IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

MLLP-UPV and RWTH Aachen Spanish ASR Systems for the IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge

Javier Jorge[1], *Adrià Martínez-Villaronga*[1], *Pavel Golik*[2], *Adrià Giménez*[1], *Joan Albert Silvestre-Cerdà*[1], *Patrick Doetsch*[2], *Vicent Andreu Císcar*[3], *Hermann Ney*[2], *Alfons Juan*[1] *and Albert Sanchis*[1]

[1] Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València (Spain)
[2] Human Language Technology and Pattern Recognition, RWTH Aachen University (Germany)
[3] Escola Tècnica Superior d'Enginyeria Informàtica, Universitat Politècnica de València (Spain)

{jjorge,amartinez1,agimenez,jsilvestre,ajuan,josanna}@dsic.upv.es
{golik,doetsch,ney}@cs.rwth-aachen.de
vicismar@inf.upv.es

Abstract

This paper describes the Automatic Speech Recognition systems built by the MLLP research group of Universitat Politècnica de València and the HLTPR research group of RWTH Aachen for the *IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge*. We participated in both the closed and the open training conditions.

The best system built for the closed condition was an hybrid BLSTM-HMM ASR system using one-pass decoding with a combination of a RNN LM and show-adapted n-gram LMs. It was trained on a set of reliable speech data extracted from the *train* and *dev1* sets using MLLP's transLectures-UPV toolkit (TLK) and TensorFlow. This system achieved 20.0% WER on the *dev2* set.

For the open condition we used approx. 3800 hours of out-of-domain training data from multiple sources and trained a one-pass hybrid BLSTM-HMM ASR system using open-source tools RASR and RETURNN developed at RWTH Aachen. This system scored 15.6% WER on the *dev2* set. The highlights of these systems include robust speech data filtering for acoustic model training and show-specific language modeling.

Index Terms: Automatic Speech Recognition, Acoustic Modeling, Language Modeling, Speech Data Filtering.

1. Introduction

This paper describes the joint collaboration between the *Machine Learning and Language Processing* (MLLP) research group from the *Universitat Politècnica de València* (UPV) and the *Human Language Technology and Pattern Recognition* (HLTPR) research group from the *RWTH Aachen University* for the participation in the IberSpeech-RTVE 2018 Speech-to-Text transcription challenge, that will be held during the IberSpeech 2018 conference in Barcelona, Spain. Our participation consisted of the submission of three systems: one primary and one contrastive for the closed system training condition, and one primary for the open condition.

The rest of the paper is structured as follows. First, Section 2 describes the RTVE database that was provided by the organizers of the challenge. Second, in Section 3 we describe the ASR systems we developed under the closed training conditions. Next, Section 4 details the ASR system that participated in the open training conditions. Finally, Section 5 provides a summary of the work and gives some concluding remarks.

2. RTVE database

The RTVE (*Radio Televisión Española*) database consists of a collection of 15 different TV shows broadcast by the Spanish public TV station between 2015 and 2018. It comprises 569 hours of audio data, from which 460 hours are provided with subtitles, and 109 hours have been human-revised. In addition, the database includes 3 million sentences extracted from the subtitles of 2017 broadcast news at the RTVE 24H Channel.

The database is provided in five partitions: *subs-C24H*, the 3M text dataset from the RTVE 24H channel; *train*, that comprises 463 hours of audio data with non-verbatim subtitles from 16 TV shows; and *dev1*, *dev2*, *test*, consisting of 57, 15 and 40 hours from 5, 2 and 8 different TV shows, respectively. *dev1* and *dev2* sets were provided with manual corrected transcripts, while the *test* set was used to gauge the performance of the participant systems. It is important to note that there is a certain overlap between *dev1*, *dev2*, *test* and the *train* set in terms of TV-shows.

In order to have available as many training data as possible during the development stage of the closed-condition system, we decided to split *dev1* into two subsets: *dev1-train*, comprising 43 hours of raw audio to be used for acoustic and language model training; and *dev1-dev*, 15 hours to be used internally as development data to optimize different components of the system. This split was done at the file level, trying to satisfy that *dev1-dev* have similar size to *dev2*, and that both *dev1* subsets include files from the same TV shows. We used *dev2* as a test set to measure the performance of the system on unseen data.

3. Closed-condition system

3.1. Speech data filtering

Under the closed training conditions, it is extremely important to make the most of the provided training data, specially when it is scarce and/or noisy. This is the case of the RTVE database: on the one hand, the *train* set comprises only 463 hours of audio, which is not much compared with the amount of data used to train current state-of-the-art systems [1, 2]. On the other hand, training data is not provided with verbatim transcripts but with approximate subtitles. This becomes a major concern when using recurrent neural networks for acoustic modeling, as the accuracy drops significantly when using noisy training data. Therefore, a robust speech data filtering procedure becomes a key point to achieve high ASR performance.

After examining some random samples from *train*, *dev1* and *dev2* sets, we first-hand checked that the provided subtitles are far from being verbatim transcripts, but also noted the

10.21437/IberSPEECH.2018-54

presence of 1) subtitle/transcription gaps, 2) subtitle files with no timestamps, 3) audio files considerably larger than their corresponding subtitle files, 4) subtitle files covering timestamps that exceed the length of their corresponding audio file, or 5) human transcription errors in *dev1*, among others.

For all these reasons, we applied the following speech data filtering pipeline. As subtitle timestamps 1) are not reliable in the *train* set, 2) are not given in *dev1*, and 3) are given only at speaker-turn level in *dev2*, we first force-aligned each audio file to its corresponding subtitle/transcript text. We did this using a preexisting hybrid CD-DNN-HMM ASR [3] system in which the search space was constrained to recognize the exact text (no language model was involved in this procedure), with the only freedom of exploring the different word pronunciations given by the lexicon model and of using an optional silence phoneme at the beginning of each word. In this way we computed the best alignment between the input frame sequence to the sequence of HMM states inferred from the subtitle/transcript text. Then, we applied a heuristic post-filtering based on state-level frame occupation and word-level alignment scores: if either an HMM state is aligned to more frames than the observed average state frame occupation + two times the observed standard deviation, or a word whose average alignment score is lower than a given threshold, then the corresponding word alignment is considered noisy and the word is removed. Next, we completely discarded those files in which more than two thirds of the words were filtered out in the previous step. Finally, we built a clean training corpus by joining words into segments whose boundaries were delimited by large-enough silences and deleted words.

Table 1: *Number of raw, aligned raw, aligned speech, and filtered speech hours as a result of applying the speech data filtering pipeline to the whole RTVE database.*

	Raw	Aligned		Filtered
		Raw	Speech	Speech
train	463	438	252	187
dev1-train	43	31	24	18
dev1-dev	14	12	9	7
dev2	15	12	9	6
Overall	535	493	294	218

Table 1 shows the result of applying this speech data filtering pipeline to the *train*, *dev1* and *dev2* sets. The second column shows the raw audio length in hours of each set. The third refers to the amount of raw hours that could be aligned to the corresponding subtitle/transcript text by our alignment system. It must be noted that there were some audio files that the system was not capable to align. This happens when none of the active hypotheses can reach the final HMM state at the last time step, due to an excessive histogram pruning or due to an non-matching transcript. For this reason, 42 hours of audio data could not be aligned. The fourth column gives the total amount of aligned speech data after removing non-speech events that were aligned to the silence phoneme. Surprisingly, the original 438 raw hours from the *train* set were reduced to 252 hours of speech data, i.e. we detected 186 hours of non-speech events. After some manual analysis of the alignments, we found that a significant portion of these 186 hours is explained by non-subtitled speech, whose corresponding audio frames were in practice aligned to the silence phoneme. Finally, the fifth column shows the number of hours of clean speech data after applying the described heuristic post-filtering procedure and after discarding files that shown a high word rejection rate. Starting with the original 535 hours of raw audio, we aligned 294 hours of speech, from which we rejected 76 hours of noisy data, ending up with 218 hours of speech suitable for acoustic training.

Table 2: *Corpus statistics of the text data used for LM training.*

	Sentences	Running words	Vocabulary
train	340K	4.3M	80K
subs-C24H	3.1M	57M	160K
RNN-train	1.8M	35M	176K
dev1-dev	9.9K	160K	13K
dev2	7.7K	150K	12K

3.2. Acoustic modeling

The acoustic models (AM) used during the development of the *MLLP-RWTH_c1-dev_closed* system were trained using filtered speech data from *train* and *dev1-train* sets, that is, 205 hours of training speech data. We extracted 16-dim. MFCC features augmented by the full first and second time derivatives, resuling in 48-dim. features.

Our acoustic models were based on the hybrid approach [4, 5]. We first trained a conventional context-dependent Gaussian mixture model hidden Markov model (CD-GMM-HMM) with three left-to-right states. The state-tying schema was estimated following a phonetic decision tree approach [6], resulting in 8.9K tied states. The GMM acoustic model was used to force align the training data. We then trained a context-dependent feed-forward DNN-HMM using a context window of 11 frames, six hidden layers with ReLU activation functions and 2048 units per layer. We used the transLectures-UPV toolkit (TLK) [7] to train both GMM and DNN acoustic models.

Apart from the feed-forward model, we also trained a BLSTM-HMM model [5]. The DNN was used to refine the alignment between input acoustic features and HMM states. We then trained the BLSTM-HMM model using the open source toolkit TensorFlow [8] and TLK. The BLSTM network consisted of four bidirectional hidden layers with 512 LSTM cells per layer and per direction.

In order to increase the amount of training data, the final submitted system (*MLLP-RWTH_p-final_closed*) was retrained on a total of about 218 hours from sets *train*, *dev1* and *dev2*.

3.3. Language modeling

Our language model (LM) for the closed condition consists of a combination of several n-gram models and a recurrent neural network (RNN) model. Also, since TV shows of each audio file are known in advance, we performed an LM adaptation at the n-gram model level.

First, we extracted sentences from all `.srt` and `.trn` files. Then we applied a common text processing pipeline to normalize capitalization, remove punctuation marks, expand contractions (i.e. *sr.* $\rightarrow$ *señor*) and transliterate numbers. As already mentioned, we split *dev1* into two subsets, *dev1-train* and *dev1-dev*, in order to include *dev1-train* in training. Thus, in this section, we will refer to the combination of *train* and *dev1-train* simply as *train*. For LSTM LM training, we concatenated the *train* and *subs-C24H* sets into a single training file and removed redundancy by discarding repeated sentences. Also, sentences were shuffled after each epoch to allow better generalization. To carry out TV-show LM adaptation experiments, we randomly extracted 500 sentences of each TV show from the *train* set to be used as validation data in the adaptation process. Table 2 provides corpus statistics after normalization.

Second, to define our closed-condition system's vocabulary, we first computed the vocabulary of both *train* and *subs-C24H* sets, and then removed singletons, so that language models can properly model unknown word probabilities. After applying these two steps, the resulting vocabulary had 132K words. The out-of-vocabulary ratios of *dev1-dev* and *dev2* sets were 0.36% and 0.53%, respectively.

Table 3: *Perplexities of the different LM components.*

	dev1-dev	*dev2*
(a) N-gram *train*	139.6	183.0
(b) N-gram *subs-C24H*	161.2	193.4
(c) N-gram show-specific	184.0	294.3
(d) N-gram general (a+b)	107.0	147.5
(e) N-gram adapt (a+b+c)	99.5	139.1
(f) RNN	92.3	110.7
(g) RNN+N-gram general (d+f)	78.2	101.8
(h) RNN+N-gram adapt (e+f)	68.9	99.2

Third, we trained two standard Kneser-Ney smoothed 4-gram LMs on the *train* and *subs-C24H* sets using the SRILM toolkit [9]. Rows (a) and (b) of Table 3 show the perplexities obtained with these models on the *dev1-dev* and *dev2* sets. In addition to these two general n-gram LMs, we trained one n-gram LM for each TV show. Row (c) of Table 3 shows the averaged perplexity of the corresponding TV-show-specific LM for each file.

Next, we trained a RNN LM using the Variance Regularization (VR) criterion [10]. This criterion reduces the computational cost during the test phase. Our models were trained on GPU devices using the CUED-RNNLM toolkit [11]. The network setup was optimized to minimize perplexity on the *dev1-dev* set. It consisted of a 1024-unit embedding layer and a hidden LSTM layer of 1024 units. The output layer is a 132K-unit softmax, whose size corresponds to the vocabulary size. The perplexities obtained with this network are depicted in Row (f) of Table 3.

Then, the combination of the LMs was done in two steps. Firstly, we performed a linear interpolation of n-gram models. For the general, non-adapted models, we interpolated the LMs estimated on the *train* and the *subs-C24H* sets by minimizing the perplexity on *dev1-dev* [12]. Row (d) of Table 3 shows the perplexities for this particular LM combination. For each show-specific LM, we performed a three-way interpolation: the individual show-specific LM, the *train* LM and the *subs-C24H* LM. In this case, interpolation weights were optimized individually for each TV show so that the perplexity was minimized on the corresponding 500-sentence show-specific validation set, similarly to the approach followed in [13, 14]. Secondly, we combined the interpolated n-gram LMs with the RNN LM. Other than the static interpolation of n-gram LMs, the result of this step is not a new monolithic model, but a set of interpolation weights to be used on-the-go by the ASR decoder during search. Perplexities for the combination of the RNN LM with the general and the adapted n-gram LMs can be found in Rows (g) and (h) of Table 3.

Finally, to take the most of the provided data, the final submitted system (*MLLP-RWTH_p-final_closed*) was trained using the same hyper-parameters values estimated during the development stage, but using also *dev1-dev* and *dev2* sets as part of the training data.

3.4. Experiments and results

In this section we describe the experiments carried out to determine the best closed training condition system. Our experiments were devoted to assess three components of the system: acoustic models, language models and voice activity detection (VAD) modules. In all cases we used the TLK toolkit decoder [7] for recognizing test data using a one-pass decoding setup.

First, we compared the performance of the CD-FFDNN-HMMs and BLSTM-HMMs acoustic models described in Section 3.2. In both cases we used the general n-gram language model described in Section 3.3. Grammar scale factor and search pruning parameters were optimized on the *dev1-dev* set. Table 4 shows the results on both *dev1-dev* and *dev2* sets. As expected, the BLSTM acoustic model outperformed the feed-forward model by 12.2% relative.

Table 4: *Comparison of the CD-FFDNN-HMM and BLSTM-HMM acoustic models using the general n-gram language model. Results in WER % and relative WER % improvement.*

	dev1-dev	*dev2*	
	WER	WER	ΔWER
FFDNN	29.7	27.1	-
BLSTM	26.5	23.8	12.2

Next, we analyzed the contribution of different LM combinations during search, leaving fixed the acoustic model to the best BLSTM neural network. Specifically, we carried out recognition experiments using (1) the general n-gram LM, (2) the RNN LM, (3) the interpolation of the RNN LM with the general n-gram LM, and (4) the interpolation of the RNN LM with the adapted, show-specific n-gram LMs. Table 5 shows perplexities and WERs for the *dev1-dev* and *dev2* sets over these four different LM setups.

Table 5: *Comparison of different language model combinations using the BLSTM-HMM acoustic model in terms of perplexity, WER % and relative WER % improvement.*

	dev1-dev		*dev2*		
	PPL	WER	PPL	WER	ΔWER
n-gram general	107	26.5	148	23.8	-
RNN	92	26.2	111	23.0	3.4
RNN + n-gram general	78	25.3	102	22.4	5.9
RNN + n-gram adapt	69	24.8	99	22.4	5.9

The best results were obtained with the combination of RNN and n-gram models, showing a consistent 6% relative improvement in both sets over the baseline general n-gram LM. It is worth noting that in terms of WER, the improvement from using adapted models does not translate to *dev2*. As *dev1-dev* and *dev2* contain different shows with strongly varying amounts of show-specific text data available for training, not all shows benefit from adaptation equally. Anyway, since the adaption does not degrade the system performance, and given the good improvement seen on *dev1-dev*, we decided to use the combination of RNN LMs plus adapted n-gram LMs for the final system.

Looking at the system outputs, after carrying out error analysis, we realized that our VAD module [15] was discarding a significant amount of speech regions in the audio files. This significantly affected the WER by increasing the number of deletions. For this reason, we decided to explore other audio segmentation approaches and compare its performance in terms of WER. Concretely, we compared the following approaches: (1) our baseline MLLP-UPV VAD system, based on a speech/non-speech GMM-HMM classifier that ranked second in the Albayzin-2012 audio segmentation challenge [15]; (2) The LIUM Speaker Diarization Tools, a VAD system based on Generalized Likelihood Ratio between speech/non-speech Gaussian models [16]; (3) The well-known CMUseg audio segmentation system using the standard configuration [17]; (4) Apply a fast pre-recognition step to segment the audio file by the recognized silences, using the best CD-FFDNN-HMM acoustic model and a pruned version of the general n-gram LM; and (5) Use the segments generated in (4), and apply VAD the system (1) to classify those segments into speech/non-speech. It is important to note that (3) and (4) are not VAD systems but just audio segmenters, so all detected segments are considered speech,

i.e. all audio is passed through to the ASR. Table 6 shows the WER for each of the five audio segmentation/VAD techniques, including the ratio of discarded audio that is dropped by the VAD prior to decoding.

Table 6: *Comparison of different audio segmentation/VAD techniques using the BLSTM-HMM acoustic model and the combination of the RNN LM + adapted n-gram LM. Results in WER % and relative WER % improvement and the ratio of dropped audio.*

	dev1-dev		*dev2*		
	% drop.	WER	% drop.	WER	ΔWER
MLLP-UPV (1)	10.9	24.8	5.9	22.4	-
LIUM (2)	7.1	23.7	3.9	20.8	7.1
CMUseg (3)	0	23.2	0	20.9	6.7
Pre-Recognition (4)	0	22.9	0	20.6	8.0
+ MLLP-UPV (5)	3.2	22.3	3.3	20.0	10.7

As we expected, the baseline VAD system (1) was discarding too much segments, as it was too aggressive compared to other techniques. With either (2) or (3) we obtained a consistent improvement. It was further increased up to 8% by using (4). We decided then to combine this segmentation with our baseline VAD system (1), which led us to achieve an 11% relative WER improvement. In absolute terms, we got a 2.4 WER points gain in *dev2*, with a final WER of 20.0%. This setup constituted our contrastive closed-condition system (*MLLP-RWTH_c1-dev_closed*), whilst our primary system (*MLLP-RWTH_p-final_closed*) was the result of re-training the same acoustic and language models with all available data, as stated in Sections 3.2 and 3.3.

Finally we analyzed the speed of submitted system in terms of Real Time Factor (RTF). We studied how tightening the pruning parameters affects the RTF and the WER. Also, to assess the speed of a fast pre-recognition step to segment the audio signal, we also did this comparison using the LIUM VAD system. Results of this analysis are shown in Table 7. First, a more aggressive pruning speeds up the submitted system by 88% while degrading the WER by 0.3% absolute. Next, if we replace the pre-recognition step on the submitted system by the LIUM VAD module, we get a speed-up of 50% at the cost of 0.9 points WER. Finally, we could afford a very significant speed-up of 375% if we tighten the prune parameters when using LIUM VAD, with a WER loss of 1.3 absolute points, although it would still be a competitive system, scoring 21.3% WER points on *dev2*.

Table 7: *Speed analysis in terms of RTF an its effect on the WER% over the dev2 set, either with the submitted system and removing the pre-precognition step, using LIUM VAD instead.*

	RTF	WER
Submitted system (1)	1.5	20.0
+ inc. prune	0.8	20.3
(1) with LIUM VAD	1.0	20.9
+ inc. prune	0.4	21.3

4. Open-condition system

The main motivation for participating in the open-condition track was the desire to evaluate a system developed in the recent months for a different purpose, not related to the IberSpeech challenge. In order to achieve this goal, we decided to keep the amount of parameter optimization as low as possible. This system is based on the software developed at RWTH Aachen University: RASR [18, 19] and RETURNN [20, 21].

The ASR system is based on a hybrid LSTM-HMM acoustic model. It was trained on a total of approx. 3800 hours of transcribed speech from several sources, covering a variety of domains and acoustic conditions. The collection consists of subtitled videos crawled from Spanish and Latin American websites.

We used a pronunciation lexicon with a vocabulary size of 325k with one or more pronunciation variants. The acoustic model takes 80-dim. MFCC features as input and estimates state posterior probabilities for 5000 tied triphone states. The state tying was obtained by estimating a classification and regression tree (CART) on all available training data. Acoustic modeling was done using a bi-directional LSTM network with four layers and 512 LSTM units in each layer. About 30% of activations are dropped in each layer for regularization purpose [22]. During training we minimized the cross-entropy of a network generated distribution in the softmax output layer at aligned label positions using a Viterbi alignment defined over the 5000 tied triphone states of the CART. We used the Adam learning rate schedule [23] with integrated Nesterov momentum and further reduced the learning rate following a variant of the Newbob scheme. We split input utterances into overlapping chunks of roughly 10 seconds and perform an L2 normalization of the gradients for each chunk. With the normalized gradients the network is updated in a stochastic gradient descent manner where batches containing up to 50 chunks are distributed over eight GPU devices and recombined into a common network after roughly 500 chunks have been processed by all devices.

The language model for the single-pass HMM decoding is a 5-gram count model trained with Kneser-Ney smoothing on a large body of text data collected from multiple publicly available sources. Its perplexity on *dev1-dev* and *dev2* is 173.5 and 173.2 respectively. This open-track system has reached a WER of 18.3% and 15.6% on *dev1-dev* and *dev2* without any speaker or domain adaptation or model tuning.

5. Conclusions

In this paper we have presented the description of the three systems that participated in the IberSpeech-RTVE 2018 Speech-to-Text transcription challenge. Two of them, one primary (*MLLP-RWTH_p-final_closed*) and one contrastive (*MLLP-RWTH_c1-dev_closed*), were submitted to the closed training conditions, while the other one (*MLLP-RWTH_p-prod_open*) participated in the open training track. On the one hand, our best development closed-condition ASR system (*MLLP-RWTH_c1-dev_closed*), consisting of a BLSTM-HMM acoustic model trained on a reliable set of 205 hours of training speech data, and a combination of both RNN and TV-show adapted n-gram language models, achieved a competitive mark of 20.0% WER on the *dev2* set. Our final, primary closed-condition ASR system (*MLLP-RWTH_p-final_closed*) should offer a similar or even better performance as it followed the same system design setup but trained with all available data, including both development sets. On the other hand, our general-purpose open-condition ASR system (*MLLP-RWTH_p-prod_open*), without carrying out any speaker, domain nor model adaptation of any kind, scored 15.6% WER on the *dev2* set.

6. Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761758 (X5gon) and the Spanish government's TIN2015-68326-R (MINECO/FEDER) research project MORE. This work also financed by grant FPU14/03981 from the Spanish Ministry of Education, Culture and Sport. Finally, we would also like to thank our colleagues at RWTH Aachen for many fruitful discussions: Eugen Beck, Tobias Menne and Albert Zeyer.

7. References

[1] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, Canada, Apr. 2018, pp. 4774–4778.

[2] W. Xiong, L. Wu, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, Canada, Apr. 2018, pp. 5934–5938.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "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, Nov. 2012.

[4] H. Bourlard and C. J. Wellekens, "Links between Markov models and multilayer perceptrons," in *Advances in Neural Information Processing Systems I*, D. Touretzky, Ed. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 502–510.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[6] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. Workshop on Human Language Technology*, Plainsboro, NJ, USA, Mar. 1994, pp. 307–312.

[7] M. del Agua, A. Giménez, N. Serrano, J. Andrés-Ferrer, J. Civera, A. Sanchis, and A. Juan, "The translectures-UPV toolkit," in *Advances in Speech and Language Technologies for Iberian Languages*, Nov. 2014, pp. 269–278.

[8] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[9] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. of the Int. Conf. on Spoken Language Processing*, Denver, CO, USA, Sep. 2002, pp. 901–904.

[10] X. Chen, X. Liu, M. J. F. Gales, and P. C. Woodland, "Improving the training and evaluation efficiency of recurrent neural network language models," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Brisbane, Australia, Apr. 2015, pp. 5401–5405.

[11] X. Chen, X. Liu, Y. Qian, M. J. F. Gales, and P. C. Woodland, "CUED-RNNLM – An open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Mar. 2016, pp. 6000–6004.

[12] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proc. Workshop on Pattern Recognition in Practice*, Amsterdam, Netherlands, Apr. 1980, pp. 381–397.

[13] A. Martínez-Villaronga, M. A. del Agua, J. Andrés-Ferrer, and A. Juan, "Language model adaptation for video lectures transcription," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 2013, pp. 8450–8454.

[14] A. Martínez-Villaronga, M. A. del Agua, J. A. Silvestre-Cerdà, J. Andrés-Ferrer, and A. Juan, "Language model adaptation for lecture transcription by document retrieval," in *Proc. IberSpeech*, Nov. 2014.

[15] J. A. Silvestre-Cerdà, A. Giménez, J. Andrés-Ferrer, J. Civera, and A. Juan, "Albayzin Evaluation: The PRHLT-UPV Audio Segmentation System," in *Proc. IberSpeech*, Madrid, Spain, Nov. 2012, pp. 596–600.

[16] S. Meignier and T. Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *Proc. CMU SPUD Workshop*, Dallas, TX, USA, Mar. 2010.

[17] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA Speech Recognition Workshop*, 1997, pp. 97–99.

[18] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "RASR - the RWTH Aachen University open source speech recognition toolkit," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Honolulu, HI, USA, Dec. 2011.

[19] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 3313–3317.

[20] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: The RWTH extensible training framework for universal recurrent neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New Orleans, LA, USA, Mar. 2017, pp. 5345–5349.

[21] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Annual Meeting of the Assoc. for Computational Linguistics*, Melbourne, Australia, Jul. 2018.

[22] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the Int. Conf. on Machine Learning*, San Diego, CA, USA, May 2015.