LIBRISPEECH: AN ASR CORPUS BASED ON PUBLIC DOMAIN AUDIO BOOKS

Vassil Panayotov, Guoguo Chen*, Daniel Povey*, Sanjeev Khudanpur*

*Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA
{vassil.panayotov,dpovey}@gmail.com, {guoguo,khudanpur}@jhu.edu

ABSTRACT

This paper introduces a new corpus of read English speech, suitable for training and evaluating speech recognition systems. The LibriSpeech corpus is derived from audiobooks that are part of the LibriVox project, and contains 1000 hours of speech sampled at 16 kHz. We have made the corpus freely available for download, along with separately prepared language-model training data and pre-built language models. We show that acoustic models trained on LibriSpeech give lower error rate on the Wall Street Journal (WSJ) test sets than models trained on WSJ itself. We are also releasing Kaldi scripts that make it easy to build these systems.

Index Terms— Speech Recognition, Corpus, LibriVox

1. INTRODUCTION

The rapid increase in the amount of multimedia content on the Internet in recent years makes it feasible to automatically collect data for the purpose of training statistical models. This is particularly true when the source data is already organized into well curated, machine readable collections. The LibriVox project[1], a volunteer effort, is currently responsible for the creation of approximately 8000 public domain audio books, the majority of which are in English. Most of the recordings are based on texts from Project Gutenberg[2], also in the public domain.

Although the use of audio books for building synthetic voices [1, 2] has previously been investigated, we are not aware of any freely available read speech corpus in English that is suitable for training and testing speech recognition systems, and which is as large scale as the one we present here. The volunteer-supported speech-gathering effort Voxforge[3], on which the acoustic models we used for alignment were trained, contains a certain amount of LibriVox audio, but the dataset is much smaller than the one we present here, with around 100 hours of English speech, and suffers from major gender and per-speaker duration imbalances.

This paper presents the LibriSpeech corpus, which is a read speech data set based on LibriVox's audio books. The corpus is freely available[4] under the very permissive CC BY 4.0 license [3] and there are example scripts in the open source Kaldi ASR toolkit [4] that demonstrate how high quality acoustic models can be trained on this data.

Section 2 presents the long audio alignment procedure that we used in the creation of this corpus. Section 3 describes the structure of the corpus. In Section 4 we describe the process we used to build the language models, which we make available with this corpus. Finally in Section 5 we present experimental results on models trained on this data set, using both the LibriSpeech dev and test sets and Wall Street Journal (WSJ) [5] test sets.

[1]https://librivox.org/
[2]http://www.gutenberg.org
[3]http://www.voxforge.org
[4]http://www.openslr.org/12/

2. AUDIO ALIGNMENT

Most acoustic model training procedures expect that the training data come in the form of relatively short utterances, usually up to few tens of seconds in length, each with corresponding text. Therefore we need to align the audio recordings with the corresponding texts, and split them into short segments. We also aim to exclude segments of audio that might not correspond exactly with the aligned text. Our procedure is similar to that described in [6], and consists of two stages. (Note: we have since become aware of a different, phone-based approach [7]).

2.1. Text preprocessing, lexicon and LM creation

Each book's text is normalized by converting it into upper-case, removing the punctuation, and expanding common abbreviations and non-standard words [8]. Then the SRILM toolkit [9] is used to train a Witten-Bell [10] smoothed bigram language model on the text of that book. We base our lexicon on CMUdict, from which we remove the numeric stress markers; the pronunciations for out-of-vocabulary (OOV) words are generated with the Sequitur G2P toolkit [11]. In order to avoid possible problems with recognizing excessively long audio recordings, the audio chapters are split into segments of up to 30 minutes in length. The audio is then recognized using the *gmm-decode-faster* decoder from the Kaldi toolkit, trained on the VoxForge dataset. For this first decoding pass we use a triphone model discriminatively trained with Boosted MMI [12], based on MFCC [13] features processed with frame-splicing over 7 frames, followed by LDA, followed by a global semi-tied covariance (STC) transform [14].

2.2. First alignment stage

We use the Smith-Waterman alignment algorithm [15] to find the best single region of alignment between the recognized audio and the chapter text. This is like doing Levenshtein alignment [16], except we do not require it to consume the whole reference or hypothesis from the beginning to end, and it also has tunable rather than fixed weights for the different kinds of errors. From this we take the largest single region of similarity (which in most cases would be the entire chapter) and discard the rest, if any. Within that region of similarity, we mark a transcript word as being part of an "island of confidence" if it is part of an exact match with the reference whose length is 12 phones or more. We now split the audio into shorter segments, of 35 seconds or less, using a dynamic programming algorithm. We

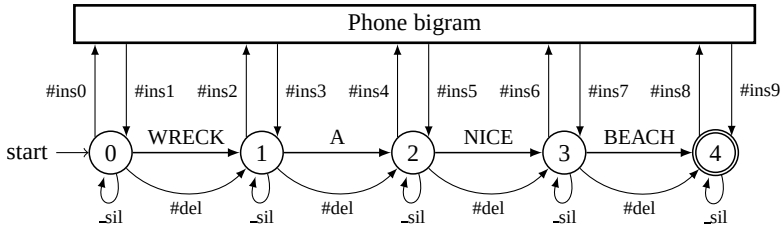

Fig. 1. Example grammar (G) acceptor for the second stage of the alignment algorithm

Reference: *A family of ten children will be always called a fine family ...*
a) **Hypothesis:** _sil A FAMILY OF TEN CHILDREN WILL #del ALWAYS #ins0016 _b _iy #ins0017 CALLED A FINE FAMILY _sil
Explanation: Transposition of "be" and "always".

Reference: *... upon her arm and ... I rushed towards her ...*
b) **Hypothesis:** _sil UPON HER ARM #ins0020 _s #ins0021 _sil AND ... I RUSHED _sil #ins0054 _ay _r _ah _sh _t #ins0055 TOWARDS HER
Explanation: Reader pronounces "arms" instead of "arm" and repeats "I rushed".

Reference: *Morning dawned before I arrived at the village of Chamounix ...*
c) **Hypothesis:** _sil MORNING DAWNED BEFORE I ARRIVED AT THE VILLAGE OF _sil #ins0018 _sh _ah _m _ow _n _iy #ins0021
Explanation: G2P error– the auto-generated dictionary entry is "CHAMOUNIX *CH AE M UW N IH K S*", which is wrong.

Fig. 2. Examples of typical text-audio discrepancies detected in second stage decoding. a) Chapter 1 of "Northanger Abbey" by J. Austen, read by Kara Shallenberg; b) Chapter 23 of "Frankenstein" by M. Shelley, read by Hugh McGuire; c) Chapter 15 of "Frankenstein" by M. Shelley, read by Gord Mackenzie

only allow it to split on silence intervals whose length is at least 0.5 second and which are inside an island of confidence. This allows us to generate, with reasonable confidence, a candidate text for each split piece of audio.

2.3. Second alignment stage

The goal of the second stage is to filter out segments where the candidate text obtained by the first stage has a high likelihood of being inaccurate. Possible sources of text-audio mismatch include inaccuracies in Project Gutenberg texts, reader-introduced insertions, deletions, substitutions and transpositions, and involuntary disfluencies [1, 17]. Other significant sources of mismatch that we noticed are inaccurate text normalization and grapheme-to-phoneme errors in the automatically generated pronunciations.

In this second stage of alignment, we use a custom-generated decoding graph for each segment. The decoding graph, diagrammed in Figure 1, is formed from a combination of the linear sequence of words in the transcript with a generic phone-level bigram language model. Our aim is to use the phone-level bigram to allow arbitrary insertions between words in the transcript, or replacement of words in the transcript; we will reject any utterance whose decoding shows any deviation from the transcript. We also experimented with a single-phone filler model to model errors, but found the phone-level bigram was more effective at finding segments with inaccurate transcripts.

The most obvious way to generate the decoding graph would be to include multiple copies of the phone-level bigram graph, but this would lead to very large decoding graphs. Instead we use a single copy of the bigram part of the decoding graph (Figure 1), but we modify the decoder so that after entering the bigram part of the model from word-position x in the transcript, we may only return at position x (corresponding to an insertion between words) or position $x+1$ (corresponding to a substitution or deletion of a word). This is like a pushdown transducer that can only store one item in the pushdown store.

In this second decoding stage, we use a speaker-adapted model [18, 19] with fMLLR transforms estimated at the speaker level, based on the transcript generated by the first decoding pass.

In most cases this algorithm succeeds in detecting text-audio mismatches, especially for native speakers. There are also instances of false rejections. A common problem is, for example, the "assimilation" of a short, 1-2 phone word into a neighboring silence period, which leads to an erroneous detection of deletion from the audio. However, since the original amount of audio in the audiobooks is so large, we can afford to lose a certain percentage of it. Figure 2 shows examples of the kinds of errors that we typically find by applying this method.

The whole alignment process took approximately 65 hours on two Amazon EC2 cc2.8xlarge instances, to produce an initial set of aligned audio of size approximately 1200 hours.

2.4. Data segmentation

The second stage of alignment, which we described above, gives us a subset of the audio segments of length up to 35 seconds, that have a good likelihood of having accurate transcripts. Next we break these long segments up into smaller segments. We used two different methods for this. For training data, our rule was to split on any silence interval longer than 0.3 seconds. For test data, we only allowed splits if those intervals coincided with a sentence break in the reference text. The idea was that data split at sentence breaks is likely to be easier to recognize from a language modeling point of view.

3. DATA SELECTION AND CORPUS STRUCTURE

3.1. Data selection

To select the audio recordings for inclusion into the corpus we use LibriVox's API[5] to collect information about the readers, the audio book projects in which they participated, and the chapters of books that they read. The URLs for audio files and reference texts were obtained by matching the information from LibriVox's API with the metadata records from the Internet Archive[6] and Project Gutenberg's RDF/XML files[7]. For a small fraction of audiobooks no exact match for the title was found in Project Gutenberg, so to improve coverage we allowed a fuzzy matching of titles.

In order to guarantee that there was no speaker overlap between the training, development and test sets, we wanted to ensure that each recording is unambiguously attributable to a single speaker. To that end we exclude such LibriVox genres as, for example, "Dramatic Reading", which include predominantly multi-reader audio chapters. As an extra precaution, in the final post-processing step of the alignment processing the recordings are processed with the LIUM speaker diarization toolkit [20] to automatically detect multi-speaker chapters. A custom GUI application was written, that makes use of the text-audio alignment information and the speaker diarization information, to allow for quick inspection and filtering out of the remaining multi-speaker recordings. This application also made it possible to quickly produce gender information for the speakers and to discard a small number of recordings that had excessive audio quality problems.

We ensured a gender balance at the speaker level and in terms of the amount of data available for each gender.

3.2. Corpus partitions

The size of the corpus makes it impractical, or at least inconvenient for some users, to distribute it as a single large archive. Thus the training portion of the corpus is split into three subsets, with approximate size 100, 360 and 500 hours respectively. A simple automatic procedure was used to select the audio in the first two sets to be, on average, of higher recording quality and with accents closer to US English. An acoustic model was trained on WSJ's si-84 data subset and was used to recognize the audio in the corpus, using a bigram LM estimated on the text of the respective books. We computed the Word Error Rate (WER) of this automatic transcript relative to our reference transcripts obtained from the book texts.

The speakers in the corpus were ranked according to the WER of the WSJ model's transcripts, and were divided roughly in the middle, with the lower-WER speakers designated as "clean" and the higher-WER speakers designated as "other". From the "clean" pool, 20 male and 20 female speakers were drawn at random and assigned to a development set. The same was repeated to form a test set. For each dev or test set speaker, approximately eight minutes of speech are used, for total of approximately 5 hours and 20 minutes each. Note that, as mentioned in Section 2.4, we use a different segmentation procedure for development and test data, than for training data.

The rest of the audio in the "clean" pool was randomly split into two training sets with approximate size 100 and 360 hours respectively. For each speaker in these training sets the amount of speech was limited to 25 minutes, in order to avoid major imbalances in per-speaker audio duration.

[5] https://librivox.org/api/info

[6] http://blog.archive.org/2011/03/31/how-archive-org-items-are-structured/

[7] http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs

subset	hours	per-spk minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Table 1. Data subsets in LibriSpeech

The "other" pool was similarly split into test and development sets, and a single training set of approximately 500 hours. For this pool, however we did not choose the development and test sets at random; instead we deliberately chose more challenging data. The WER we computed using the WSJ models was used to rank the speakers in order of increasing difficulty, and the speakers for the test and development set were randomly chosen from the third quartile of this sorted list. Table 1 provides a summary of all subsets in the corpus.

4. LANGUAGE MODELS

To make it easy to reproduce the results we report here, we have released language model training data and pre-built language models online[8], along with the text data that we used to build the language models. The language model training material is carefully selected to avoid any overlap with the texts which appear in the test and development sets.

The source material for these language models is Project Gutenberg books. All books, in their entirety, on which the test and development sets are based were filtered out, as well as any book whose title has cosine similarity, over letter 3-grams, greater than 0.7 with any of the titles of these books. After filtering on titles, the text of approximately 22 000 candidate books was downloaded from Project Gutenberg. An inverted index of all 5-grams, with stop words deleted, is built for the books in the test and development sets. All candidate books are then checked against this index and each book for which more than one percent of the 5-grams which appear in it, appear in any of the books in the test and development sets, is removed from the candidate set. The method is effective for finding shared text such as, for example, popular fairy tales which are present in more than one fairy tales collection, long citations of poems in other works, and so on. We used other heuristics to also filter out texts such as numeric tables, sequences from the Human Genome Project, and other types of documents that were deemed inappropriate for language model training.

After the above steps, approximately 14 500 public domain books, containing around 803 million tokens in total and 900 000 unique words, remained.

To select a lexicon, the words in the corpus were ranked by frequency, and the 200 000 most frequent words were selected. Around one third of these words are present in the CMU pronunciation dictionary, accounting for around 97.5% of all tokens in the evaluation sets; we generated pronunciations for the remaining words using the Sequitur G2P toolkit [11]. Modified Kneser-Ney smoothed 3- and 4-grams [21, 22] are trained. The perplexity for the 3-gram model is 170, and the out of vocabulary token rate is approximately 0.4%

[8] http://www.openslr.org/11/

on average. For the 4-gram language model the perplexity is around 150.

5. EXPERIMENTS

In this section we present decoding results using models trained using various amounts of LibriSpeech data, and on WSJ data, on both LibriSpeech and WSJ test sets. The recordings available from LibriVox are not completely ideal for training acoustic models for other domains, because the audio is MP3-compressed and because the site's guidelines for upload recommend noise removal[9] and volume normalization[10]. These practices are not consistently enforced, however, so there is a significant fraction of noisy and non-processed audio available, combined with audio that has been subjected to automatic noise removal.

In order to assess the performance of the acoustic models on non-compressed audio we use the Wall Street Journal read speech corpus [5], as a baseline. We employ language models, trained on the text material the WSJ corpus provides, in conjunction with acoustic models trained on the LibriSpeech data to decode WSJ's test sets, and compare the results with those for state-of-the-art models trained on WSJ's own si-284 set (which contains 82 hours of speech data). The WSJ results we present in Table 2 are for the "open-vocabulary" (60K) test condition, using not the standard 60K word dictionary supplied with WSJ but an extended version that we built to cover more of the words that appear in the WSJ language models. For the language model we used a pruned version of the standard trigram language model that is distributed with the WSJ corpus. The acoustic models, referred to as *SAT* in the tables, are speaker-adapted GMM models [18, 19], and those referred to as *DNN*, are based on deep neural networks with p-norm non-linearities [23], trained and tested on top of fMLLR features. The models estimated on LibriSpeech's training data are named after the amount of audio they were built on. The models marked with *460h* are trained on the union of the "train-clean-100" and "train-clean-360" subsets, and those marked with *960h* are trained on all of LibriSpeech's training sets.

Acoustic model		**eval'92**	**dev'93**	**eval'93**
LS	SAT 100h	5.72	10.10	9.14
	SAT 460h	5.49	8.96	7.69
	SAT 960h	5.33	8.87	8.32
	DNN 100h	4.08	7.31	6.73
	DNN 460h	3.90	6.75	5.95
	DNN 960h	3.63	6.52	5.66
WSJ	SAT si-284	6.26	9.39	9.19
	DNN si-284	3.92	6.97	5.74

Table 2. WERs on WSJ's test sets under the "open vocabulary" (60K) test condition

Similarly LibriSpeech's language models are used with WSJ acoustic models to decode LibriSpeech's test sets. For these tests the results in Table 3 were obtained by rescoring with the full 4-gram language model from Section 4.

In order to be able to rescore lattices using large language models in a memory efficient manner, we implemented a new rescoring tool, which is now part of the Kaldi toolkit. Table 4 shows the word error rates for language models of different size. The first pass decoding is performed using the 3-gram model pruned with threshold 3×10^{-7} using SRILM's pruning method; the other numbers are obtained through lattice rescoring.

Acoustic model		**dev-clean**	**test-clean**	**dev-other**	**test-other**
LS	SAT 100h	8.19	9.32	29.31	31.52
	SAT 460h	7.26	8.34	26.27	28.11
	SAT 960h	7.08	8.04	21.14	22.65
	DNN 100h	5.93	6.59	20.42	22.52
	DNN 460h	5.27	5.78	17.67	19.12
	DNN 960h	4.90	5.51	12.98	13.97
WSJ	SAT si-284	10.87	12.44	39.44	41.26
	DNN si-284	7.80	8.49	27.39	30.01

Table 3. WERs on LibriSpeech's test sets; all results are obtained by rescoring with a 4-gram language model.

Language model	**dev-clean**	**test-clean**	**dev-other**	**test-other**
3-gram prn. thresh. 3e-7	7.54	8.02	18.51	19.41
3-gram prn. thresh. 1e-7	6.57	7.21	16.72	17.66
3-gram full	5.14	5.74	13.89	14.77
4-gram full	4.90	5.51	12.98	13.97

Table 4. LM rescoring results for the 960 hour DNN model

6. CONCLUSIONS

We have automatically aligned and segmented English read speech from audiobooks with the corresponding book text, and filtered out segments with noisy transcripts, in order to produce a corpus of English read speech suitable for training speech recognition systems. We have demonstrated that models trained with our corpus do better on the standard Wall Street Journal (WSJ) test sets than models built on WSJ itself – the larger size of our corpus (1000 hours, versus the 82 hours of WSJ's si-284 data) outweighs the audio mismatch. We are releasing this corpus online[11] and have introduced scripts into the Kaldi speech recognition toolkit so that others can easily replicate these results.

7. REFERENCES

[1] K. Prahallad, *Automatic building of synthetic voices from audio books*, Ph.D. thesis, CMU, Pittsburgh, 2010.

[2] S. King and V. Karaiskos, "The Blizzard Challenge 2012," in *Proceedings Blizzard Workshop*, 2012.

[3] "Creative Commons Attribution 4.0 International Public License," `https://creativecommons.org/licenses/by/4.0/`, November 2013.

[4] D. Povey, A. Ghoshal, et al., "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[9] http://wiki.librivox.org/index.php/Noise_Cleaning

[10] http://wiki.librivox.org/index.php/Questions_and_Answers

[11] http://www.openslr.org/12/

[5] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[6] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings," in *In Proc. Interspeech*, 2006.

[7] X. Anguera, J. Luque, and C. Gracia, "Audio-to-text alignment for speech recognition with very limited resources," in *Interspeech*, 2014.

[8] R. Sproat et al., "Normalization of non-standard words," *Computer Speech & Language*, vol. 15, no. 3, pp. 287–333, 2001.

[9] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *ICSLP*, 2002.

[10] I.H. Witten and T.C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, 1991.

[11] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion.," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[12] D. Povey and D. Kanevsky and B. Kingsbury and B. Ramabhadran and G. Saon and K. Visweswariah, "Boosted MMI for Feature and Model Space Discriminative Training," in *ICASSP*, 2008.

[13] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[14] M.J.F. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.

[15] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[16] V.I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, vol. 10, pp. 707, 1966.

[17] N. Braunschweiler, M. J. F. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings.," in *INTERSPEECH*. 2010, pp. 2222–2225, ISCA.

[18] M. J. F. Gales and P. C. Woodland, "Mean and Variance Adaptation Within the MLLR Framework," *Computer Speech and Language*, vol. 10, pp. 249–264, 1996.

[19] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A Compact Model for Speaker-Adaptive Training," in *ICSLP*, 1996.

[20] S. Meignier and T. Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, Dallas (Texas, USA), March 2010.

[21] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *ICASSP*, 1995, vol. 1, pp. 181–184.

[22] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.

[23] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 215–219.