




Article

A Comparison of Hybrid and End-to-End ASR Systems for the IberSpeech-RTVE 2020 Speech-to-Text Transcription Challenge

Juan M. Perero-Codosero ^{1,2,*} , Fernando M. Espinoza-Cuadros ^{1,2,*}  and Luis A. Hernández-Gómez ^{2,*} ¹ Sigma Technologies S.L.U., 28050 Madrid, Spain² GAPS Signal Processing Applications Group, Universidad Politécnica de Madrid, 28040 Madrid, Spain* Correspondence: jmperero@sigma-ai.com (J.M.P.-C.); fmespinoza@sigma-ai.com (F.M.E.-C.); luisalfonso.hernandez@upm.es (L.A.H.-G.)

Abstract: This paper describes a comparison between hybrid and end-to-end Automatic Speech Recognition (ASR) systems, which were evaluated on the IberSpeech-RTVE 2020 Speech-to-Text Transcription Challenge. Deep Neural Networks (DNNs) are becoming the most promising technology for ASR at present. In the last few years, traditional hybrid models have been evaluated and compared to other end-to-end ASR systems in terms of accuracy and efficiency. We contribute two different approaches: a hybrid ASR system based on a DNN-HMM and two state-of-the-art end-to-end ASR systems, based on Lattice-Free Maximum Mutual Information (LF-MMI). To address the high difficulty in the speech-to-text transcription of recordings with different speaking styles and acoustic conditions from TV studios to live recordings, data augmentation and Domain Adversarial Training (DAT) techniques were studied. Multi-condition data augmentation applied to our hybrid DNN-HMM demonstrated WER improvements in noisy scenarios (about 10% relatively). In contrast, the results obtained using an end-to-end PyChain-based ASR system were far from our expectations. Nevertheless, we found that when including DAT techniques, a relative WER improvement of 2.87% was obtained as compared to the PyChain-based system.

Keywords: TV show speech-to-text transcription; ASR systems; hybrid DNN-HMM; end-to-end deep learning; domain adversarial training



Citation: Perero-Codosero, J.M.; Espinoza-Cuadros, F.M.; Hernández-Gómez, L.A. A Comparison of Hybrid and End-to-End ASR Systems for the IberSpeech-RTVE 2020 Speech-to-Text Transcription Challenge. *Appl. Sci.* **2022**, *12*, 903. <https://doi.org/10.3390/app12020903>

Academic Editors: António Joaquim da Silva Teixeira, Francesc Alías, Valentin Cardeñoso-Payo, David Escudero-Mancebo and César González-Ferreras

Received: 22 December 2021

Accepted: 15 January 2022

Published: 17 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the advancement of deep learning techniques has been able to improve the performance of Automatic Speech Recognition (ASR) systems. At the beginning, Deep Neural Networks (DNNs) became a fundamental part of conventional hybrid ASR systems [1]. According to some research studies [2], these models perform better in many scenarios with a small amount of training data, but they usually require strong context-dependent trees to train the models [3].

Nevertheless, end-to-end approaches are emerging [4,5] due to the reduction of the complexity associated with the training process. Whilst hybrid systems need to use Hidden Markov Model (HMM) state probabilities to train the outputs of a DNN, end-to-end systems are trained to map an input feature sequence to a sequence of characters [6,7]. Furthermore, the independence of intermediate modeling (e.g., acoustic, pronunciation, and language models) makes it easier to build an ASR model. They neither require any phoneme alignment for framewise cross-entropy, nor a sophisticated beam search decoder [8].

Several approaches have appeared such as Connectionist Temporal Classification (CTC) [4], the Recurrent Neural Network Transducer (RNN-T) [3], and the sequence-to-sequence attention-based encoder–decoder [5,9]. This trend presents an easy-to-use and easy-to-update pipeline. First, the training process does not have several stages, in which more than a single model would be involved. Second, the continuous advances in deep-learning-based technologies have allowed the quick development of powerful open-source libraries for machine learning, such as PyTorch [10] or TensorFlow [11], among others.

The promising results reported by many end-to-end ASR systems depend on the scenarios, as well as on the availability of datasets. Thus, end-to-end ASR models have achieved state-of-the-art results on the LibriSpeech database [12] and large public [13] or proprietary datasets [6]. These end-to-end models demand the availability of large training datasets [6], required for training very complex deep architectures [12]. However, in noisy scenarios and low-resource domains, such as CHiME-6 [14], end-to-end methods are still far from reaching the performance of HMM-based systems, as was reported, for example, in Ref. [15]. Thus far, end-to-end systems have not been able to overcome the best conventional hybrid models in those challenging conditions.

It is a fact that up to now, there is still a gap between end-to-end systems and hybrid models. To tackle this, some studies, such as SpeechStew [16], focused on demonstrating that a model is able to learn powerful transfer learning representations from a high volume of available speech data. The authors pointed out that training large models is expensive and not practical to perform frequently, but transfer learning allows fine-tuning a model pretrained on a combination of several public speech recognition datasets.

Recent developments focused on reducing this gap have reported good results, as was the case of ESPNet [17] or PyChain [18]. In PyChain, the end-to-end LF-MMI criterion, which is the state-of-the-art for hybrid models in Kaldi [19], is implemented by combining a single-stage training and a full parallelization under the PyTorch framework. Other speech recognition challenges, such as multichannel robust end-to-end ASR, have been addressed by a joint training of DNN-based front-end (speech enhancement) and back-end (speech recognition) models based on CTC-Attention and the RNN-T [15].

Besides that, to improve the performance of end-to-end ASR systems, a variety of techniques commonly applied in deep Learning have been introduced. Data augmentation techniques [20,21] have been developed to increase the quality and variety of training data following some criteria to improve the model robustness. Thus, a variety of scenarios can be simulated trying to cover the more challenging acoustic conditions in a cost-effective way.

Other works have been focused on the enhancement of deep acoustic models, where a sequence of local feature vectors is squeezed into a single global context vector [22], representing both speaker and environment information. In addition, model agnostic meta-learning has also been applied to rapidly adapt ASR models on cross-accented speech [23].

Other recent deep-learning-based techniques, such as Domain Adversarial Training (DAT) [24], have demonstrated that the model is able to reuse a latent space to improve performance on unseen input domains. Acoustic features must be robust to model the wide variety of speaker characteristics [25] and can play a relevant role in avoiding the bias in ASR systems with regard to diversity in gender, age, regional accents, and non-native accents, as was reported in Ref. [26]. To this end, DAT has been applied to ASR tasks by learning features invariant to different conditions, such as acoustic variabilities [27,28], accented speech [29], and inter-speaker feature variability [30].

In this paper, our aim was to contribute to the comparison of both hybrid and end-to-end ASR systems under the conditions of the IberSpeech-RTVE 2020 Speech-to-Text Transcription Challenge [31]. This can be considered one of the aforementioned complex scenarios containing a variety of TV shows and broadcast news, in different noisy environments and challenging scenarios, such as TV debates. For this purpose:

- We firstly studied state-of-the-art techniques for hybrid and end-to-end ASR systems;
- We report the use of data augmentation techniques to improve our Kaldi-based hybrid ASR system presented in the IberSpeech-RTVE 2018 edition [32];
- Then, we evaluated a baseline end-to-end system on a real TV content dataset. We chose PyChain because it is based on the state-of-the-art LF-MMI approach, for which good results have been previously reported;
- Finally, looking to improve the end-to-end ASR system, we propose the use of DAT to learn features invariant to the environmental conditions and TV show format. Thus, we developed a novel improved version of the PyChain baseline including DAT. This

implementation allowed us to compare the performance of both end-to-end systems in the case of having low-computational or -speech data resources.

The rest of the paper is structured as follows. In Section 2, we describe the architecture of the ASR systems: a Kaldi-based hybrid ASR and two PyChain-based end-to-end systems. Section 3 explains the experimental protocols we followed under the IberSpeech-RTVE 2020 Challenge. Results are shown and discussed in Section 4. Section 5 presents related works comparing our systems with those submitted to IberSpeech-RTVE 2020 and other approaches in prior work. Finally, we present our conclusions in Section 6.

2. Architectures

2.1. DNN-HMM ASR

This system is based on the Sigma ASR system [32] submitted to the Albayzin-RTVE 2018 Speech-to-Text Challenge [33], where it was in the top 2 ranking for both closed- and open-condition evaluation.

This hybrid ASR system was built by using the Kaldi Toolkit [2]. The acoustic model is based on Deep Neural Networks and Hidden Markov Models (DNN-HMMs), following the so-called chain models [19], whose neural part is a subsampled Time-Delay Neural Network (TDNN) [34]. This implementation uses a 3-fold reduced frame rate at the output of the network.

We used the conventional feature pipeline that involves splicing 13-dimensional MFCC coefficients across 9 frames, followed by applying Linear Discriminant Analysis (LDA) to reduce the dimension to 40 and further decorrelation by means of Maximum Likelihood Linear Transform (MLLT). In addition, Feature-space Maximum Likelihood Linear Regression (fMLLR) was applied in a speaker-adaptive way. The input feature vectors were represented by 40-dimensional MFCC spliced coefficients across 7 frames and LDA+MLLT+fMLLR corresponding to 3 frames on each side of the central frame. In addition, 100-dimensional i-vectors were appended to the 40-dimensional acoustic space on each frame.

Our main conclusion from the results of Albayzin-RTVE 2018 [33] was the need for more robust DNN training, looking for accuracy improvements, required in the most challenging scenarios (street interviews, game shows, risky sports documentaries, etc.). The environmental robustness of acoustic models has been significantly improved by using multi-condition training data. However, the data collection process is very costly compared to the artificial generation of new training data, which has become a common alternative [20].

Thus, in our current contribution, we extended the amount of training data through data augmentation techniques. In particular, we added reverberation to the available training speech data following the approach presented in Ref. [35]. Depending on the expected scenarios and distances, different Room Impulse Responses (RIRs) can be used. They sample the room parameters and receiver position in the room and then randomly generate a number of RIRs according to different speaker positions. In short, three sets of simulated RIRs were applied: small room (1–10 m), medium room (10–30 m), and large room (30–50 m). The real computation was carried out at the feature extraction level, where the original data were mixed with their reverberated copies. The result was a 2-fold training set.

This data augmentation technique was added to other data augmentation techniques already used in the recipe followed in our previous system such as volume and speed perturbations [20,21].

2.2. End-to-End LF-MMI ASR

Aiming to explore new state-of-the-art end-to-end ASR systems, we evaluated an alternative to the developed Kaldi-based hybrid ASR system. That was the new end-to-end ASR Lattice-Free Maximum Mutual Information (LF-MMI) approach [19], which is also

used in Kaldi's chain models. Thus, we found it interesting to perform a reliable comparison of these two systems based on LF-MMI under IberSpeech-RTVE 2020 Challenge scenarios.

This end-to-end ASR system is based on PyChain [18], a powerful PyTorch-based implementation, which is intended to have an easy-to-use pipeline in which the data preparation and final decoding are carried out in Kaldi for efficiency, while data loading and network training are performed in PyTorch [10]. It should be noted that no alignment was necessary, i.e., the HMM-GMM training stage is not required, unlike other systems [36,37].

Data preparation consisted of both feature extraction (40-dimensional MFCC) and numerator/denominator graph (FSTs) generation. By following Kaldi's method for LF-MMI, HMM graphs were used for supervision. Consequently, the final LF-MMI loss function can be expressed as follows:

$$L_{MMI} = \sum_{u=1}^U \log \frac{P(X^{(u)}|\mathbb{G}_{num}^{(u)})}{P(X^{(u)}|\mathbb{G}_{den}^{(u)})} \quad (1)$$

where $X^{(u)}$ is the input frame sequences for the u -th utterance, while \mathbb{G}_{num} and \mathbb{G}_{den} are the numerator and denominator graph, respectively. These graphs are a combination of an n -gram phone Language Model (LM) with the acoustic part encoding all possible word sequences. As is widely known, \mathbb{G}_{den} is generated from any possible transcription, while \mathbb{G}_{num} makes use of the true transcription.

The probability distribution function (pdf) is used to estimate the likelihood of an HMM emission [2]. In this case, the network output and the occupation probability are computed from a pdf-index (pdf-id) instead of an HMM state. More specific details were presented in Ref. [18].

Once the data are loaded, the PyTorch model tries to simulate a TDNN [34] by including 1D dilated convolution in addition to batch normalization, ReLU, and dropout. This sequence is stacked in that order up to 6 layers with residual connections. At the end of the sequence, a fully connected layer is added (as described in Ref. [18]). From now on, this system is called the PyChain-based baseline system.

2.3. End-to-End LF-MMI ASR Applying Domain Adversarial Training

Different acoustic conditions of TV shows can have a negative impact on the PyChain-based baseline's performance. To reduce this effect, we explored the integration of DAT [24], trying to improve the PyChain-based baseline system. More specifically, in this approach, we aimed to make acoustic representations invariant to the domain of the TV show characteristics by using a Domain Adversarial Neural Network (DANN).

For this adversarial architecture, a training dataset denoted as $\{x_i, y_i, z_i\}_{i=1}^N$ is composed of x_i , which are the acoustic features, and y_i, z_i , which are the posteriors of the senones and the type of TV show, respectively.

Different from the PyChain-based baseline system training, in which the acoustic representation is trained so as to minimize the LF-MMI loss function, in DAT, the acoustic representations are learned adversarially against the secondary task (i.e., TV show classification). In this way, the domain-dependent information is suppressed in the representation, as it is irrelevant for the primary task (i.e., posterior classifier).

As can be seen in Figure 1, the parameters of our adversarial architecture consist of three parts, $\theta = \{\theta_x, \theta_y, \theta_z\}$, where θ_x denotes the parameters of the first layers of the TDNN used as the feature extractor and θ_y and θ_z denote the parameters of the pdf posteriors and the TV show classifier sub-networks, respectively.

Between the feature extractor and the TV show classifier, a Gradient Reversal Layer (GRL) [24] was implemented. In the forward propagation, the GRL keeps the input unchanged and reverses the gradient by multiplying it by a negative coefficient during the backpropagation.

According to [24], for this adversarial training, the objective function for the TV show classifier L_{dom} is defined as:

$$L_{dom}(\theta_x, \theta_z) = - \sum_{i=1}^N \log P(z_i | x_i; \theta_x, \theta_z) \tag{2}$$

The DNN acoustic model and the adversarial branch were jointly trained to optimize the following:

$$\min_{\theta_x, \theta_y} \max_{\theta_z} L_{MMI}(\theta_x, \theta_y) - \lambda L_{dom}(\theta_x, \theta_z), \tag{3}$$

where λ is a trade-off parameter between the pdf classification loss L_{MMI} , which corresponds to the LF-MMI loss defined in Equation (1), and the domain loss L_{dom} , related to the TV show classification task, which aims to make deep acoustic features invariant to the domain of the TV show characteristics.

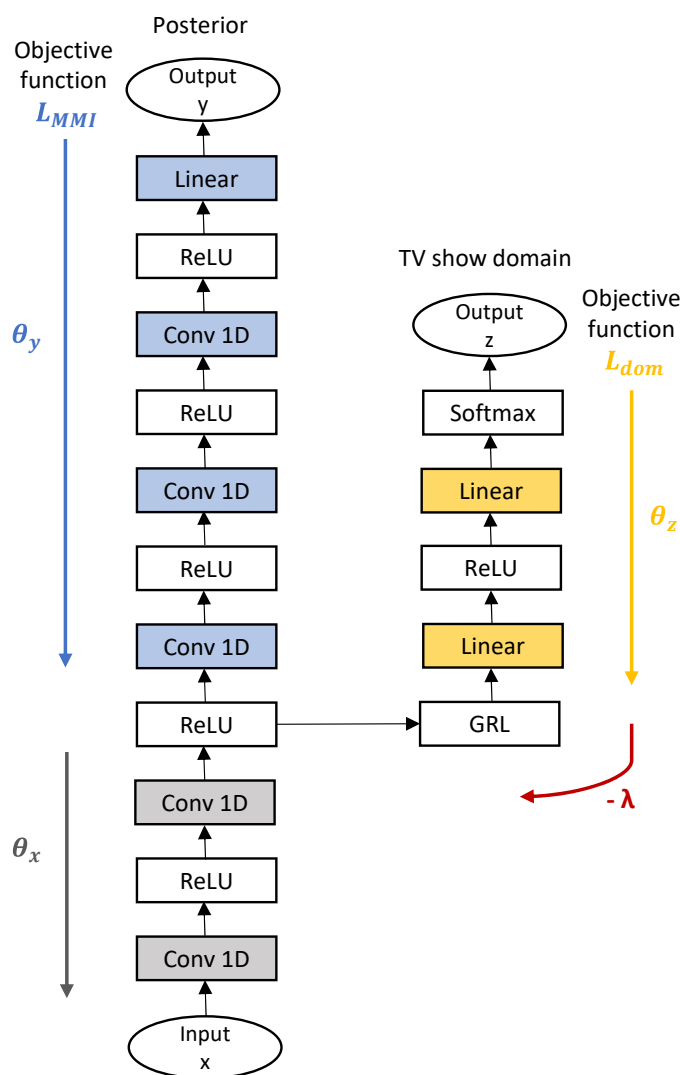


Figure 1. Architecture of the end-to-end LF-MMI approach applying DAT. An adversarial branch (TV show classifier) is added to the second layer of the main PyChain architecture (posterior classifier).

3. Experimental Setup

3.1. RTVE2020 Database

The proposed ASR systems were evaluated under the IberSpeech-RTVE 2020 Challenge conditions. The RTVE2020 Database [38] was provided to the participants. This is an extension of the RTVE2018 Database, which contains a collection of Spanish TV shows and

broadcast news from 2015 to 2019. The training partition consists of audio files, partially subtitled, presenting the following limitations:

- Subtitles were generated by means of a re-speaking procedure that sometimes changed the sentences and summarized what had been said, obtaining non-reliable transcriptions;
- Transcriptions were not supervised by humans. Only 109 h from the dev1, dev2, and test partitions contain human-revised transcriptions;
- Timestamps were not properly aligned with the speech signal.

Regarding all these limitations, we tried to avoid the use of these low-quality transcriptions, which could poorly model the acoustic space. As shown in Figure 2, we carried out a semi-supervised annotation process, which allowed training accurate acoustic models. First, a baseline acoustic model was trained using our own databases (see Section 3.2). Once the acoustic model was prepared, the unlabeled speech data were initially aligned to obtain a provisional transcription. To improve the quality of these automatic transcriptions, a human annotator team was responsible for the supervision process.

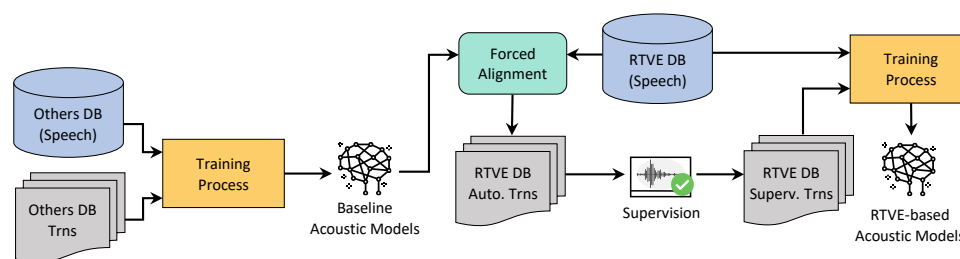


Figure 2. Entire process of obtaining high-quality transcriptions to train the acoustic models.

The RTVE training partition was prepared under this supervision process to obtain reliable transcriptions aligned with the speech signal. Hereafter, we were able to develop our first ASR models based on TV content [32].

Due to some limitations during the supervision process, two resulting datasets were used for training the systems: RTVE_train350 (350 h from RTVE training set) and RTVE_train100 (100 h from RTVE_train350). The validation datasets were 20% of the training data. It is worth noting that we tried to balance the partitions at any time, trying to cover the different scenarios represented in the whole RTVE dataset, such as political and economic news, in-depth interviews, debate and live magazines, among others. Consequently, for testing purposes, several datasets corresponding to 1 h in duration each were built from the RTVE_dev1 and RTVE_dev2 development partitions.

Finally, the RTVE2020 database was completed adding a collection of TV shows that belong to a wide range of genres and broadcasts from 2018 to 2019. This was composed of 70.3 h of human transcribed audio. It was used as the test partition for the Speech-to-Text Transcription Challenge (RTVE2020_test).

3.2. Other Databases

Additional datasets were added to train the system in an open training condition scenario:

The VESLIM database consists of 103 h of clean Spanish voice, where the speakers read a set of sentences. More details are in Ref. [39].

OWNMEDIA is composed of 162 h of TV shows, interviews, lectures, and several multimedia contents. It was used for training the baseline acoustic model, which allowed the initial alignment of the unlabeled speech data.

Finally, data augmentation techniques related to the hybrid ASR system were carried out by means of the reverberation database (<http://www.openslr.org/28/>, accessed on 14 January 2022), which was described in Section 2.1.

3.3. Training Setup

The acoustic model of the hybrid ASR system was trained using the RTVE_train350, VESLIM, and OWNMEDIA databases, following the SWBD Kaldi recipe for chain models. Some modifications were included following the ASpIRE recipe for multi-condition tasks.

Otherwise, end-to-end LF-MMI models without DAT were trained by using only RTVE_train100. This relatively small amount of data allowed a light training process to test the system performance. We used PyChain-example (https://github.com/YiwenShaoStephen/pychain_example, accessed on 14 January 2022) as a reference by adding some changes in terms of data loading and data parallelization to use more than one GPU.

The data preparation was carried out in Kaldi. To convert input features into PyTorch tensors, we used kaldi_io (<https://github.com/vesis84/kaldi-io-for-python>, accessed on 14 January 2022), as suggested in Ref. [18]. For the adversarial training, note that the number of pdf posteriors was $y_i = 62$, corresponding to the senones, and the number of TV shows was $z_i = 13$ because of the different TV shows that the RTVE_train100 partition contains. To reduce the bias effect due to unbalanced classes in TV shows at training time, the data were previously merged according to their acoustic characteristics. As a result, four new groups were defined (see also Table 1):

1. Live TV shows: a variety of content for the whole family;
2. Documentaries: show broadcasts about risky sports, adventure, street reports, and current information in different Spanish regions;
3. TV game shows: content related to comedy competitions, road safety, or culture dissemination, among others;
4. Interviews: moderated debates with analysis, political and economic news, and weather information.

Table 1. Description of the domain classes according to the number of samples and the characteristics of the TV shows. More details related to the TV shows are described in the RTVE2020 Database specifications [38].

Class	# of Samples	Examples of TV Shows
1. live TV shows	11,239	La Mañana
2. documentaries	4671	Al filo de lo Imposible, Comando Actualidad, España en Comunidad
3. TV-game shows	7995	Arranca en Verde, Dicho y Hecho, Saber y Ganar
4. interviews	22,194	Latinoamerica 24H, La Tarde en 24H, Millenium

As a consequence, the labels of the training data for the adversarial branch (i.e., the TV show classifier sub-network) are defined as $z_i = 0, 1, 2, 3$. In the adversarial architecture, the second hidden layer of the TDNN was used as the input to the adversarial branch, which consisted of a dense layer of size 384 and the ReLU activation function, followed by a softmax output layer, whose output dimension corresponded to the number of TV shows (i.e., 4). The cross-entropy loss function was used in the adversarial training. To select the optimal trade-off parameter λ , several values were tested. The best performance was achieved for $\lambda = 0.041$. In addition, all the systems were evaluated with the same 3-gram LM. As described in Ref. [32], it was trained on several corpora: subtitles provided in the RTVE2018 Database, supervised transcriptions, news between 2015 and 2018, interviews, and file captions.

3.4. Resources

Several computational resources were required to carry out this work. A server with 2 Xeon E5-2630V4, 2.2 GHz, 10C/20 TH, and 3 GPUs Nvidia GTX 1080 Ti was used for the hybrid ASR system. GPU calculation was necessary for the DNN stage, and only CPU mode was used for the HMM stage and final decoding.

4. Results

4.1. Hybrid ASR

The proposal of applying data augmentation techniques to improve the hybrid ASR performance was fulfilled. The addition of reverberation to our whole training dataset (over 600 h of speech) improved the performance in most of the scenarios represented by every TV show set. As shown in Table 2, the model applied to the Comando Actualidad (CA) dataset achieved a relative improvement of around 10%, as compared to the baseline system. This might be due to the trained model having learned to model these speech artifacts that can appear in the challenging scenarios described in Section 2.1. However, the improvements in the rest of the TV shows were not so remarkable (e.g., 20H or LM) because the contents were related to daily news with more favorable acoustic conditions.

As we mentioned in Ref. [32], the reference master of the transcriptions was not reviewed. As usual, we evaluated the possible impact of transcription errors by means of a new test using an external dataset. It consisted of 3.5 h of TV news broadcasts (similar to 20H). Applying the reverb-trained models of our Kaldi-based hybrid system, we reduced the WER from 8.51% to 7.96%, being our new best results achieved so far. Table 3 shows that data augmentation also maintained the WER improvement of around 10% relative on the RTVE2020 test partition.

Table 2. WER (%) on the different datasets for hybrid and end-to-end ASR systems. Each one of the evaluation datasets contains one hour of speech. In bold, the improvements of the Kaldi-based system after applying reverberated data augmentation, and the improvements related to the Pychain-based system after applying DAT.

	20H_dev1	AP_dev1	CA_dev1	LM_dev1	Mill_dev1	LN24H_dev1
Hybrid ASR						
Kaldi-based baseline [32]	14.88	20.94	49.55	21.44	17.01	24.13
Reverb. data augmentation	14.76	21.00	44.69	21.03	16.42	23.62
Kaldi-based baseline (RTVE_train100)	16.09	22.32	51.23	23.02	17.70	25.53
End-to-end LF-MMI ASR						
PyChain-based baseline	23.66	33.31	59.34	29.95	35.09	25.08
Domain adversarial training	23.53	32.99	59.25	29.91	34.67	25.16

Table 3. WER (%) on the RTVE2020 test partition for all the systems. Results were obtained after the submission.

	RTVE2020_test
Hybrid ASR	
Kaldi-based baseline [32]	31.01
Reverb. data augmentation	27.68
End-to-end LF-MMI ASR	
PyChain-based baseline	40.90
Domain adversarial training	42.89

4.2. End-to-End LF-MMI ASR

Our PyChain-based baseline had a good performance in relation to the number of parameters and the easier training process compared to other end-to-end frameworks. The WER achieved for standard TV news (e.g., 20H, LN24H) was between 23% and 26%, as Table 2 shows. These results are within the expected range where commercial ASR systems operate.

To compare both hybrid and end-to-end systems, we also trained a hybrid model by using only the RTVE_train100 partition. In this case, multi-condition data augmentation was not applied. The PyChain-based system was still far from the Kaldi-based hybrid system, with a WER increase of 17% in the worst-case scenario. This gap could be reduced with the application of some data augmentation techniques (e.g., speed or volume pertur-

bation). Despite the fact that augmented data caused a slight improvement of the WER for the PyChain system [18], TV shows can contain some acoustic characteristics that could be better modeled by some audio perturbations.

On the other hand, we evaluated the effect of learning acoustic representations invariant to the TV show domain. After applying DAT, the results in Table 2 showed improvements, in terms of the WER, up to 2.87% as compared to the PyChain-based baseline. We are aware that those results are still far from the Kaldi-based hybrid ASR system based on the DNN-HMM. Nevertheless, the results in Table 2 gave us an insight into how the use of DAT in the end-to-end LF-MMI model can improve its performance in most of the scenarios. Furthermore, it seemed that DAT was able to generate deep acoustic features invariant to different TV shows with different acoustic conditions without the need for data augmentation techniques.

In addition, Table 3 shows that applying DAT did not reduce the WER on the RTVE2020 test partition. The main reason was DAT alleviated the labeled domain conditions in the training dataset. Thus, invariant features were trained without regarding these unseen external factors.

Finally, regarding computational requirements for speech transcription, we considered that PyChain carries out two main stages: a first decoding stage and a second four-gram rescoring stage. Real-Time factors (RT) for different test partitions are presented in Table 4. In all cases, the two GPU resources described in Section 3.4 were used. The results in Table 4 show that time requirements depend on the characteristics of the test partitions. Less time is required to transcribe the best acoustic conditions and less challenging scenarios. This makes sense as far as the confusion of the graph model is less complex to transcribe accurately.

Table 4. Real-Time factor (RT) for the different stages carried out in the PyChain-based baseline according to the different datasets.

Datasets	Decoding	LM Rescoring
20H_dev1	0.033	0.115
AP_dev1	0.035	0.175
CA_dev1	0.225	1.976
LM_dev1	0.092	0.450
Mill_dev1	0.082	0.442
LN24H_dev1	0.383	0.148

5. Related Works

As the evaluation of our systems was carried out under the conditions of the IberSpeech-RTVE 2020 Challenge [31], we can now compare our work to other ASR systems participating in this challenge and thus also trained on this specific domain related to TV programs. As a first general comment, we can say that all the results for the developed ASR systems showed that end-to-end systems are still far from hybrid systems in the challenging conditions of RTVE 2020. Kocour et al. [40] developed an end-to-end system based on the wav2letter architecture [7], which was not able to generalize very well on the acoustic conditions of the RTVE2020 database, reporting a WER of at least 13% higher than their best hybrid ASR system. Álvarez et al. [41] presented a Quartznet-based [42] ASR implementation showing promising results due to the use of more than a hundred hours of speech data. Nevertheless, the results in terms of the WER from that system were 9% worse when compared to other hybrid systems they developed for the challenge.

Furthermore, the acoustic conditions of databases provided in other international challenges, such as CHiME-6 [14], have also been responsible for the poor performance of end-to-end ASR systems. Different approaches based on RNNs and transformers, along with the RNN-T and CTC-Attention [15] have been evaluated on one of the CHiME partitions, concluding that speech-enhancement techniques contribute to the reduction of to the gap between end-to-end and hybrid systems. The difficulty of obtaining high-

quality labeled speech leads to emergent machine-learning-based paradigms, such as Self-Supervised Learning (SSL), whose goal for ASR is to learn powerful speech representations from unlabeled examples (e.g., wav2vec 2.0 [43]). Other recent works have tried to mitigate the effect of limited training data [44] or noisy environment conditions [45]. In the case of [44], the use of CTC and end-to-end LF-MMI to fine-tune a wav2vec 2.0 model showed similar performance even for out-of-domain and cross-lingual adaptation. Regarding [45], the authors integrated SSL with contrastive learning from original–noisy speech pairs to model representations with noise robustness.

Along the same lines, we proposed the application of DAT in our baseline end-to-end ASR systems as an alternative to obtain robust features invariant to the domain, i.e., different acoustic conditions of TV shows. It is a fact that DAT is beneficial for building robust embeddings. In recent works, it has been even integrated with wav2vec embeddings to have an accent-robust speech recognition [46]. The authors reported good results when no accent labels were available for training.

6. Conclusions and Future Work

In this paper, we developed both hybrid and end-to-end ASR approaches exploring some techniques to improve the performance of Speech-to-Text tasks under IberSpeech-RTVE 2020 Challenge. We showed that Hybrid DNN-HMMs can be adapted to the TV show domain by means of multi-condition data augmentation. The addition of reverberated data to the training data decreased the WER significantly (10% relative). A WER of 7.96% was achieved in better conditions. We demonstrated that the lack of data augmentation techniques could be the main reason of the gap between the Kaldi-based hybrid system and PyChain-based system. Moreover, a higher volume of data used to train the end-to-end system could contribute to increasing the ASR performance. However, other easy-to-apply techniques, such as DAT, can overcome this gap, yielding improvements in end-to-end ASR systems. We found that using DAT, acoustic features invariant to different TV domains can be learned, achieving a WER improvement of 2.87%.

The IberSpeech-RTVE 2020 Challenge has provided some findings pointing out that end-to-end approaches are close to being competitive (between 28% and 40% WER) by using more than 600 h of speech. However, they are still far from the hybrid models.

As future work, besides data augmentation, we believe that exploring speech-enhancement techniques could help to close the performance gap between hybrid and end-to-end systems. In addition, unsupervised machine learning methods (e.g., clustering) or automatic perceptual speech quality methods (e.g., PESQ) could contribute to a more accurate TV show classification prior to DAT. Finally, we also believe that the combination of DAT with a self-supervised approach could be useful to achieve significant improvements in ASR systems through robust embeddings trained even without supervised data.

Author Contributions: Conceptualization, J.M.P.-C., F.M.E.-C. and L.A.H.-G.; data curation, J.M.P.-C.; formal analysis, J.M.P.-C., F.M.E.-C. and L.A.H.-G.; investigation, J.M.P.-C., F.M.E.-C. and L.A.H.-G.; methodology, J.M.P.-C.; software, J.M.P.-C. and F.M.E.-C.; supervision, F.M.E.-C. and L.A.H.-G.; validation, J.M.P.-C. and F.M.E.-C.; writing–original draft, J.M.P.-C.; writing–review & editing, F.M.E.-C. and L.A.H.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: RTVE 2020 database was used under a license agreement. It is available upon request in <http://catedrartve.unizar.es/rtvedatabase.html>, accessed on 14 January 2022.

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

References

1. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
2. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Waikoloa, HI, USA, 11–15 December 2011.
3. Graves, A. Sequence transduction with recurrent neural networks. *arXiv* **2012**, arXiv:1211.3711.
4. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
5. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964.
6. Chiu, C.C.; Sainath, T.N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z.; Kannan, A.; Weiss, R.J.; Rao, K.; Gonina, E.; et al. State-of-the-art speech recognition with sequence-to-sequence models. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4774–4778.
7. Collobert, R.; Puhresch, C.; Synnaeve, G. Wav2letter: An end-to-end convnet-based speech recognition system. *arXiv* **2016**, arXiv:1609.03193.
8. Zeyer, A.; Irie, K.; Schlüter, R.; Ney, H. Improved training of end-to-end attention models for speech recognition. *arXiv* **2018**, arXiv:1805.03294.
9. Bahdanau, D.; Chorowski, J.; Serdyuk, D.; Brakel, P.; Bengio, Y. End-to-end attention-based large vocabulary speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4945–4949.
10. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
11. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
12. Zhang, Y.; Qin, J.; Park, D.S.; Han, W.; Chiu, C.C.; Pang, R.; Le, Q.V.; Wu, Y. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv* **2020**, arXiv:2010.10504.
13. Chen, G.; Chai, S.; Wang, G.; Du, J.; Zhang, W.Q.; Weng, C.; Su, D.; Povey, D.; Trmal, J.; Zhang, J.; et al. GigaSpeech: An Evolving, Multi-domain ASR Corpus with 10,000 h of Transcribed Audio. *arXiv* **2021**, arXiv:2106.06909.
14. Watanabe, S.; Mandel, M.; Barker, J.; Vincent, E.; Arora, A.; Chang, X.; Khudanpur, S.; Manohar, V.; Povey, D.; Raj, D.; et al. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. *arXiv* **2020**, arXiv:2004.09249.
15. Andrusenko, A.; Laptev, A.; Medennikov, I. Towards a competitive end-to-end speech recognition for chime-6 dinner party transcription. *arXiv* **2020**, arXiv:2004.10799.
16. Chan, W.; Park, D.; Lee, C.; Zhang, Y.; Le, Q.; Norouzi, M. SpeechStew: Simply mix all available speech recognition data to train one large neural network. *arXiv* **2021**, arXiv:2104.02133.
17. Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Soplin, N.E.Y.; Heymann, J.; Wiesner, M.; Chen, N.; et al. Espnet: End-to-end speech processing toolkit. *arXiv* **2018**, arXiv:1804.00015.
18. Shao, Y.; Wang, Y.; Povey, D.; Khudanpur, S. PyChain: A Fully Parallelized PyTorch Implementation of LF-MMI for End-to-End ASR. *arXiv* **2020**, arXiv:2005.09824.
19. Povey, D.; Peddinti, V.; Galvez, D.; Ghahremani, P.; Manohar, V.; Na, X.; Wang, Y.; Khudanpur, S. *Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI*; Interspeech: San Francisco, CA, USA, 2016; pp. 2751–2755.
20. Peddinti, V.; Chen, G.; Manohar, V.; Ko, T.; Povey, D.; Khudanpur, S. *JHU ASPIRE System: Robust LVCSR with TDNNS, iVector Adaptation and RNN-LMS*; In Proceedings of the IEEE 2015 Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; pp. 539–546.
21. Ko, T.; Peddinti, V.; Povey, D.; Khudanpur, S. Audio augmentation for speech recognition. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
22. Han, W.; Zhang, Z.; Zhang, Y.; Yu, J.; Chiu, C.C.; Qin, J.; Gulati, A.; Pang, R.; Wu, Y. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *arXiv* **2020**, arXiv:2005.03191.
23. Winata, G.I.; Cahyawijaya, S.; Liu, Z.; Lin, Z.; Madotto, A.; Xu, P.; Fung, P. Learning Fast Adaptation on Cross-Accented Speech Recognition. *arXiv* **2020**, arXiv:eess.AS/2003.01901.
24. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
25. Dhakal, P.; Damacharla, P.; Javaid, A.Y.; Devabhaktuni, V. A near real-time automatic speaker recognition architecture for voice-based user interface. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 504–520. [CrossRef]

26. Feng, S.; Kudina, O.; Halpern, B.M.; Scharenborg, O. Quantifying bias in automatic speech recognition. *arXiv* **2021**, arXiv:2103.15122.
27. Serdyuk, D.; Audhkhasi, K.; Brakel, P.; Ramabhadran, B.; Thomas, S.; Bengio, Y. Invariant representations for noisy speech recognition. *arXiv* **2016**, arXiv:1612.01928.
28. Shinohara, Y. *Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition*; Interspeech: San Francisco, CA, USA, 2016; pp. 2369–2372.
29. Sun, S.; Yeh, C.F.; Hwang, M.Y.; Ostendorf, M.; Xie, L. Domain adversarial training for accented speech recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4854–4858.
30. Meng, Z.; Li, J.; Chen, Z.; Zhao, Y.; Mazalov, V.; Gang, Y.; Juang, B.H. Speaker-invariant training via adversarial learning. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5969–5973.
31. Lleida, E.; Ortega, A.; Miguel, A.; Bazán-Gil, V.; Pérez, C.; Gómez, M.; de Prada, A. Albayzin Evaluation: IberSPEECH-RTVE 2020 Speech to Text Transcription Challenge. 2020. Available online: <http://catedrartve.unizar.es/reto2020/EvalPlan-S2T-2020-v1.pdf> (accessed on 14 January 2022).
32. Perero-Codosero, J.M.; Antón-Martín, J.; Merino, D.T.; Gonzalo, E.L.; Gómez, L.A.H. *Exploring Open-Source Deep Learning ASR for Speech-to-Text TV Program Transcription*; IberSPEECH: Valladolid, Spain, 2018; pp. 262–266.
33. Lleida, E.; Ortega, A.; Miguel, A.; Bazán-Gil, 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. [\[CrossRef\]](#)
34. Peddinti, V.; Povey, D.; Khudanpur, S. A time delay neural network architecture for efficient modeling of long temporal contexts. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
35. Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M.L.; Khudanpur, S. A study on data augmentation of reverberant speech for robust 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. 5220–5224.
36. Ravanelli, M.; Parcollet, T.; Bengio, Y. The pytorch-kaldi speech recognition toolkit. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6465–6469.
37. Can, D.; Martinez, V.R.; Papadopoulos, P.; Narayanan, S.S. Pykaldi: A python wrapper for kaldi. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5889–5893.
38. Lleida, E.; Ortega, A.; Miguel, A.; Bazán-Gil, V.; Pérez, C.; Gómez, M.; de Prada, A. RTVE2020 Database Description. 2020. Available online: <http://catedrartve.unizar.es/reto2020/RTVE2020DB.pdf> (accessed on 14 January 2022).
39. Toledano, D.T.; Gómez, L.A.H.; Grande, L.V. Automatic phonetic segmentation. *IEEE Trans. Speech Audio Process.* **2003**, *11*, 617–625. [\[CrossRef\]](#)
40. 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. *arXiv* **2021**, arXiv:2101.12729.
41. Alvarez, A.; Arzelus, H.; Torre, I.G.; González-Docasal, A. *The Vicomtech Speech Transcription Systems for the Albayzin-RTVE 2020 Speech to Text Transcription Challenge*; IberSPEECH: Virtual Valladolid, Spain, 2021; pp. 104–107.
42. Kriman, S.; Beliaev, S.; Ginsburg, B.; Huang, J.; Kuchaiev, O.; Lavrukhin, V.; Leary, R.; Li, J.; Zhang, Y. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual Barcelona, Spain, 4–8 May 2020; pp. 6124–6128.
43. Baevski, A.; Zhou, H.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv* **2020**, arXiv:2006.11477.
44. Vyas, A.; Madikeri, S.; Bourlard, H. Comparing CTC and LFMMI for out-of-domain adaptation of wav2vec 2.0 acoustic model. *arXiv* **2021**, arXiv:2104.02558.
45. Wang, Y.; Li, J.; Wang, H.; Qian, Y.; Wang, C.; Wu, Y. Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. *arXiv* **2021**, arXiv:2110.04934.
46. Li, J.; Manohar, V.; Chitkara, P.; Tjandra, A.; Picheny, M.; Zhang, F.; Zhang, X.; Saraf, Y. Accent-Robust Automatic Speech Recognition Using Supervised and Unsupervised Wav2vec Embeddings. *arXiv* **2021**, arXiv:2110.03520.