TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation

François Hernandez[1], Vincent Nguyen[1], Sahar Ghannay[2],
Natalia Tomashenko[2], and Yannick Estève[2]

[1] Ubiqus, Paris, France flast@ubiqus.com
https://www.ubiqus.com
[2] LIUM, University of Le Mans, France first.last@univ-lemans.fr
https://lium.univ-lemans.fr/

Abstract. In this paper, we present TED-LIUM release 3 corpus dedicated to speech recognition in English, that multiplies by more than two the available data to train acoustic models in comparison with TED-LIUM 2. We present the recent development on Automatic Speech Recognition (ASR) systems in comparison with the two previous releases of the TED-LIUM Corpus from 2012 and 2014. We demonstrate that, passing from 207 to 452 hours of transcribed speech training data is really more useful for end-to-end ASR systems than for HMM-based state-of-the-art ones, even if the HMM-based ASR system still outperforms end-to-end ASR system when the size of audio training data is 452 hours, with respectively a Word Error Rate (WER) of 6.6% and 13.7%. Last, we propose two repartitions of the TED-LIUM release 3 corpus: the *legacy* one that is the same as the one existing in release 2, and a new one, calibrated and designed to make experiments on *speaker adaptation*. Like the two first releases, TED-LIUM 3 corpus will be freely available for the research community.

Keywords: Speech recognition · opensource corpus · deep learning · speaker adaptation · TED-LIUM.

1 Introduction

Back in May 2012 and May 2014, the LIUM team released two versions (respectively 118 hours of audio and 207 hours of audio) from the TED conference videos which were since widely used by the ASR community for research purpose. These corpora were called TED-LIUM, release 1 and release 2, presented respectively in [11] and [12]. Ubiqus joined these efforts to pursue the improvements both from an increased data standpoint as well as from a technical achievement. We believe that this corpus has become a reference and will still be used by the community to improve further the results. In this paper, we present a few enhancements about the dataset, by using a new engine to realign the original data, leading to an increased amount of audio/text, and by adding new TED talks which, combined with the new alignment process, gives us 452 hours of aligned audio. A new data distribution is also proposed that is more fitted to experiment speaker adaptation techniques in addition to the *legacy* distribution already used on TED-LIUM release 1 and 2. Section 2 gives details about the new TED-LIUM 3 corpus. We present experimental results with different ASR architectures, by using Time Delay Neural Network (TDNN) [6] and Factored TDNN (TDNN-F) acoustic models [8] on the *legacy* distribution of TED-LIUM 3 in section 3, and also exploring the use of pure neural end-to-end system in section 4. In section 5, we report experimental results obtained on the *speaker adaptation* distribution by exploiting TDNN-Long Short-Term Memory (TDNN-LSTM) acoustic models [7] and Gaussian Mixture Model-derived (GMMD) features [14]. The last section is dedicated to discussion and conclusion.

2 TED-LIUM 3 Corpus description

2.1 Data, alignment and filtering

All raw data (acoustic signals and closed captions) were extracted from the TED website. For each talk, we built a `sphere` audio file, and its corresponding transcript in `stm` format. The text from each `.stm` file was automatically aligned to the corresponding `.sph` file using the Kaldi toolkit [9]. This consists in the adaptation of existing scripts [3], meant to first decode the audio files with a biased language model, and then align the obtained `.ctm` file with the reference transcript. To maximize the quality of alignments, we used our best model (at the time of corpus preparation) trained on the previous release of the TED-LIUM corpus. This model achieved a WER of 9.2% on both development and test sets without any rescoring. This means the ratio of aligned speech versus audio from the original 1,495 talks of releases 1 and 2 has changed, as well as the quantity of words kept. It increased by around 40% the amount of usable data from the same basis files (Table 1). In the previous release, aligned speech represented only around 58.9% of all audio duration (351 hours). With these new alignments, we now cover around 83.0% of audio.

Table 1. Maximizing alignments - TED-LIUM release 2 talks

Characteristic	Alignments		Evolution
	Original	New	
Speech	207h	290h	40.1%
Words	2.2M	3.2M	43.1%

A first set of experiments was conducted to compare equivalent systems trained on the two sets of data (Table 2). With strictly equivalent models, there is no clear improvement of results for the proposed new alignments. Yet, there are no degradation of performance either. We will show in further experiments that the increased amount of data will not just be harmless but also useful.

[3] `https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/steps/cleanup/segment_long_utterances.sh`

Table 2. Comparison of training on original and new alignments for TED-LIUM release 2 data (Experiments conducted with the Kaldi toolkit - details in Section 3)

Model (rescoring)	Original - 207h		New - 290h	
	Dev	Test	Dev	Test
HMM-GMM (none)	19.0%	17.6%	18.7%	17.2%
HMM-GMM (Ngram)	17.8%	16.5%	17.7%	16.1%
HMM-TDNN-F (none)	8.5%	8.3%	8.2%	8.3%
HMM-TDNN-F (Ngram)	7.8%	7.8%	7.7%	7.9%
HMM-TDNN-F (RNN)	6.8%	6.8%	6.6%	6.7%

2.2 Corpus distribution: legacy version

The whole corpus is released as what we call a *legacy* version, for which we keep the same development and test sets as the first releases. Table 3 summarizes the characteristics of text and audio data of the new release of the TED-LIUM corpus. Statistics from the previous and new releases are presented, as well as the evolution between the two. Additionally, we mention that aligned speech (including some noises and silences) represents around 82.6% of audio duration (540 hours).

Table 3. TED-LIUM 3 corpus characteristics

Characteristic	Corpus		Evolution
	v2	v3	
Total duration	207h	452h	118.4%
- Male	141h	316h	124.1%
- Female	66h	134h	103.0%
Mean duration	10m 12s	11m 30s	12.7%
Number of unique speakers	1242	2028	63.3%
Number of talks	1495	2351	57.3%
Number of segments	92976	268231	188.5%
Number of words	2.2M	4.9M	122.7%

2.3 Corpus distribution: speaker adaptation version

Speaker adaptation of acoustic models (AMs) is an important mechanism to reduce the mismatch between the AMs and test data from a particular speaker, and today it is still a very active research area. In order to design a suitable corpus for exploring speaker adaptation algorithms, additional factors and data set characteristics, such as number of speakers, amount of pure speech data per speaker, and others, should be taken into account. In this paper, we also propose and describe the training, development and test data sets specially designed for the speaker adaptation task. These datasets are obtained from the proposed TED-LIUM 3 training corpus, but the development and test sets are more balanced and representative in characteristics (number of speakers, gender, duration) than the original ones and more suitable for speaker adaptation experiments. Also, for the development and test datasets we chose only speakers who are not present in the training data set in other talks. The statistics for the proposed data sets are given in Table 4.

Table 4. Data sets statistics for the speaker adaptation task. Unlike the other tables, the duration is calculated only for pure speech (excluding silence, noise, etc.).

Characteristic		Data set		
		Train	Dev.	Test
Duration of speech, hours	Total	346.17	3.73	3.76
	Male	242.22	2.34	2.34
	Female	104.0	1.39	1.41
Duration of speech per speaker, minutes	Mean	10.7	14.0	14.1
	Min.	1.0	13.6	13.6
	Max.	25.6	14.4	14.5
Number of speakers	Total	1938	16	16
	Male	1303	10	10
	Female	635	6	6
Number of words	Total	4437K	47753	43931
Number of talks	Total	2281	16	16

3 Experiments with state-of-the-art HMM-based ASR system

We conducted a first set of experiments on the TED-LIUM release 2 and 3 corpora using the Kaldi toolkit. These experiments were based on the existing recipe [4], mainly changing model configurations and rescoring strategies. We also kept the lexicon from the original release, containing 159,848 entries. For this experiments, and for all the other ones in the paper, no *glm* files were applied to deal with equivalences between word spelling (*e.g.* doctor *vs.* dr).

3.1 Acoustic models

All experiments were conducted using chain models [10] with the now well-known TDNN architecture [6] as well as the recent TDNN-F architecture [8]. Training audio samples were randomly perturbed in speed and volume during the training process. This approach is commonly called *audio augmentation* and is known to be benefit for speech recognition [5].

3.2 Language model

Two approaches were used, both aiming at rescoring lattices. The first one is a N-gram model of order 4 trained with the *pocolm* toolkit[5], which was pruned to 10 million N-grams. We also considered a RNNLM with letter-based features and importance sampling [17], coupled with a pruned approach to lattice-rescoring

[4] `https://github.com/kaldi-asr/kaldi/tree/master/egs/tedlium/s5_r2`

[5] `https://github.com/danpovey/pocolm`

[16]. The RNNLM we kept was a mixture of three TDNN layers with two interspersed LSTMP layers [13] containing around 10 million parameters. The latter helps reducing drastically word error rate. We used the same corpus and vocabulary in both methods, which are the ones released along with TED-LIUM release 2. These experiments were conducted prior to the full preparation of the new release, so we only appended text from the original alignments of release 2 to this corpus. In total, the textual corpus used to train language models contains around 255 millions of words. These source data are described in [12].

3.3 Experimental results

In this section we present the recent development on Automatic Speech Recognition (ASR) systems that can be compared with the two previous releases of the TED-LIUM Corpus from 2012 and 2014. While the first version of the corpus achieved a 17.4% of WER at that time, the second one decreased it to 11.1% using additional data and Deep Neural Network (DNN) techniques.

TDNN Our basis chain-TDNN setup is based on 6 layers with batch normalization, and a total context of (-15,12). Prior tuning experiments on TED-LIUM release 2 showed us that the model did not improve beyond the dimension of 450. More than doubling the training data allows to train bigger, and better, models of the same architecture as shown in Table 5.

Table 5. Tuning the hidden dimension of chain-TDNN setup on TED-LIUM release 3 corpus

Dimension	WER		WER - Ngram		WER - RNN	
	Dev	Test	Dev	Test	Dev	Test
450	9.0%	9.1%	8.0%	8.4%	6.9%	7.3%
600	8.7%	8.9%	8.0%	8.4%	6.6%	7.3%
768	8.3%	8.6%	7.6%	8.1%	6.5%	7.0%
1024	8.3%	8.5%	7.5%	8.0%	6.4%	6.9%

As part of experiments in tuning Kaldi models, it appeared that a form of L2 regularization could help in training for longer with less risk to overfit. This was implemented in Kaldi as the `proportional-shrink` option. Some tuning on TED-LIUM release 2 data gave the best result for a value of 20. All experiments presented in Table 5 were realized with this value to keep a consistent baseline. Aiming to reduce even more the WER, and with time constraints, we chose to train again the model with dimension 1024, with a proportional-shrink value of 10 (as we approximately doubled the size of the corpus). After RNNLM lattice-rescoring, the WER went down to 6.2% on the dev set and 6.7% on the test set.

TDNN-F As a final set of experiments, we tried the recently introduced factorized TDNN approach, which gave again significant improvements in WER for both TED-LIUM release 2 and 3 corpora (Table 6).

Table 6. Factorized TDNN experiments on TED-LIUM release 2 and 3 corpora

Corpus	Model	WER		WER - Ngram		WER - RNN	
		Dev	Test	Dev	Test	Dev	Test
r2	TDNN-F - 11 layers - 1280/256 - ps20	8.5%	8.3%	7.8%	7.8%	6.8%	6.8%
r3	TDNN-F - 11 layers - 1280/256 - ps10	7.9%	8.0%	7.2%	7.5%	6.1%	6.6%

4 Experiments with fully neural end-to-end ASR system

We also conducted experiments to evaluate the impact of adding data in the training corpus in order to build a neural end-to-end ASR. The system we experimented does not use a vocabulary to produce words, since it emits sequences of characters.

4.1 Model architecture

The fully end-to-end architecture used in this study is similar to the Deep Speech 2 neural ASR system proposed by Baidu in [1]. This architecture is composed of nc convolution layers (CNN), followed by nr uni or bidirectional recurrent layers, a lookahead convolution layer [15], and one fully connected layer just before the softmax layer, as shown in Figure 1. The system is trained end-to-end by using the CTC loss function [3], in order to predict a sequence of characters from the input audio. In our experiments we used two CNN layers and six bidirectional recurrent layers with batch normalization as mentioned in [1]. Given an utterance x^i and label y^i sampled from a training set $X = (x^1, y^1), (x^2, y^2), ...,$ the RNN architecture has to train to convert an input sequence x^i into a final

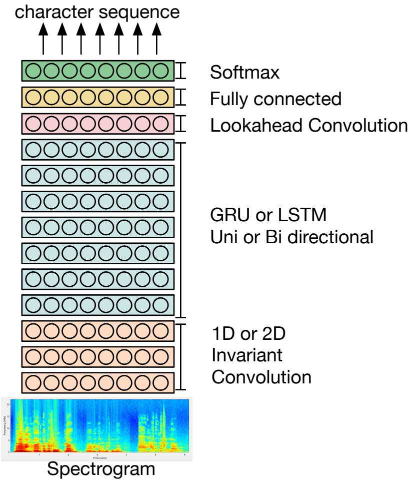

Fig. 1. Deep Speech 2 -like end-to-end architecture for speech recognition.

transcription y^is. For notational convenience, we drop the superscripts and use x to denote a chosen utterance and y the corresponding label. The RNN takes as input an utterance x represented by a sequence of log-spectrograms of power normalized audio clips, calculated on 20ms windows. As output, all the characters l of a language alphabet may be emitted, in addition to the space character used to segment character sequences into word sequences (space denotes word boundaries) and a *blank* character useful to absorb the difference in a time series length between input and output in the CTC framework. The RNN makes a prediction $p(l_t|x)$ at each output time step t. At test time, the CTC model can be coupled with a language model trained on a large textual corpus. A specialized beam search CTC decoder [4] is used to find the transcription y that maximizes:

$$Q(y) = log(p(l_t|x)) + \alpha log(pLM(y)) + \beta wc(y) \tag{1}$$

where wc(y) is the number of words in the transcription y. The weight α controls the relative contributions of the language model and the CTC network. The weight β controls the number of words in the transcription.

4.2 Experimental results

Experiments were made on the *legacy* distribution of the TED-LIUM 3 corpus in order to evaluate the impact on WER of training data size for an end-to-end speech recognition system inspired from Deep Speech 2. In these experiments, we used an open source Pytorch implementation[6].

Three training datasets were used: TED-LIUM 2 with original alignment (207h of speech), TED-LIUM 2 with new alignment (290h), and TED-LIUM 3 (452h), as presented in section 2.1 and section 2.2. They correspond to the three possible abscissa values (207, 290, 452) in figure 4.2. Whatever the training dataset, the ASR tuning and the evaluation were respectively made on the TED-LIUM release 2 development and test dataset, like in the experiments presented in section 3.3. Figure 4.2 presents both results in WER (left side), and Character Error Rate (CER, right side) on the test dataset. Evaluation in CER is interesting because the end-to-end ASR system is trained to product sequence of characters, instead of sequence of words.

For each training dataset, three configurations have been tested:

- the *Greedy* one, in blue in Figure 4.2 that consists in evaluating sequences of characters directly emitted from the neural network by gluing all the characters (including spaces to delimit words);
- the *Greedy+augmentation* one, in red, which is similar to the Greedy one, but in which each training audio samples is randomly perturbed in gain and tempo for each iteration [5];
- the *Beam+augmentation* one, in brown, got by applying a language model through a beam search decoding on the top of the neural network hypotheses using the Greedy+augmentation configuration. This language model is the *cantab-TEDLIUM-pruned.lm3* provided with the Kaldi TEDLIUM recipe.

[6] `https://github.com/SeanNaren/deepspeech.pytorch`

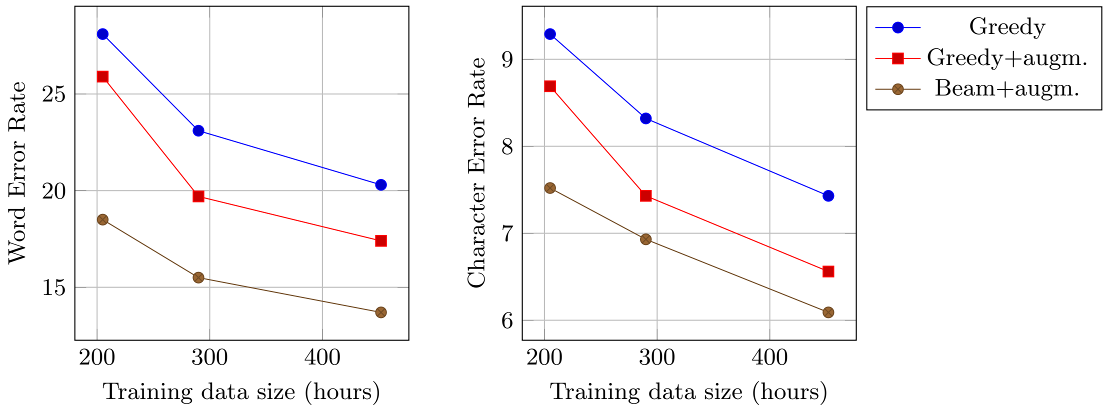

Fig. 2. Word error rate (left) and character error rate (right) on the TED-LIUM 3 *legacy* test data for three end-to-end configurations according to the training data size

As expected, the best results in WER and CER are reached by the *Beam+augmentation* configuration, with 13.7% of WER and 6.1% of CER. Whatever the configuration, increasing training data size improves strongly the transcription quality: for instance, while with the original TED-LIUM 2 data the Greedy mode reached 28.1% of WER, with TED-LIUM 3 it reaches 20.3%. We can notice that with TED-LIUM 3, the *Greedy+augmentation* configuration gets a lower WER than the *Beam+augmentation* one when trained with the original TED-LIUM 2 data. This shows that increasing the training data size for the pure end-to-end architecture offers an higher potential of WER reduction than using an external language model in a beam search decoding.

5 Experiments with the *speaker adaptation* distribution

In this section, we present results with speaker adaptation experiments on the adaptation version of the corpus described in Section 2.3. In this series of experiments, we trained two AMs: a speaker-independent AM and speaker adaptive trained (SAT) AM. The SAT AM was trained using GMM-derived (GMMD) features [14], adapted with maximum a posteriori adaptation (MAP) algorithm [2]. The Kaldi speech recognition toolkit [9] was used for these experiments. Both AMs have TDNN-LSTM topology described in [7], and differ only in the input features. For the SI AM, 40-dimensional Mel-frequency cepstral coefficients (MFCCs) without cepstral truncation were used as the input into the neural network. Input features for SAT TDNN-LSTM AMs were 168-dimensional speaker-adapted GMMD features concatenated with conventional 40-dimensional MFCCs. Both AMs were trained using LF-MMI criterion [10] and 3-fold reduced frame rate. The 4-gram pruned LM was used for the evaluation. The adaptation experiments were conducted in an unsupervised mode on the test data using transcriptions obtained from the first decoding pass by the SI baseline AM. Results in terms of WER are presented in Table7.

Table 7. Speaker adaptation results for the speaker adaptation task (on the corpus described in Section 2.3)

Model	WER,% for Dev.	WER,% for Test
SI	6.74	6.52
SAT	6.35	6.26

6 Discussion and Conclusion

In this paper we proposed a new release of the TED-LIUM corpus, that doubles the quantity of audio with aligned text for acoustic model training. We showed that increasing this training data reduces very slightly the word error rate obtained by a state-of-the-art HMM-based ASR system, passing from 6.8% (release 2) to 6.6% (release 3) on the *legacy* test data (and from 6.8% to 6.1% on the *legacy* dev data). To measure the recent advances realized in ASR technology, this word error rate can be compared to the 11.1% reached by a such state-of-the-art system in 2014 [11]. We were also interested to emergent neural end-to-end ASR technology, known as very voracious in training data. We noticed that without external knowledge, *i.e* by using only aligned audio from TED-LIUM 3, such technology gets 17.4% of WER, that is exactly the WER reached by state-of-the-art ASR technology in 2012 with the TED-LIUM 1 training data. Assisted by a classical 3-gram language model used in a beam search on top of the end-to-end architecture, this WER decreases to 13.7% with the TED-LIUM 3 training data, while with the TED-LIUM 2 training data the same system reached a WER of 20.3%. Increasing training data composed of audio with aligned text for this kind of ASR architecture seems still very important in comparison to the HMM-based ASR architecture that reaches a plateau on such TED data, with a low WER of 6.6%. Last, we propose a new data distribution dedicated to experiment on speaker adaptation, and propose some results than can be considered as a baseline for future work.

References

1. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al.: Deep speech 2: End-to-end speech recognition in english and mandarin. In: International Conference on Machine Learning. pp. 173–182 (2016)
2. Gauvain, J.L., Lee, C.H.: Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. IEEE Trans. Speech and Audio Proc. **2**, 291–298 (1994). https://doi.org/10.1109/89.279278
3. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. ACM (2006)
4. Hannun, A.Y., Maas, A.L., Jurafsky, D., Ng, A.Y.: First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns. arXiv preprint arXiv:1408.2873 (2014)

5. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
6. Peddinti, V., Povey, D., Khudanpur, S.: A time delay neural network architecture for efficient modeling of long temporal contexts. In: INTERSPEECH (2015)
7. Peddinti, V., Wang, Y., Povey, D., Khudanpur, S.: Low latency acoustic modeling using temporal convolution and lstms. IEEE Signal Processing Letters **25**(3), 373–377 (2018)
8. Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., Khudanpur, S.: Semi-orthogonal low-rank matrix factorization for deep neural networks. In: INTERSPEECH (2018 - submitted)
9. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society (Dec 2011), iEEE Catalog No.: CFP11SRW-USB
10. Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S.: Purely sequence-trained neural networks for asr based on lattice-free mmi. In: INTERSPEECH (2016)
11. Rousseau, A., Deléglise, P., Estève, Y.: TED-LIUM: an automatic speech recognition dedicated corpus. In: LREC. pp. 125–129 (2012)
12. Rousseau, A., Deléglise, P., Estève, Y.: Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In: LREC. pp. 3935–3939 (2014)
13. Sak, H., Senior, A.W., Beaufays, F.: Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: INTERSPEECH (2014)
14. Tomashenko, N., Khokhlov, Y.: Speaker adaptation of context dependent deep neural networks based on MAP-adaptation and GMM-derived feature processing. In: INTERSPEECH. pp. 2997–3001 (2014)
15. Wang, C., Yogatama, D., Coates, A., Han, T., Hannun, A., Xiao, B.: Lookahead convolution layer for unidirectional recurrent neural networks (2016)
16. Xu, H., Chen, T., Gao, D., Wang, Y., Li, K., Goel, N., Carmiel, Y., Povey, D., Khudanpur, S.: A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition (2017)
17. Xu, H., Li, K., Wang, Y., Wang, J., Kang, S., Chen, X., Povey, D., Khudanpur, S.: Neural network language modeling with letter-based features and importance sampling (2017)