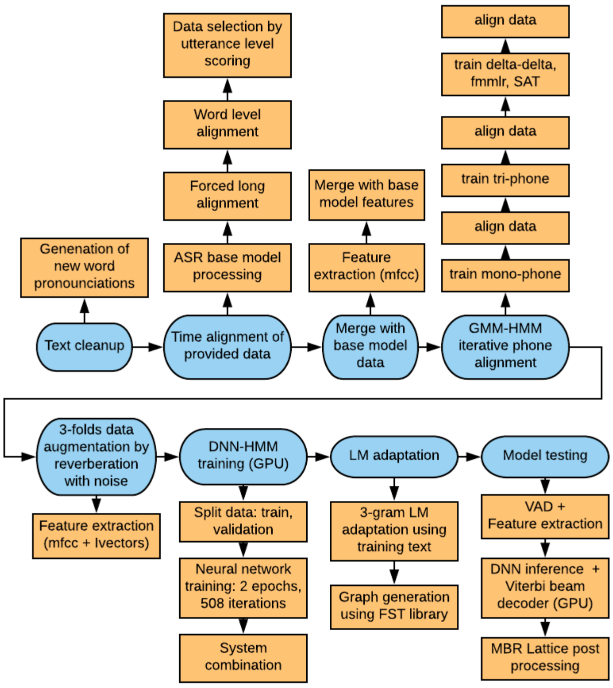
The resulting ASR model is tested using selected audio and transcriptions from the *dev2* subsection of the provided Iberspeech 2018 training and development data. Data preparation and selection procedure described in Section 2 is used also for the preparation of the test data since the development data also suffers from the mis-alignment problem.

Audio files are segmented using voice activity detection based on adaptive thresholding method proposed by Otsu [18]. Feature extraction is applied on produced segments in order to obtain high resolution MFCC's and iVectors with scaling factor 0.75. The extracted features are processed by a Viterbi beam decoder considering LM 3-gram probabilities coupled with TDNN tri-phone output probabilities. A GPU implementation of Kaldi *nnet3-latgen-faster* decoder is used with parameters *--beam=15.0, --lattice-beam=2.5, max-active=7000, --acoustic-scale=1.0, --frame-subsampling-factor=2*. Produced ASR lattice output is post-processed using *minimum bayes risk* (MBR) decoding [19] in order to find the best path result. ASR best path transcription results with word level timing information in *ctm* format are scored using the NIST *sclite* utility where ground truth transcriptions for selected utterances of *dev2* subsection are used in *stm* reference format. Words from the ASR results which are not in the start end time interval of any reference utterance are omitted in the scoring process. An average WER result of 23.9 is obtained in the final model testing. Base model WER result obtained using the same testing method and decoding parameters is 35.3.

Real time factor in the decoding process including VAD, feature extraction and lattice post-processing is 0.022 using a 4 cores Intel i7-4820K CPU @ 3.70GHz and a single NVidia GeForce GTX 1080Ti GPU.

The provided *test* data for Iberspeech 2018 competition is processed with the produced ASR model using the same procedure described above and resulting transcriptions are submitted in plain text format.

Figure 1: *ASR system training / testing components.*


Data selection by utterance level scoring
Word level alignment
Forced long alignment
ASR base model processing
Merge with base model features
Feature extraction (mfcc)
Generation of new word pronounciations
Text cleanup
Time alignment of provided data
Merge with base model data
GMM-HMM iterative phone alignment
align data
train delta-delta, fmmlr, SAT
align data
train tri-phone
align data
train mono-phone
3-folds data augmentation by reverberation with noise
Feature extraction (mfcc + Ivectors)
DNN-HMM training (GPU)
Split data: train, validation
Neural network training: 2 epochs, 508 iterations
System combination
LM adaptation
3-gram LM adaptation using training text
Graph generation using FST library
Model testing
VAD + Feature extraction
DNN inference + Viterbi beam decoder (GPU)
MBR Lattice post processing

4. Discussion

The main challenge in the ASR model training and testing process with the provided data was the wrong or missing utterance level time alignment information of the provided training and development data. Therefore, time alignment and selection of the provided training and development data was necessary prior to acoustic model training. Since only selected utterances are used also in the model testing stage, and the ASR results which do not match with the selected time intervals are omitted from the WER result obtained by the model testing process using the *dev2* subsection of the provided data, do not represent a WER result for all the data. If all the ground truth with true time alignments could be used, the obtained WER result is expected to be higher than is presented here since a selective process is used in the current WER calculation. However, the WER result calculated in the model testing process was helpful for the benchmarking of the produced ASR model with the base model, and some other experiment results prior to the production of the final ASR model.

Much higher WER results are obtained (average WER 58.3 for the final produced model, average WER 105.7 for the base model) when they are calculated using all the provided ground truth text of the *dev2* subsection ignoring provided wrong time information by using *txt* formatted ASR hypothesis files. A detailed analysis of the word level *sclite* logs shows that these values are not reliable because of mis-alignment problem of *sclite* utility usage without time information for such long reference and hypothesis files. This observation is the basis for the necessity of the two-step time alignment process used in this work prior to model training and testing.

A different value of frame-subsampling-factor compared to training process is chosen in the model testing (*frame-subsampling-factor=3* in training and *frame-subsampling-factor=2* in the testing) since it yields more accurate results in the testing of audio with Viterbi decoding using a LM.

5. Conclusion

The acoustic model building process with a data preparation and selection using a two-step time alignment procedure and utterance level thresholding with WER values yielded a good working acoustic model when the new training data is merged with the training data of the base model. The two-step time alignment procedure together with the utterance level data selection mechanism described in Section 2 enabled the usage of the provided data for the acoustic model training step of the ASR system generatıon. Model testing with the development data using an adapted version of the base LM showed a significant reduction in the WER results compared to the base model results used in the experiments.

6. Acknowledgements

The research leading to the results presented in this paper has been (partially) granted by the EU H2020 research and innovation program under grant number 769872

7. References

[1] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. W. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[2] A. Graves, N, Jailty, “Towards end-to-end speech recognition with recurrent neural networks,” ICML'14 Proceedings of the 31st International Conference on International Conference on Machine Learning, *Conference, June 21 - 26, Beijing, China, Proceedings*, 2014, pp. II-1764-II-1772

[3] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369–376.

[4] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks.” in Proceedings of INTERSPEECH, 2013, pp. 2345–2349.

[5] A. L. Zorrilla, N. Dugan, M. I. Torres, C. Glackin, G. Chollet and N. Cannings, “Some ASR experiments using Deep Neural Networks on Spanish databases,” IberSpeech 2016 *Conference, November 23 – 25, Lisbon, Portugal, Proceedings*, 2016, pp. 149-158

[6] SRILM Toolkit, http://www.speech.sri.com/projects/srilm

[7] Asunción Moreno, Dolors Poch, Antonio Bonafonte, Eduardo Lleida, Joaquim Llisterri, José B. Mariño, and Climent Nadeu, “Albayzin speech database: design of the phonetic corpus.,” *in EUROSPEECH*. 1993, ISCA.

[8] José miguel Benedí, Eduardo Lleida, Amparo Varona, María josé Castro, Isabel Galiano, Raquel Justo, Iñigo López De Letona, and Antonio Miguel, “Design and acquisition of a telephone spontaneous speech dialogue corpus in spanish: Dihana,” *in In Fifth LREC*, 2006, pp. 1636–1639.

[9] Luis J. Rodríguez and Torres M. Inés, “Spontaneous speech events in two speech databases of human-computer and human-human dialogs in spanish,” *Language and Speech, vol. 49, no. 3*, pp. 333–366, 2006.

[10] Henk van den Heuvel, Khalid Choukri, Christian Gollan, Asuncion Moreno, Djamel Mostefa: "TC-STAR: New language resources for ASR and SLT purposes” LREC 2006

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, *The Kaldi Speech Recognition Toolki*t, IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hawaii, IEEE Signal Processing Society, 2011.

[12] M. Karafiat, L. Burget, P. Matejka, O. Glembek, and J. Cernocky, “iVector-based discriminative adaptation for automatic speech recognition,” in 2011 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, Dec. 2011, pp. 152–157

[13] *Speech Recognition Scoring Toolkit (SCTK)*, National Institute of Standards and Technology, US Department of Commerce.

[14] N. Dugan, Forced alignment Python script, Intelligent Voice LTD, https://github.com/IntelligentVoice/Aligner

[15] Kaldi aspire recipe: https://github.com/kaldi-asr/kaldi/tree/master/egs/aspire

[16] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in Proceedings of INTERSPEECH, 2015.

[17] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, “JHU ASpIRE system: Robust LVCSR with TDNNs, i-vector Adaptation, and RNN-LMs,” in Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, 2015.

[18] N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. Systems, Man and Cybernetics, vol. 9, pp. 62–66, 1979.

[19] H. Xu, D. Povey, L. Mangu, J. Zhu, “Minimum Bayes Risk decoding and system combination based on a recursion for edit distance,” Computer Speech & Language, Volume 25, Issue 4, October 2011, pp. 802-828