IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

MLLP-VRAIN Spanish ASR Systems for the Albayzin-RTVE 2020 Speech-To-Text Challenge

*Javier Jorge, Adrià Giménez, Pau Baquero-Arnal, Javier Iranzo-Sánchez,
Alejandro Pérez, Gonçal V. Garcés Díaz-Munío, Joan Albert Silvestre-Cerdà,
Jorge Civera, Albert Sanchis and Alfons Juan*

Machine Learning and Language Processing (MLLP) research group
Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politècnica de València
Camí de Vera s/n, 46022, València, Spain

{jajorca,adgipas,pabaar,jairsan,alpegon2,gogardia,
juasilce,jorcisai,josanna2,ajuanci}@vrain.upv.es

Abstract

This paper describes the automatic speech recognition (ASR) systems built by the MLLP-VRAIN research group of Universitat Politècnica de València for the Albayzin-RTVE 2020 Speech-to-Text Challenge.

The primary system (*p-streaming_1500ms_nlt*) was a hybrid BLSTM-HMM ASR system using streaming one-pass decoding with a context window of 1.5 seconds and a linear combination of an n-gram, a LSTM, and a Transformer language model (LM). The acoustic model was trained on nearly 4,000 hours of speech data from different sources, using the MLLP's transLectures-UPV toolkit (TLK) and TensorFlow; whilst LMs were trained using SRILM (n-gram), CUED-RNNLM (LSTM), and Fairseq (Transformer), with up to 102G tokens. This system achieved 11.6% and 16.0% WER on the *test-2018* and *test-2020* sets, respectively. As it is streaming-enabled, it could be put into production environments for automatic captioning of live media streams, with a theoretical delay of 1.5 seconds.

Along with the primary system, we also submitted three contrastive systems. From these, we highlight the system *c2-streaming_600ms_t* that, following the same configuration of the primary one, but using a smaller context window of 0.6 seconds and a Transformer LM, scored 12.3% and 16.9% WER points respectively on the same test sets, with a measured empirical latency of 0.81±0.09 seconds (mean±stdev). This is, we obtained state-of-the-art latencies for high-quality automatic live captioning with a small WER degradation of 6% relative.

Index Terms: natural language processing, automatic speech recognition, streaming.

1. Introduction

This paper describes the participation of the *Machine Learning and Language Processing* (MLLP) research group from the *Valencian Research Institute for Artificial Intelligence* (VRAIN), hosted at the *Universitat Politècnica de València* (UPV), in the Albayzin-RTVE 2020 Speech-to-Text (S2T) Challenge.

Live audio and video streams such as TV broadcasts, conferences, lectures, as well as general-public video streaming services (e.g. Youtube) over the Internet have increased dramatically in recent years because of the advances in networking with high speed connections and proper bandwidth. Also, due to the COVID-19 pandemic, video meeting/conferencing platforms have experienced an exponential growth of usage, as public and private companies have leveraged teleworking for their employees to comply with the social distancing measures recommended by health authorities.

Automatic transcription and translation of such audio streams is a key feature in a globalized and interconnected world, in order to reach wider audiences or to ensure proper understanding between native and non-native speakers, depending on the use-case. Also, public governments are enforcing TV broadcasters by law to provide accessibility options to people with hearing disabilities, with a yearly increasing amount of contents to be captioned at a minimum [1, 2].

Some TV broadcasters and other live streaming services have assumed manual transcription from scratch of live audio or video streams, as an initial solution to comply with the current legislation, and/or to satisfy user expectations. However, it is a really hard task for professional linguists that, under very stressful conditions, are very prone to generate captioning errors. Besides, it is difficult to scale up such a service, as in these organizations, the amount of human resources devoted to this particular task is typically scarce.

Due to these reasons, the need and demand for high-quality real-time streaming Automatic Speech Recognition (ASR) has increased drastically in the last years. Automatic live audio stream subtitling enables professional linguists to correct live transcripts provided by these ASR systems, if they are not publishable as they come. This would dramatically expedite their productivity and significantly reduce the probability of producing transcription errors. However, the application of state-of-the-art ASR technology to video streaming is a highly complex and challenging task due to real-time and low-latency recognition constraints.

The MLLP-VRAIN, being aware of these demands from the society, have focused its research efforts in the past two years on streaming ASR. This work aims to disseminate our latest developments in this area, showing how our hybrid ASR technology can be successfully applied under streaming conditions, by providing high-quality transcriptions and state-of-the-art system latencies on real-life tasks such as the RTVE (*Radio Televisión Española*) database. Therefore, our participation in the Albayzin-RTVE 2020 S2T Challenge consisted on the submission of a primary, performance-focused streaming ASR system, plus three contrastive systems: two latency-focused streaming ASR systems, and one conventional off-line ASR system.

10.21437/IberSPEECH.2021-25

Table 1: *Transcribed Spanish speech resources for AM training.*

Resource	Duration (h)
Internal: entertainment	2932
Internal: educational	406
Internal: user-generated content	202
Internal: parliamentary data	158
Voxforge [8]	21
RTVE2018: *train*	187
RTVE2018: *dev1-train*	18
TOTAL	3924

The rest of the paper is structured as follows. First, Section 2 briefly describes the Albayzin-RTVE 2020 S2T Challenge and the RTVE databases provided by the organizers. Next, Section 3 provides a detailed description of our participant ASR systems. Finally, Section 4 gives a summary of the work plus some concluding remarks.

2. Challenge description and databases

The Albayzin-RTVE 2020 Speech-To-Text Challenge consists of automatically transcribing different types of TV shows from the RTVE Spanish public TV station, and the assessment of ASR system performance in terms of Word Error Rate (WER) by comparing those automatic transcriptions with correct reference transcriptions [3].

The MLLP-VRAIN participated in the 2018 edition of the challenge [4] in a joint collaboration with the *Human Language Technology and Pattern Recognition* (HLTPR) research group from the *RWTH Aachen University*. The evaluation was carried out on the RTVE2018 database [5], that includes 575 hours of audio from 15 different TV shows broadcasted between 2015 and 2018. This database is allocated into four sets: *train, dev1, dev2* and *test* (*test-2018*). Our systems won in both the open-condition and closed-condition tracks [6], scoring 16.5% and 22.0% WER points respectively in the *test-2018* set.

For the 2020 edition of the challenge, the participation has been limited to a single open-condition track, and system evaluations have been carried out over the *test* (*test-2020*) set from the RTVE2020 database, which includes 78.4 hours from 15 different TV shows broadcasted between 2018 and 2019 [7].

3. MLLP-VRAIN Systems

In this section we describe the hybrid ASR systems developed by the MLLP-VRAIN that participated in the Albayzin-RTVE 2020 S2T Challenge.

3.1. Acoustic Modelling

Our acoustic models (AM) were trained using 205 filtered speech hours from the *train* set (187h) and our internal *dev1-train* set (18h), as in [4], plus about 3.7K hours of other resources crawled from the Internet. Table 1 summarises all training datasets along with their total duration (in hours). From this data, first, we extracted 16-dimensional MFCC features plus first and second derivatives (48-dimensional feature vectors) every 10ms to train a context-dependent feed-forward DNN-HMM with three left-to-right tied states using the transLectures-UPV toolkit (TLK) [9]. The state-tying schema followed a phonetic decision tree approach [10] that produced 10K tied states. Then, feed-forward models were used to bootstrap a BLSTM-HMM AM, trained with 85-dimensional filterbank features, following the procedure described in [11]. The BLSTM network was trained using both TLK and TensorFlow [12], and had 8 bidirectional hidden layers with 512 LSTM cells per layer and direction. As in [11], we performed chunking during training by considering a context to perform back-propagation through time to a window size of 50 frames. Additionally, SpecAugmentation was applied by means of time and frequency distortions [13].

3.2. Language Modelling

Regarding language modelling, we trained count-based (n-gram) and neural-based (LSTM, Transformer) Language Models (LMs) to perform one-pass decoding with different linear combinations of them [14], using the text data sources and corpora described in Table 2.

On the one hand, we trained 4-gram LMs using SRILM [15] with all text resources plus the Google-counts v2 corpus [16], accounting for 102G running words. The vocabulary size was limited to 254K words, with an OOV ratio of 0.6% computed over our internal development set.

On the other hand, regarding neural LMs, we considered the LSTM and Transformer architectures. In both cases, LMs were trained using a 1-gigaword subset randomly extracted from all available text resources, except Google-counts. Their vocabulary was defined as the intersection between the n-gram vocabulary (254K words) and that derived from the aforementioned training subset. We did this to avoid having zero probabilities for words that are present in the system vocabulary but not in the training subset. This is taken into account when computing perplexities by renormalizing the unknown-word score accordingly.

Specific training details for each neural LM architecture are as follows. Firstly, LSTM LMs were trained using the CUED-RNNLM toolkit [17]. Noise Contrastive Estimation (NCE) criterion [18] was used to speed up model training, and the normalization constant learned from training was used during decoding [19]. Based on the lowest perplexity observed on our internal development set, we selected as final model that with a 256-unit embedding layer and two hidden LSTM layers of 2048 units. Secondly, Transformer LMs (TLMs) were trained using a customized version of the FairSeq toolkit [20], selecting the following configuration that minimized perplexity in our internal development set: 24-layer network with 768 units per layer, 4096-unit FFN, 12 attention heads, and an embedding of 768 dimensions. These models were trained until convergence with batches limited to 512 tokens, 512 sentences, and 512 words per sentence. Parameters were updated every 32 batches. During inference, Variance Regularization (VR) was applied to speed up the computation of the TLM score [21].

3.3. Decoding strategy

Our hybrid ASR systems follow a real-time one-pass decoding by means of a History Conditioned Search (HCS) strategy, as described in [14]. This approach allows us to benefit from the direct usage of additional LMs during decoding while satisfying real-time constraints. This decoding strategy introduces two additional and relevant parameters to control the trade-off between Real Time Factor (RTF) and WER: LM history recombination (LMHR), and LM histogram prunning (LMHP). The static look-ahead table, needed by the decoder to use pre-computed look-ahead LM scores, was generated from a prunned

Table 2: *Statistics of Spanish text resources for LM training. S=Sentences, RW=Running words, V=Vocabulary. Units are in thousands (K).*

Corpus	S(K)	RW(K)	V(K)
Opensubtitles [22]	212635	1146861	1576
UFAL [23]	92873	910728	2179
Wikipedia [24]	32686	586068	3373
UN [25]	11196	343594	381
News Crawl [26]	7532	198545	648
Internal: entertainment	4799	59235	307
eldiario.es [27]	1665	47542	247
El Periódico [28]	2677	46637	291
Common Crawl [29]	1719	41792	486
Internal: parliamentary data	1361	35170	126
News Commentary [26]	207	5448	83
Internal: educational	87	1526	35
TOTAL	369434	3423146	5785
Google-counts v2 [16]	-	97447282	3693

Table 3: *Basic statistics of development and tests sets of RTVE databases, including our internal dev1-dev set: total duration (in hours), number of files, average duration of samples in seconds plus-minus standard deviation ($d_\mu \pm \sigma$), and running words (RW) in thousands (K).*

Set	Duration(h)	Files	d_μ	$\pm\ \sigma$	RW(K)
dev1-dev	11.9	10	4267	± 1549	120
dev2	15.2	12	4564	± 1557	149
test-2018	39.3	59	2395	± 1673	377
test-2020	78.4	87	2314	± 1576	519

version of the n-gram LM.

For streaming ASR, as the full sequence (context) is not available during decoding, BLSTM AMs are queried with a sliding, overlapping context window of limited size over the input sequence, averaging outputs of all windows for each frame to obtain the corresponding acoustic score [30]. The size of the context window (in frames or seconds) is set in decoding, and defines the theoretical latency of the system. This limitation of the context prevents us to perform a Full Sequence Normalization (FSN), that is typically applied under the off-line setting. Instead, we applied the Weighted Moving Average (WMA) technique, that uses the content of the current context window to update normalization statistics on-the-fly, weighted by previous context from past windows with an α parameter [31]. Finally, as Transformer LMs have the inherent capacity of attending to potentially infinite word sequences, history is limited to a given maximum number of words, in order to meet the strict computational time constraints imposed by the streaming scenario [21]. By applying all these modifications, our decoder acquires the capacity to deliver live transcriptions for incoming audio streams of potentially infinite length, with latencies lower-bounded by the context window size.

3.4. Experiments and results

To carry out our experiments, we used the development and test sets from the RTVE2018 database. More precisely, we devoted our internal *dev1-dev* set [4] for development purposes, whilst *dev2* and *test-2018* were dedicated to test ASR performance. Finally, *test-2020* was the blind test used by the organisation to rank the participant systems. Table 3 provides basic statistics of these sets.

Table 4: *Perplexity (PPL) and interpolation weights, computed over the dev1-dev set, of all possible linear combinations of n-gram (ng), LSTM (ls) and Transformer (tf) LMs.*

LM comb.	PPL	Weights(%)
ng	179.5	-
ls	98.4	-
tf	63.3	-
ng + ls	93.2	15 + 85
ng + tf	61.6	6 + 94
ls + tf	60.7	13 + 87
ng + ls + tf	59.5	5 + 10 + 85

First, we studied the perplexity (PPL) on the *dev1-dev* set of all possible linear combinations for the three types of LMs considered in this work. Table 4 shows the PPLs of these interpolations, along with the optimum LM weights that minimized PPL in the *dev1-dev* set. The Transformer LM provides significant lower perplexities in all cases, and accordingly, takes very high weight values when combined with other LMs. Indeed, the TLM in isolation already delivers a strong perplexity baseline value of 63.3, while the maximum PPL improvement is of just 6% relative when all three LMs are combined.

Second, we tuned decoding parameters to provide a good WER-RTF tradeoff on *dev1-dev*, with the hard constraint of RTF<1 to ensure a real-time processing of the input. From these hiperparameters, we highlight, due to their relevance, LMHR=12, LMHP=20, and TLM history limited to 40 words.

At this point, we defined our participant off-line hybrid ASR system identified as *c3-offline* (contrastive system no. 3), consisting of a fast pre-recognition + Voice Activity Detection (VAD) step to detect speech/no-speech segments as in [4], followed by a real-time one-pass decoding with our BLSTM-HMM AM, using a FSN normalization scheme and a linear combination of the three types of LMs: n-gram, LSTM and Transformer. This system scored 12.3 and 17.1 WER points on test-2018 and test-2020, respectively.

Next, as our focus was to develop the best-performing streaming-capable hybrid ASR system for this competition, we explored streaming-related decoding parameters to optimize WER on *dev1-dev*, using the BLSTM-HMM AM and a linear combination of all three LMs. This resulted on using a context window size of 1.5 seconds and α=0.95 for the WMA normalization technique. This configuration defined our primary system, identified as *p-streaming_1500ms_nlt*, that showed WER rates of 11.6 and 16.0 in *test-2018* and *test-2020*, respectively. It is important to note that this system does not integrate any VAD module. This task is implicitly carried out by the decoder via the non-speech model of the BLSTM-HMM AM.

A small change on the configuration of the primary system, consisting on the removal of the LSTM LM from the linear interpolation, defined the contrastive system no. 1, identified as *c1-streaming_1500ms_nt*. The motivation behind this change is that the computation of LSTM LM scores is quite expensive in computational terms, and its contribution to PPL is negligible with respect to the n-gram LM + TLM combination (3% relative improvement). Hence, for the sake of system latency stability, we obtained nearly no degradation in terms of WER: 11.6 and 16.1 points in *test-2018* and *test-2020*, respectively.

Both streaming ASR systems, *p-streaming_1500ms_nlt* and *c1-streaming_1500ms_nt*, share the same theoretical latency of 1.5 seconds, as it is determined by the context window size. As

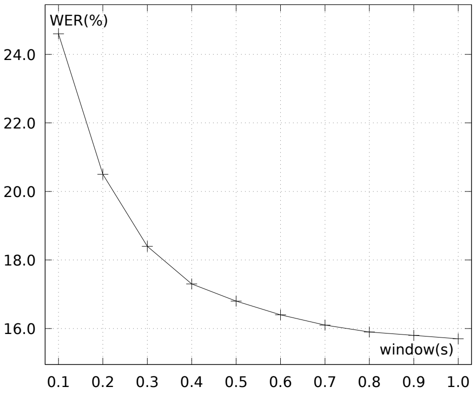

Figure 1: *WER as a function of context window size (in seconds) for the streaming setup, computed over the dev1-dev set.*

stated in Section 3.3, this parameter can be adjusted in decoding time. This allows us to configure the decoder for lower latency responses or better transcription quality. Hence, our last commitment for this challenge was to find a proper system configuration that could provide state-of-the-art, stable latencies with minimal WER degradation. Figure 1 illustrates the evolution of WER on *dev1-dev* as a function of the context window size, limited to one second at maximum. As we focused on gauging AM performance, we used the n-gram LM in isolation for efficiency reasons. At the light of the results, we chose a window size of 0.6 seconds, as it brings a good balance between transcription quality and theoretical latency.

The last step to set up our latency-focused streaming system was to measure WER and empirical latencies as a function of different prunning parameters and LM combinations. In our experiments, latency is measured as the time elapsed between the instant at which an acoustic frame is generated, and the instant at it is fully processed by the decoder. We provide latency figures at the dataset level, computed as the average of the latencies observed at the frame level on the whole dataset. Figure 2 shows WER vs mean empirical latency figures, computed over *dev1-dev*, with different prunning parameter values, and comparing the LM combinations that include the Transformer LM. These measurements were run on an Intel i7-3820 CPU @ 3.60GHz, with 64GB of RAM and a RTX 2080 Ti GPU card. On the one hand, we can see how combinations involving LSTM LMs are systematically shifted rightwards w.r.t. other combinations. This means that the LSTM LM has a clear negative impact on system latency, with little to no effect on system quality. This evidence corroborates our decision of removing the LSTM LM to define our contrastive system *c1-streaming_1500ms_nt*. On the other hand, TLM alone generally provides a good baseline that is slightly improved in terms of WER if we include the other LMs. However, this comes with the cost of increasing latency. Hence, we selected the Transformer LM in isolation for our final latency-focused streaming system. This system was our contrastive system no. 2, identified as *c2-streaming_600ms_t*. Its empirical latency on *dev1-dev* was 0.81±0.09 seconds (mean±stdev), and its performance was 12.3 and 16.9 WER points in *test-2018* and *test-2020*, respectively. This is, with just a very small relative WER degradation of 6% w.r.t. the primary system, we got state-of-the-art (mean=0.81s) and very stable (stdev=0.09s) empirical latencies. This system has a baseline consumption (when idle) of 9GB RAM and 3.5GB GPU memory (on a single GPU card), adding 256MB RAM and one CPU thread per each decoding (audio stream). For instance, the decoding of four simultaneous audio streams in a single machine would use four CPU threads, 10GB RAM and 3.5GB GPU memory.

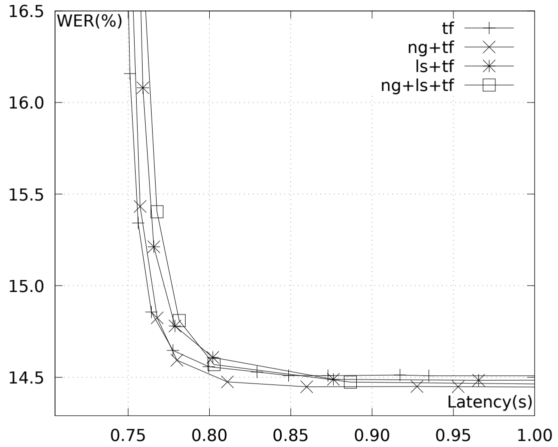

Figure 2: *WER versus mean empirical latency (in seconds) on dev1-dev, measured with different prunning parameters, and considering only interpolation schemes that include TLM.*

Table 5: *WER of the participant systems, including our open-condition system submitted to the 2018 challenge, computed over the dev2, test-2018 and test-2020 sets.*

System	*dev2*	*test-2018*	*test-2020*
p-streaming_1500ms_nlt	11.2	11.6	16.0
c1-streaming_1500ms_nt	-	11.6	16.1
c2-streaming_600ms_t	12.0	12.3	16.9
c3-offline	-	12.0	17.1
2018 open-cond. winner [4]	15.6	16.5	-

Table 5 summarises the results obtained with all the four participant ASR systems in the *dev2*, *test-2018* and *test-2020* sets, and adds the results obtained with our 2018 open-condition system for comparison. On the one hand, surprisingly, the offline system is surpassed by the three streaming ones in *test-2020*, by up to 1.1 absolute WER points (6% relative). We believe that this is caused, first, by an improvable VAD module, based on Gaussian Mixture HMMs, that, in our experience, suffers from false negatives (speech segments labelled as non-speech). As the non-speech model was trained with music and noise audio segments, and given the inherent limitations of GMMs, it is likely to misclassify speech passages with loud background music and noise (often present in TV programmes) as non-speech. Second, the FSN technique might not be appropriate for some types of TV shows, as local acoustic condition changes become diluted in the full-sequence normalization, and acoustic scores computed for those frames may present some perturbations that can degrade system performance at that point. On the other hand, it is remarkable that our primary 2020 system significantly outperforms the 2018 winning system by 28% relative WER points on both *dev2* and *test-2018* (25% in the case of our latency-focused system *c2-streaming_600ms_t*), while adding the novel streaming capability at the same time.

All these streaming ASR systems can be easily put into production environments using our custom gRPC-based serverclient infrastructure[1]. Indeed, ASR systems comparable to *c2-streaming_600ms_t* and *c1-streaming_1500ms_nt* are already in production at our Transcription and Translation Plarform (TTP)[2] for streaming and off-line processing, respectively. Both can be freely tested using our public APIs, accessible via TTP.

4. Conclusions

In this paper we have described our four ASR systems that participated in the Albayzin-RTVE 2020 Speech-to-Text Challenge. The primary one, a streaming-enabled performance-focused hybrid ASR system (*p-streaming_1500ms_nlt*) provided a good score of 16.0 WER points in the *test-2020* set, and a remarkable 28% relative WER improvement over the 2018 winning ASR system on *test-2018*, with a theoretical latency of 1.5 seconds. Nearly the same performance was delivered by our first contrastive system (*c1-streaming_1500ms_nt*): 16.1 WER points on *test-2020*, at a significant lower computational cost. In pursuit of low, state-of-the-art system latencies, our second contrastive system (*c2-streaming_600ms_t*) provided a groundbreaking WER-latency balance, with a solid performance of 16.9 WER points on *test-2020* at an empirical latency of 0.81 ± 0.09 seconds (mean±stdev). Finally, our contrastive off-line ASR system with VAD (*c3-offline*) provides the highest, yet still competitive, WER mark of 17.1 points, attributable to an improvable VAD module and to the limitations of FSN when dealing with local acoustic condition changes.

With a configurable system latency in decoding time, our ASR technology offers the flexibility to produce fast system responses for streaming applications, or to generate maximum quality transcriptions whenever hard time constraints do not apply. Also, results demonstrate that our streaming ASR technology is mature enough to be systematically put into production environments for high-quality automatic live captioning in TV stations, distance learning, conferencing platforms, or general-purpose video/audio streaming services, among others.

5. Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 761758 (X5Gon); the Government of Spain's research project Multisub (ref. RTI2018-094879-B-I00, MCIU/AEI/FEDER,EU) and FPU scholarships FPU14/03981 and FPU18/04135; and the Generalitat Valenciana's research project Classroom Activity Recognition (ref. PROMETEO/2019/111) and predoctoral research scholarship ACIF/2017/055.

6. References

[1] "RD 1494/2007, del 12 de noviembre," 2007. [Online]. Available: https://www.boe.es/buscar/act.php?id=BOE-A-2007-19968

[2] "Llei 1/2006, de 19 d'abril, GVA," 2006. [Online]. Available: https://www.dogv.gva.es/va/eli/es-vc/l/2006/04/19/1/dof/vci-spa/pdf

[3] E. Lleida *et al.*, "IberSPEECH-RTVE 2020 speech to text transcription challenge," 2020. [Online]. Available: http://catedrartve.unizar.es/reto2020/EvalPlan-S2T-2020-v1.pdf

[4] J. Jorge *et al.*, "MLLP-UPV and RWTH Aachen Spanish ASR Systems for the IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge," in *Proc. IberSPEECH 2018*, 2018, pp. 257–261.

[5] E. Lleida *et al.*, "RTVE2018 database description," 2018. [Online]. Available: http://catedrartve.unizar.es/reto2018/RTVE2018DB.pdf

[6] E. L. et al., "Albayzin 2018 Evaluation: The IberSpeech-RTVE Challenge on Speech Technologies for Spanish Broadcast Media," *Applied Sciences*, vol. 9, no. 24, p. 5412, 2019.

[7] E. Lleida *et al.*, "RTVE2020 database description," 2020. [Online]. Available: http://catedrartve.unizar.es/reto2020/RTVE2020DB.pdf

[8] "Voxforge." [Online]. Available: http://www.voxforge.org

[9] M. del Agua *et al.*, "The translectures-UPV toolkit," in *Advances in Speech and Language Technologies for Iberian Languages*, Nov. 2014, pp. 269–278.

[10] S. J. Young *et al.*, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. of Workshop on Human Language Technology*, 1994, pp. 307–312.

[11] A. Zeyer *et al.*, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," in *Proc. of ICASSP*, 2017, pp. 2462–2466.

[12] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[13] D. S. Park *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. of Interspeech*, 2019, pp. 2613–2617.

[14] J. Jorge *et al.*, "Real-Time One-Pass Decoder for Speech Recognition Using LSTM Language Models," in *Proc. of Interspeech*, 2019, pp. 3820–3824.

[15] A. Stolcke, "SRILM - an extensible language modeling toolkit." in *Proc. of Interspeech*, 2002, pp. 901–904.

[16] Y. Lin *et al.*, "Syntactic annotations for the google books ngram corpus," in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012.

[17] X. Chen *et al.*, "CUED-RNNLM – An open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *Proc. of ICASSP*, 2016, pp. 6000–6004.

[18] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *arXiv preprint arXiv:1206.6426*, 2012.

[19] X. Chen *et al.*, "Improving the training and evaluation efficiency of recurrent neural network language models," in *Proc. of ICASSP*, 2015, pp. 5401–5405.

[20] M. Ott *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. of NAACL-HLT*, 2019, pp. 48–53.

[21] P. Baquero-Arnal *et al.*, "Improved Hybrid Streaming ASR with Transformer Language Models," in *Proc. of InterSpeech*, 2020, pp. 2127–2131.

[22] "OpenSubtitles," http://www.opensubtitles.org/.

[23] "UFAL Medical Corpus," http://ufal.mff.cuni.cz/ufal_medical_corpus.

[24] "Wikipedia," https://www.wikipedia.org/.

[25] C. Callison-Burch *et al.*, "Findings of the 2012 workshop on statistical machine translation," in *Proc. of WMT*, 2012, pp. 10–51.

[26] "News Crawl corpus (WMT workshop) 2015," http://www.statmt.org/wmt15/translation-task.html.

[27] "Eldiario.es," https://www.eldiario.es/.

[28] "ElPeriodico.com," https://www.elperiodico.com/.

[29] "CommonCrawl 2014," http://commoncrawl.org/.

[30] J. Jorge *et al.*, "LSTM-Based One-Pass Decoder for Low-Latency Streaming," in *Proc. of ICASSP*, 2020, pp. 7814–7818.

[31] J. Jorge *et al.*, "Live Streaming Speech Recognition using Deep Bidirectional LSTM Acoustic Models and Interpolated Language Models," submitted to: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[1] https://mllp.upv.es/git-pub/jjorge/MLLP_Streaming_API

[2] https://ttp.mllp.upv.es/