applied sciences

MDPI

*Article*

# MLLP-VRAIN Spanish ASR Systems for the Albayzín-RTVE 2020 Speech-to-Text Challenge: Extension

**Pau Baquero-Arnal *, Javier Jorge, Adrià Giménez, Javier Iranzo-Sánchez, Alejandro Pérez, Gonçal Vicent 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@vrain.upv.es (J.J.); adgipas@vrain.upv.es (A.G.); jairsan@upv.es (J.I.-S.); alpegon2@vrain.upv.es (A.P.); gogardia@vrain.upv.es (G.V.G.D.-M.); jsilvestre@dsic.upv.es (J.A.S.-C.); jorcisai@vrain.upv.es (J.C.); josanna2@vrain.upv.es (A.S.); ajuan@dsic.upv.es (A.J.)
* Correspondence: pabaar@upv.es

**Featured Application: This work has direct application in live automatic captioning of audiovisual material, which is fundamental in accessibility.**

**Abstract:** This paper describes the automatic speech recognition (ASR) systems built by the MLLP-VRAIN research group of Universitat Politècnica de València for the Albayzín-RTVE 2020 Speech-to-Text Challenge, and includes an extension of the work consisting of building and evaluating equivalent systems under the closed data conditions from the 2018 challenge. The primary system (*p-streaming_1500ms_nlt*) was a hybrid ASR system using streaming one-pass decoding with a context window of 1.5 seconds. This system achieved 16.0% WER on the *test-2020* set. We also submitted three contrastive systems. From these, we highlight the system *c2-streaming_600ms_t* which, following a similar configuration as the primary system with a smaller context window of 0.6 s, scored 16.9% WER points on the same test set, with a measured empirical latency of 0.81 ± 0.09 s (mean ± stdev). That is, we obtained state-of-the-art latencies for high-quality automatic live captioning with a small WER degradation of 6% relative. As an extension, the equivalent closed-condition systems obtained 23.3% WER and 23.5% WER, respectively. When evaluated with an unconstrained language model, we obtained 19.9% WER and 20.4% WER; i.e., not far behind the top-performing systems with only 5% of the full acoustic data and with the extra ability of being streaming-capable. Indeed, all of these streaming systems could be put into production environments for automatic captioning of live media streams.

**Keywords:** natural language processing; automatic speech recognition; streaming



**Citation:** Baquero-Arnal, P.; Jorge, J.; Giménez, A.; Iranzo-Sánchez, J.; Pérez, A.; Garcés Díaz-Munío, G.V.; Silvestre-Cerdà, J.A.; Civera, J.; Sanchis, A.; Juan, A. MLLP-VRAIN Spanish ASR Systems for the Albayzín-RTVE 2020 Speech-to-Text Challenge: Extension. *Appl. Sci.* **2022**, *12*, 804. https://doi.org/10.3390/app12020804

Academic Editor: Francesc Alías

Received: 29 November 2021
Accepted: 8 January 2022
Published: 13 January 2022



## 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 Albayzín-RTVE 2020 Speech-to-Text (S2T) Challenge, with an extension focused on building equivalent systems under the 2018 closed data conditions. The article is an extended version of the original submission to the Challenge, presented in IberSPEECH 2020 [1].

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 advances in networking with high speed connections and proper bandwidth. Additionally, due to the COVID-19 pandemic, video meeting/conferencing platforms have experienced an exponential growth of usage, as public

and private companies have leveraged remote working 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. Additionally, more and more countries are requiring by law that TV broadcasters provide accessibility options to people with hearing disabilities, with the minimum amount of content to be captioned increasing year by year [2,3].

Some TV broadcasters and other live streaming services are using manual transcription from scratch of live audio or video streams, as an initial solution to comply with the current legislations and to satisfy user expectations. However, this is a hard task for professional transcribers who, under stressful conditions, are prone to produce 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 might be 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 transcribers to correct live transcripts provided by these ASR systems, when they are not fit for broadcast as they are. In this way, they can 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 research group has focused its research efforts in the past two years on streaming ASR. This work aims to describe our latest developments in this area, showing how advanced 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 (Radiotelevisión Española) database.

Current state-of-the-art ASR systems are based on the hybrid approach, combining Hidden Markov Models (HMM) with deep neural networks (DNN) [4] for both acoustic and language models (AM and LM). For acoustic modelling, deep Bidirectional Long-Short Term Memory (BLSTM) networks have shown to be a robust architecture achieving a good performance in an ample range of ASR tasks [5–8]. As for language modelling, Transformer-based architectures have achieved very promising results [9,10], though Long-Short Term Memory (LSTM) recurrent neural networks (RNN) are still in broad use [11]. Apart from hybrid systems, end-to-end systems have received great attention in recent years, including a number of proposals for low-latency streaming decoding [12–14]. However, despite their simplicity and promising prospects, it is still unclear whether or not they will soon surpass state-of-the-art hybrid systems combining independent models trained from vast amounts of data. In this work, we focus on the hybrid approach. One-pass decoding is a recent development within this approach, where a neural LM is used directly during search [15,16], instead of the conventional rescoring of lattices or $n$-best hypotheses in a two-pass decoding approach [17–19]. In [20], we have recently extended this architecture for real-time one-pass decoding to include BLSTM AMs with a time sliding window. This window is also used for on-the-fly acoustic feature normalization, thus enabling full streaming ASR hybrid recognition with neural AM and LMs. More recently, this architecture has been refined to include streaming-adapted Transformer LMs besides, or even replacing, LSTM-RNN LMs [21].

Our participation in the Albayzín-RTVE 2020 S2T Challenge consisted of 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. These systems capitalized on the full streaming ASR hybrid recognition approach described above, and were built using our in-house transLectures-UPV ASR toolkit (TLK) [22]. In contrast, most other participants used the readily-available Kaldi

toolkit for DNN-HMM ASR systems [23–26] with occasional end-to-end systems as contrastive submissions.

The rest of the paper is structured as follows. First, Section 2 briefly describes the Albayzín-RTVE 2020 S2T Challenge and the RTVE databases provided by the organizers. Next, Section 3 provides a detailed description of our participant ASR systems. Section 4 presents a report of equivalent ASR systems under the 2018 closed data conditions. Finally, Section 5 gives a summary of the work plus some concluding remarks.

## 2. Challenge Description and Databases

The Albayzín-RTVE 2020 Speech-To-Text Challenge consists of automatically transcribing different types of TV shows from the Spanish public TV station RTVE, and the assessment of ASR system performance in terms of Word Error Rate (WER) by comparing those automatic transcriptions with correct reference transcriptions [27].

The MLLP-VRAIN previously participated in the 2018 edition of the challenge [28], in a joint collaboration with the Human Language Technology and Pattern Recognition (HLTPR) research group of RWTH Aachen University. The evaluation was carried out on the RTVE2018 database [29], which includes 575 hours of audio from 15 different TV shows broadcast 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 [30], scoring 16.5% and 22.0% WER points respectively in the *test-2018* set.

For the 2020 edition of the challenge, a single open-condition track was proposed, and system evaluations have been carried out over the *test* (*test-2020*) set from the RTVE2020 database, which includes 78.4 speech hours at a sampling rate of 16 kHz from 15 different TV shows broadcast between 2018 and 2019 [31].

## 3. MLLP-VRAIN Systems

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

### 3.1. Acoustic Modelling

Our acoustic models (AM) were trained using 205 filtered speech hours from the RTVE2018 *train* set (187 h) and our internally split *dev1-train* set (18 h), as in [28], plus about 3.7 K hours of other resources crawled from the Internet in 2016 and earlier. 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 10 ms to train a context-dependent feed-forward DNN-HMM with three left-to-right tied states using the transLectures-UPV toolkit (TLK) [22]. The state-tying schema followed a phonetic decision tree approach [32] that produced 10 K 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 [7]. The BLSTM network was trained using both TLK and TensorFlow [33], and had 8 bidirectional hidden layers with 512 LSTM cells per layer and direction. As in [7], 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 [34].

**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 [35] | 21 |
| RTVE2018: *train* | 187 |
| RTVE2018: *dev1-train* | 18 |
| TOTAL | 3924 |

### *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 [16], using the text data sources and corpora described in Table 2.

On the one hand, we trained 4-gram LMs using SRILM [36] with all text resources plus the Google-counts v2 corpus [37], accounting for 102G running words. The vocabulary size was limited to 254 K 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 (254 K 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 [38]. The Noise Contrastive Estimation (NCE) criterion [39] was used to speed up model training, and the normalization constant learned from training was used during decoding [40]. Based on the lowest perplexity observed on our internal development set, a model with a 256-unit embedding layer and two hidden LSTM layers of 2048 units was selected. Secondly, Transformer LMs (TLMs) were trained using a customized version of the FairSeq toolkit [41], with the following configuration minimizing perplexity in our internal development set: 24-layer network with 768 units per layer, 4096-unit feed-forward neural network, 12 attention heads, and an embedding of 768 dimensions. These models were trained until convergence with batches limited to 512 tokens. Parameters were updated every 32 batches. During inference, Variance Regularization was applied to speed up the computation of TLM scores [21].

**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 [42] | 212,635 | 1,146,861 | 1576 |
| UFAL [43] | 92,873 | 910,728 | 2179 |
| Wikipedia [44] | 32,686 | 586,068 | 3373 |
| UN [45] | 11,196 | 343,594 | 381 |
| News Crawl [46] | 7532 | 198,545 | 648 |
| Internal: entertainment | 4799 | 59,235 | 307 |
| eldiario.es [47] | 1665 | 47,542 | 247 |
| El Periódico [48] | 2677 | 46,637 | 291 |

**Table 2.** *Cont.*

| Corpus | S (K) | RW (K) | V (K) |
|---|---|---|---|
| Common Crawl [49] | 1719 | 41,792 | 486 |
| Internal: parliamentary data | 1361 | 35,170 | 126 |
| News Commentary [46] | 207 | 5448 | 83 |
| Internal: educational | 87 | 1526 | 35 |
| TOTAL | 369,434 | 3,423,146 | 5785 |
| Google-counts v2 [37] | - | 97,447,282 | 3693 |

### *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 [16]. HCS groups hypotheses by their LM history, with each group representing all state hypotheses sharing a common history. In this way, word histories only need to be considered when handling word labels, and thus can be ignored during dynamic programming at intra-word state level [50]. 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, and LM histogram pruning. The static look-ahead table, needed by the decoder to use pre-computed look-ahead LM scores, is generated from a pruned 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 [20]. 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 from performing a Full Sequence Normalization, which is typically applied under the off-line setting. Instead, we applied the Weighted Moving Average (WMA) technique, which uses the content of the current context window to update normalization statistics on-the-fly [51]. This technique is applied over a batch $B_j$ of frames as

$$\hat{B}_j = B_j - \hat{\mu}_j \tag{1}$$

where

$$\hat{\mu}_j = \frac{\boldsymbol{f}_{j-1} + \sum_{t=1}^{b+w} B_{j,t}}{n_{j-1} + b + w} \tag{2}$$

being $\boldsymbol{f}_{j-1}$ the accumulated values of previous frames until batch $B_{j-1}$, $B_{j,t}$ the $t$-th frame in batch $B_j$, $n_{j-1}$ the number of frames until batch $B_{j-1}$, and $b$ and $w$ the batch and window sizes, respectively. $\boldsymbol{f}_j$ and $n_j$ are accumulated values that are updated by weighting the contribution of previous batches with an adjustable parameter $\alpha$:

$$\boldsymbol{f}_j = \alpha \cdot \boldsymbol{f}_{j-1} + \sum_{t=1}^{b} B_{j,t} \tag{3}$$

$$n_j = \alpha \cdot n_{j-1} + b \tag{4}$$

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 and evaluate our system development, we used the dev and test sets from the RTVE2018 database. For the experiments, we devoted our internally split *dev1-dev* set [28] for development purposes, whilst *dev2* and *test-2018* were used 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 3.** Basic statistics of RTVE development and tests sets, including our internally split *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 | $d_\mu$ | $\pm$ | $\sigma$ | RW (K) |
|---|---|---|---|---|---|---|
| *dev1-dev* | 11.9 | 10 | 4267 | $\pm$ | 1549 | 120 |
| *dev2* | 15.2 | 12 | 4564 | $\pm$ | 1557 | 149 |
| *test-2018* | 39.3 | 59 | 2395 | $\pm$ | 1673 | 377 |
| *test-2020* | 78.4 | 87 | 2314 | $\pm$ | 1576 | 519 |

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 significantly 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 just 6% relative when all three LMs are combined.

**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 |

Second, we tuned decoding parameters to provide a good WER-RTF tradeoff on *dev1-dev*, with the hard constraint of RTF $< 1$ to ensure real-time processing of the input. From these hyperparameters, we highlight, due to their relevance, a LM history recombination of 12, LM histogram pruning of 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/non-speech segments as in [28], followed by real-time one-pass decoding with our BLSTM-HMM AM, using a Full Sequence 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. From this exploration, a context window size of 1.5 s and $\alpha = 0.95$ was chosen for the WMA normalization technique. This configuration was used for our primary system, labelled *p-streaming_1500ms_nlt*, that showed WER rates of 11.6 and 16.0 on *test-2018* and *test-2020*, respectively. It is important to note that this system does not integrate any VAD

module. Instead, this task is left for the decoder to carry out via the implicit non-speech model of the BLSTM-HMM AM.

A small change in the configuration of the primary system, consisting in the removal of the LSTM LM from the linear interpolation, led to 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 computationally expensive, and its contribution to PPL is negligible with respect to the n-gram LM + TLM combination (3% relative improvement). We thus increased system latency stability while experiencing nearly no degradation in terms of WER: 11.6 and 16.1 points on *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 s due to the context window size. As stated in Section 3.3, this parameter can be adjusted at decoding time. This allows us to configure the decoder for lower latency responses or better transcription quality. Hence, our final goal for the challenge was to find a proper system configuration providing 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. In light of the results, we chose a window size of 0.6 s, as it brings a good balance between transcription quality and theoretical latency.

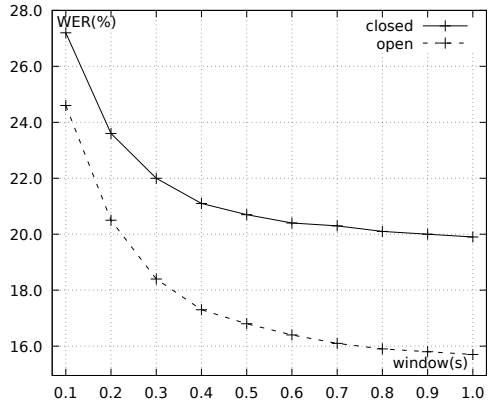


**Figure 1.** WER as a function of context window size (in seconds) for the streaming setup, computed over the *dev1-dev* set. This figure includes both the setup in this section (dashed line) and the setup for the closed-condition system described in Section 4 (solid line).

The last step to set up our latency-focused streaming system was to measure WER and empirical latencies as a function of different pruning 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 which it is fully processed by the decoder. Latency figures are provided 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 pruning parameter values, and comparing the LM combinations including the Transformer LM. These measurements were made on an Intel i7-3820 CPU @ 3.60GHz, with 64GB of RAM and a GeForce RTX 2080 Ti GPU. On the one hand, we can see how combinations involving LSTM LMs are systematically shifted rightwards with respect to 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 to remove the LSTM LM to define our contrastive system *c1-streaming_1500ms_nt*. On the other hand, the TLM alone generally provides a good baseline that is slightly improved in terms of WER if we include the other LMs. However, this comes at 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 s (mean ± stdev), and its performance was 12.3 and 16.9 WER points on *test-2018* and *test-2020*, respectively. This is, with just a very small relative WER degradation of 6% with respect to the primary system, we got state-of-the-art (mean = 0.81 s) and very stable (stdev = 0.09 s) empirical latencies. This system has a baseline consumption (when idle) of 9 GB RAM and 3.5 GB GPU memory (on a single GPU), adding 256 MB RAM and one CPU thread per decoding (audio stream). For instance, decoding four simultaneous audio streams in a single machine would use four CPU threads, 10 GB RAM and 3.5 GB GPU memory.

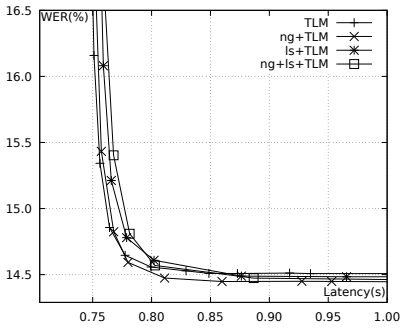


**Figure 2.** WER versus mean empirical latency (in seconds) on *dev1-dev*, measured with different pruning parameters for the search, and considering only interpolation schemes including TLM. An acoustic window size of 0.6s was used in all cases (plotted points).

Table 5 summarises the results obtained for all four participant ASR systems on the *dev2*, *test-2018* and *test-2020* sets, and also includes 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 on *test-2020*, by up to 1.1 absolute WER points (6% relative). We believe that this is caused, first, by the Gaussian mixture HMM-based VAD module producing 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 Gaussian Mixture models, it is likely to misclassify speech passages with loud background music and noise (often present in TV programmes) as non-speech. Second, the Full Sequence Normalization technique might not be appropriate for some types of TV shows, as local acoustic condition changes become diluted in the full-sequence normalization, leading to somewhat inaccurate acoustic scores 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*), and also works under strict streaming conditions.

**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. In bold, the result from our primary submission in the contest main test set *test-2020*.

| 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 [28] | 15.6 | 16.5 | - |

All these streaming ASR systems can be easily put into production environments using our custom gRPC-based server-client infrastructure ( https://mllp.upv.es/git-pub/jjorge/MLLP_Streaming_API, accessed on 7 January 2022). Indeed, ASR systems comparable to *c2-streaming_600ms_t* and *c1-streaming_1500ms_nt* are already in production at our MLLP

Transcription and Translation Platform (https://ttp.mllp.upv.es/, accessed on 7 January 2022) for streaming and off-line processing, respectively. Both can be freely tested using our public APIs, accessible via the MLLP Platform.

## 4. Closed-Condition Systems

For better comparison with results from the 2018 challenge, experiments similar to those reported above were carried out using only the data available for the 2018 challenge under closed data conditions. These experiments and the results obtained are reported in this section.

### 4.1. Acoustic Modelling

As in [28], acoustic models were trained using only the *train* and *dev1-train* sets (see Table 1); that is, 205 hours in total, accounting only for 5.2% of the 3924 h used for our open challenge submissions. They were preprocessed as described in Section 3, using MFCC features, HMMs with state tying, BLSTMs, and SpecAugmentation.

### 4.2. Language Modelling

For language modelling, we followed the same approach as in the previous section and trained both n-gram and neural LMs to perform one-pass decoding with linear combinations of them. We used significantly fewer text data to comply with the closed condition constraint. Specifically, we used the same data as in [28], comprising 5.2 M sentences (representing 1.4% of the full data) and 96 M running words (2.2%) with a vocabulary size of 132 K (2.3%), obtaining an OOV ratio less than 0.6% computed over our internal development sets and *test2018*, and less than 0.8% over *test2020*.

For the n-gram LMs, we used a single model making use of the fraction of the data that was available, in contrast to our submissions to the open challenge where we used an interpolation of several n-gram LMs trained on different subsets. As for the neural models, we decided against the use of LSTM LMs, and thus only Transformer LMs were considered. This decision was based on the empirical results reported in Section 3.4, where we also decided to remove the LSTM LM to define our contrastive system *c1-streaming_1500ms_nt*. Apart from this decision, both the neural architecture and training methodology were kept the same.

### 4.3. Experiments and Results

An empirical study similar to that described in Section 3 was carried out using the development and test sets in Table 3. The first step was to compare n-gram and Transformer LMs, as well as their combination, in terms of perplexity on the *dev1-dev* set. Table 6 shows the results obtained, including the optimal interpolation weights we found for their combination.

**Table 6.** Perplexity (PPL) and interpolation weights, computed over the *dev1-dev* set, of the n-gram (ng) and Transformer (tf) LMs and their combination when trained with restricted data.

| LM Comb. | PPL | Weights (%) |
|---|---|---|
| ng | 164 | - |
| tf | 103 | - |
| ng + tf | 84 | 70 + 30 |

As a second step, we studied the effect of the theoretical acoustic latency on system performance. Figure 1 shows the WER as a function of the AM context window size, for window sizes under one second, and using the simpler n-gram LM in isolation. For comparison, the performance of the open-condition system is also shown. From the results in Figure 1, and in agreement with the open-condition experimentation of Section 3, an acoustic window size of 0.6 s was selected for further experiments under the streaming setup aimed at measuring real latencies.

For the final streaming and latency tests, the WER was plotted against empirical latencies with different pruning parameters and LM combinations. All measurements were made on the same hardware we used to assess the open condition systems (Section 3). Figure 3 shows the WER as a function of the mean empirical latency computed over *dev1-dev*. For comparison, the corresponding results for the open-condition systems are also included. In contrast to the open-condition results, a significant improvement is achieved by combining the n-gram and Transformer LMs, instead of just using the Transformer LM alone. Therefore, for comparative purposes, the systems using interpolated n-gram and Transformer LMs are considered our final closed-condition systems.

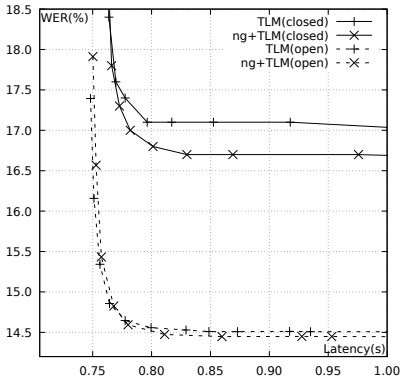


**Figure 3.** WER as a function of system latency (in seconds) for the closed-condition systems (solid line), computed over the *dev1-dev* set by trying different values for the search (pruning) parameters. An acoustic window size of 0.6s was used in all cases (plotted points). For comparison, relevant results for the open-condition systems are also shown (dashed line).

Table 7 contains a summary of the most relevant results obtained with our systems, both under open and closed data conditions. For the closed-condition systems, results for data-unrestricted LMs are also included to check how far we can go without expanding the acoustic data, better reflecting a usual scenario where text data is relatively much easier to obtain than audio with transcriptions. For comparison, Table 7 includes the figures for the 2018 contest winners and results achieved by other contestants in the 2020 challenge.

**Table 7.** WER of our main (open) submissions and our closed-condition systems described in this section, including for comparison the winning open- and closed-condition systems of the 2018 challenge, computed over the *dev2*, *test-2018* and *test-2020* sets. Highlited in bold, the most relevant results to compare open and closed conditions in the main test set *test-2020*.

| System | Duration (h) | *dev2* | *test-2018* | *test-2020* |
|---|---|---|---|---|
| *open-p-streaming_1500ms_nlt* | 3924 | 11.2 | 11.6 | **16.0** |
| *open-c2-streaming_600ms_t* | 3924 | 12.0 | 12.3 | **16.9** |
| *closed-streaming_600ms_nt2018* | 205 | 15.0 | 15.3 | 23.5 |
| *+ open ng* | | 15.6 | 15.9 | 23.2 |
| *+ open ng+tf* | | 13.3 | 13.7 | **20.4** |
| *closed-streaming_1500ms_nt2018* | 205 | 14.7 | 15.3 | 23.1 |
| *+ open ng* | | 15.3 | 15.8 | 22.9 |
| *+ open ng+tf* | | 13.0 | 13.7 | **19.9** |
| 2018 open-cond. winner [28] | 3800 | 15.6 | 16.5 | - |
| 2018 closed-cond. winner [28] | 205 | 20.0 | 22.0 | - |
| 2020 *Vicomtech* [23] | 743 | n/p | n/p | 19.3 |
| 2020 *BRNO* [24] | 780 | 12.8 | 13.3 | 23.2 |
| 2020 *Sigma-UPM* [25] | 615 | n/p | n/p | 27.7 |
| 2020 *Biometric Vox System* [26] | 1000 | 17.8 | 22.0 | 30.3 |

From the results in *test-2020*, we can observe that, while there is a wide margin between the open- and closed-condition systems, we can still obtain a performant streaming ASR system when using only around 5% of the acoustic data and 2% of the text data. Moreover, the WER reduction entailed by moving from an acoustic latency of 1.5 to 0.6 s in the open-condition system is largely gone in the closed-condition systems: it is almost negligible on *test-2018* and just 0.5 WER on *test-2020*. Furthermore, lifting the text data restriction, the *closed-streaming_1500ms_nt2018* system is able to shed over 3 WER points, from 23.1 to 19.9 WER, that is, about a 15% relative improvement and not far behind the top-performing systems submitted by other participants under open conditions. Regarding this change from the closed-condition LM to the unrestricted LM, both in the 1500 ms and the 600 ms window sizes, most of the improvement is due to the substitution rate: from 10.7 to 7.3 in the 1500 ms case, and from 10.9 to 7.4 in the 600 ms case. This accounts mostly for the unrestricted LM retrieving the correct word due to its expanded lexicon—the OOV rate fell from 0.8 to 0.6, as previously mentioned—or due to better tuned LM probabilities. Some examples of correctly recognized words that were errors in the closed-condition LM are "desacreditarme", "dosificador" or "hermeneuta".

## 5. Conclusions

In this work, we have first described our four ASR systems that participated in the Albayzín-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 on 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 s. Nearly the same performance was delivered by our first contrastive system (*c1-streaming_1500ms_nt*): 16.1 WER points on *test-2020*, at a significantly 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 $0.81 \pm 0.09$ s (mean $\pm$ stdev). Finally, our fourth ASR system was a contrastive off-line ASR system with VAD (*c3-offline*) providing the highest, yet still competitive, WER score of 17.1 points, attributable to an improvable VAD module and to the limitations of Full Sequence Normalization when dealing with local acoustic condition changes.

Apart from the four ASR systems participating in the 2020 (open) challenge, two additional streaming systems have been described which, for better comparison with results from the 2018 challenge, were trained under the 2018 closed data conditions (about 5% and 2%, respectively, of the speech and text data we used in 2020). The first, latency-focused system (*closed-streaming_600ms_nt2018*) achieves 23.5 WER points on test-2020, while the second, performance-focused system (*closed-streaming_1500ms_nt2018*) reduces this figure to 23.1 WER points. Moreover, when using these systems with unconstrained language models (i.e., trained with all the text data used in 2020), these figures are further reduced to 20.4 (at 600 ms latency) and 19.9 (at 1500 ms latency). It is worth noting that these WER figures have been achieved under streaming conditions using only 205 h of RTVE2018 speech training data, thus not including any post-2018 RTVE speech data, and accounting for only 5% of the training data used in the 2020 open challenge. Nevertheless, these WER figures are not far behind those of the second-best 2020 open-condition system (19.3) and they are significantly ahead of those reported by other participants.

**Author Contributions:** Conceptualization, P.B.-A., J.J., A.G., J.A.S.-C., J.C., A.S. and A.J.; methodology, P.B.-A., J.J., A.G., J.A.S.-C., A.S. and A.J.; software, P.B.-A., J.J., A.G., J.I.-S., A.P. and J.A.S.-C.; validation, P.B.-A., J.J., A.G., J.A.S.-C., A.S. and A.J.; formal analysis, P.B.-A., J.J., A.G., J.A.S.-C., A.S. and A.J.; investigation, P.B.-A., J.J., A.G.; resources, A.P., J.A.S.-C.; data curation, J.A.S.-C.; writing—original draft preparation, P.B.-A., J.J., J.A.S.-C.; writing—review and editing, P.B.-A., J.J., A.G., J.I.-S., G.V.G.D.-M., J.A.S.-C., A.S. and A.J.; visualization, P.B.-A., J.J., J.A.S.-C.; supervision, J.A.S.-C., A.S. and A.J.; project administration, J.A.S.-C., J.C., A.S. and A.J.; funding acquisition, P.B.-A., J.J., A.G.,

J.I.-S., A.P., G.V.G.D.-M., J.A.S.-C., J.C., A.S. and A.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements no. 761758 (X5Gon) and 952215 (TAILOR), and Erasmus+ Education programme under grant agreement no. 20-226-093604-SCH (EXPERT); the Government of Spain's grant RTI2018-094879-B-I00 (Multisub) funded by MCIN/AEI/10.13039/501100011033 & "ERDF A way of making Europe", and FPU scholarships FPU14/03981 and FPU18/04135; the Generalitat Valenciana's research project Classroom Activity Recognition (ref. PROMETEO/2019/111), and predoctoral research scholarship ACIF/2017/055; and the Universitat Politècnica de València's PAID-01-17 R&D support programme.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The RTVE2018 and RTVE2020 speech databases used in this article for ASR training and evaluation are available subject to the terms of a licence agreement with Corporación de Radio y Televisión Española (RTVE). These data were obtained from RTVE and are available at http://catedrartve.unizar.es/rtvedatabase.html, accessed on 7 January 2022.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| AM | Acoustic Model |
| ASR | Automatic Speech Recognition |
| BLSTM | Bidirectional Long-Short Term Memory |
| dev | development (validation) |
| DNN | Deep Neural Network |
| G | billion ($10^9$) |
| HCS | History Conditioned Search |
| HMM | Hidden Markov Model |
| K | thousand ($10^3$) |
| LSTM | Long-Short Term Memory |
| LM | Language Model |
| M | million ($10^6$) |
| MFCC | Mel-Frequency Cepstral Coefficients |
| PPL | Perplexity |
| RNN | Recurrent Neural Network |
| RTF | Real Time Factor |
| RTVE | Corporación de Radio y Televisión Española |
| RW | Running Words |
| S2T | Speech to Text |
| TLM | Transformer Language Model |
| VAD | Voice Activity Detection |
| WER | Word Error Rate |
| WMA | Weighted Moving Average |

## References

1. Jorge, J.; Giménez, A.; Baquero-Arnal, P.; Iranzo-Sánchez, J.; de Martos, A.P.G.; Garcés Díaz-Munío, G.V.; Silvestre-Cerdà, J.A.; Civera, J.; Sanchis, A.; Juan, A. MLLP-VRAIN Spanish ASR Systems for the Albayzin-RTVE 2020 Speech-To-Text Challenge. In Proceedings of the IberSPEECH, Valladolid, Spain, 24–25 March 2021; pp. 118–122.
2. Royal Decree 1494/2007 (Spain) on Accessibility to the Media. Available online: https://www.boe.es/buscar/act.php?id=BOE-A-2007-19968 (accessed on 7 January 2022).
3. Law 1/2006 (Comunitat Valenciana, Spain) on the Audiovisual Sector. Available online: https://www.dogv.gva.es/va/eli/es-vc/l/2006/04/19/1/dof/vci-spa/pdf (accessed on 7 January 2022).
4. Yu, D.; Deng, L. *Automatic Speech Recognition: A Deep Learning Approach*; Springer Publishing Company, Incorporated: New York, NY, USA, 2014.

5. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
6. Chen, K.; Huo, Q. Training deep bidirectional LSTM acoustic model for LVCSR by a Context-Sensitive-Chunk BPTT approach. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 1185–1193. [CrossRef]
7. Zeyer, A.; Doetsch, P.; Voigtlaender, P.; Schlüter, R.; Ney, H. A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2462–2466.
8. Zeyer, A.; Schlüter, R.; Ney, H. Towards online-recognition with deep bidirectional LSTM acoustic models. In Proceedings of the InterSpeech, San Francisco, CA, USA, 8–12 September 2016; pp. 3424–3428. [CrossRef]
9. Irie, K.; Zeyer, A.; Schlüter, R.; Ney, H. Language modeling with deep transformers. In Proceedings of the InterSpeech, Graz, Austria, 15–19 Septembar 2019; pp. 3905–3909. [CrossRef]
10. Beck, E.; Schlüter, R.; Ney, H. LVCSR with Transformer Language Models. In Proceedings of the InterSpeech, Shanghai, China, 25–29 October 2020; pp. 1798–1802. [CrossRef]
11. Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; Wu, Y. Exploring the limits of language modeling. *arXiv* **2016**, arXiv:1602.02410.
12. Moritz, N.; Hori, T.; Le Roux, J. Streaming automatic speech recognition with the transformer Model. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6074–6078.
13. Miao, H.; Cheng, G.; Gao, C.; Zhang, P.; Yan, Y. Transformer-based online CTC/attention end-to-end speech recognition architecture. *arXiv* **2020**, arXiv:2001.08290.
14. Nguyen, T.S.; Pham, N.Q.; Stüker, S.; Waibel, A. High performance sequence-to-sequence model for streaming speech recognition. *arXiv* **2020**, arXiv:2003.10022.
15. Beck, E.; Zhou, W.; Schlüter, R.; Ney, H. LSTM Language Models for LVCSR in First-Pass Decoding and Lattice-Rescoring. *arxiv* **2019**, arxiv:1907.01030.
16. Jorge, J.; Giménez, A.; Iranzo-Sánchez, J.; Saiz, J.C.; Sanchís, A.; Juan-Císcar, A. Real-Time One-Pass Decoder for Speech Recognition Using LSTM Language Models. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 3820–3824. [CrossRef]
17. Chen, X.; Liu, X.; Ragni, A.; Wang, Y.; Gales, M.J.F. Future word contexts in neural network language models. In Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, Japan, 16–20 December 2017; pp. 97–103.
18. Ogawa, A.; Delcroix, M.; Karita, S.; Nakatani, T. Rescoring N-Best speech recognition list based on one-on-one hypothesis comparison using encoder-classifier model. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 6099–6103.
19. Kombrink, S.; Mikolov, T.; Karafiát, M.; Burget, L. Recurrent Neural Network based language modeling in meeting recognition. In Proceedings of the Interspeech, Florence, Italy, 27–31 August 2011; pp. 2877–2880. [CrossRef]
20. Jorge, J.; Giménez, A.; Iranzo-Sánchez, J.; Silvestre-Cerdà, J.A.; Civera, J.; Sanchis, A.; Juan, A. LSTM-Based One-Pass Decoder for Low-Latency Streaming. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7814–7818.
21. Baquero-Arnal, P.; Jorge, J.; Giménez, A.; Silvestre-Cerdà, J.A.; Iranzo-Sánchez, J.; Sanchís, A.; Saiz, J.C.; Juan-Císcar, A. Improved Hybrid Streaming ASR with Transformer Language Models. In Proceedings of the Interspeech, Shaghai, China, 25–29 April 2020; pp. 2127–2131. [CrossRef]
22. del Agua, M.; Giménez, A.; Serrano, N.; Andrés-Ferrer, J.; Civera, J.; Sanchis, A.; Juan, A. The translectures-UPV toolkit. In *Advances in Speech and Language Technologies for Iberian Languages*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 269–278.
23. Álvarez, A.; Arzelus, H.; Torre, I.G.; González-Docasal, A. The Vicomtech Speech Transcription Systems for the Albayzín-RTVE 2020 Speech to Text Transcription Challenge. In Proceedings of the IberSPEECH 2021, Valladolid, Spain, 24–25 March 2021; pp. 104–107. [CrossRef]
24. Kocour, M.; Cámbara, G.; Luque, J.; Bonet, D.; Farrús, M.; Karafiát, M.; Veselý, K.; Černocký, J. BCN2BRNO: ASR System Fusion for Albayzin 2020 Speech to Text Challenge. In Proceedings of the IberSPEECH, Valladolid, Spain, 24–25 March 2021; pp. 113–117. [CrossRef]
25. Perero-Codosero, J.M.; Espinoza-Cuadros, F.M.; Hernández-Gómez, L.A. Sigma-UPM ASR Systems for the IberSpeech-RTVE 2020 Speech-to-Text Transcription Challenge. In Proceedings of the IberSPEECH, Valladolid, Spain, 24–25 March 2021; pp. 108–112. [CrossRef]
26. Font, R.; Grau, T. The Biometric Vox System for the Albayzin-RTVE 2020 Speech-to-Text Challenge. In Proceedings of the IberSPEECH, Valladolid, Spain, 24–25 March 2021; pp. 99–103. [CrossRef]
27. Lleida, E.; Ortega, A.; Miguel, A.; Bazán, V.; Pérez, C.; Gómez, M.; de Prada, A. Albayzin Evaluation: IberSPEECH-RTVE 2020 Speech to Text Transcription Challenge. Available online: http://catedrartve.unizar.es/reto2020/EvalPlan-S2T-2020-v1.pdf (accessed on 7 January 2022).

28. Jorge, J.; Martínez-Villaronga, A.; Golik, P.; Giménez, A.; Silvestre-Cerdà, J.A.; Doetsch, P.; Císcar, V.A.; Ney, H.; Juan, A.; Sanchis, A. MLLP-UPV and RWTH Aachen Spanish ASR Systems for the IberSpeech-RTVE 2018 Speech-to-Text Transcription Challenge. In Proceedings of the IberSPEECH, Barcelona, Spain, 21–23 November 2018; pp. 257–261.
29. Lleida, E.; Ortega, A.; Miguel, A.; Bazán, V.; Pérez, C.; Gómez, M.; de Prada, A. RTVE2018 Database Description. Available online: http://catedrartve.unizar.es/reto2018/RTVE2018DB.pdf (accessed on 7 January 2022).
30. Lleida, E.; Ortega, A.; Miguel, A.; Bazán, V.; Pérez, C.; Gómez, M.; de Prada, A. Albayzin 2018 Evaluation: The IberSpeech-RTVE Challenge on Speech Technologies for Spanish Broadcast Media. *Appl. Sci.* **2019**, *9*, 5412. [CrossRef]
31. Lleida, E.; Ortega, A.; Miguel, A.; Bazán, V.; Pérez, C.; Gómez, M.; de Prada, A. RTVE2020 Database Description. Available online: http://catedrartve.unizar.es/reto2020/RTVE2020DB.pdf (accessed on 7 January 2022).
32. Young, S.J.; Odell, J.J.; Woodland, P.C. Tree-based State Tying for High Accuracy Acoustic Modelling. In Proceedings of the Workshop on Human Language Technology, Plainsboro, NJ, USA, 8–11 March 1994; pp. 307–312.
33. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available online: https://www.tensorflow.org/ (accessed on 7 January 2022).
34. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 2613–2617. [CrossRef]
35. VoxForge. Available online: http://www.voxforge.org (accessed on 7 January 2022).
36. Stolcke, A. SRILM—An extensible language modeling toolkit. In Proceedings of the Interspeech, Denver, CO, USA, 16–20 September 2019; pp. 901–904.
37. Lin, Y.; Michel, J.B.; Lieberman Aiden, E.; Orwant, J.; Brockman, W.; Petrov, S. Syntactic annotations for the Google Books Ngram Corpus. In Proceedings of the ACL 2012 System Demonstrations, Jeju Island, Korea, 10 July 2012.
38. Chen, X.; Liu, X.; Qian, Y.; Gales, M.; Woodland, P. CUED-RNNLM—An open-source toolkit for efficient training and evaluation of recurrent neural network language models. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 6000–6004. [CrossRef]
39. Mnih, A.; Teh, Y.W. A fast and simple algorithm for training neural probabilistic language models. *arXiv* **2012**, arXiv:1206.6426.
40. Chen, X.; Liu, X.; Gales, M.; Woodland, P. Improving the training and evaluation efficiency of recurrent neural network language models. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5401–5405.
41. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 48–53.
42. OpenSubtitles. Available online: http://www.opensubtitles.org/ (accessed on 7 January 2022).
43. UFAL Medical Corpus. Available online: http://ufal.mff.cuni.cz/ufal_medical_corpus (accessed on 7 January 2022).
44. Wikipedia. Available online: https://www.wikipedia.org/ (accessed on 7 January 2022).
45. Callison-Burch, C.; Koehn, P.; Monz, C.; Post, M.; Soricut, R.; Specia, L. Findings of the 2012 Workshop on Statistical Machine Translation. In Proceedings of the WMT, Montréal, QC, Canada, 7–8 June 2012; pp. 10–51.
46. News Crawl corpus (WMT Workshop) 2015. Available online: http://www.statmt.org/wmt15/translation-task.html (accessed on 7 January 2022).
47. Eldiario.es. Available online: https://www.eldiario.es/ (accessed on 7 January 2022).
48. ElPeriodico.com. Available online: https://www.elperiodico.com/ (accessed on 7 January 2022).
49. CommonCrawl 2014. Available online: http://commoncrawl.org/ (accessed on 7 January 2022).
50. Ney, H.; Ortmanns, S. Progress in dynamic programming search for LVCSR. *Proc. IEEE* **2000**, *88*, 1224–1240. [CrossRef]
51. Jorge, J.; Giménez, A.; Sivestre-Cerdà, J.A.; Civera, J.; Sanchis, A.; Juan, A. 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. Available online: https://ieeexplore.ieee.org/document/9645238 (accessed on 7 January 2022).