DarSpeech: An Automatic Speech Recognition System for the Moroccan Dialect

Omar Aitoulghazi
School of Science and Engineering
Al Akhawayn University
Ifrane, Morocco
o.aitoulghazi@aui.ma

Ahmed Jaafari
School of Science and Engineering
Al Akhawayn University
Ifrane, Morocco
a.jaafari@aui.ma

Asmaa Mourhir
School of Science and Engineering
Al Akhawayn University
Ifrane, Morocco
a.mourhir@aui.ma

2022 International Conference on Intelligent Systems and Computer Vision (ISCV) | 978-1-6654-9558-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCV54655.2022.9806105

Abstract—**Due to the continuous increase of information and data, it has been proven that Automatic Speech Recognition (ASR) systems are more efficient and less expensive when it comes to a variety of important tasks, such as customer relationship management. However, the most complex and accurate speech recognition models are developed and implemented for languages in which data is highly available, such as English and French. This document proposes an automatic speech recognition system for the Moroccan dialect, a very low-resource language, that is spoken by almost every Moroccan citizen and adopted in many organizations that are both public and private. The proposed solution is based on a state-of-the-art architecture, named Deep Speech 2 by Baidu. We tested the model on 24 hours of speech and obtained 22.7% word error rate and 6.03% character error rate.**

Keywords—automatic speech recognition, low-resource language, Moroccan dialect, Deep Speech 2

I. INTRODUCTION

The importance of automatic speech recognition has motivated research to give birth to many approaches that are both statistical and probabilistic. However, with the emergence of big data and the increase in demand of Natural Language Processing (NLP) applications, research has shown that computational neural networks, as a machine learning technique, prove to be more efficient and better suited for such tasks [4, 8, 13, 14]. First, artificial neural networks are language agnostic, meaning that they do not necessarily depend on specific languages. In NLP tasks, we are usually dealing with unstructured data such as text, audio or images. Deep learning makes better sense of unstructured data, through its various architectures like Recurrent Neural Networks (RNNs) for sequence data like text and audio, and Convolutional Neural Networks (CNNs) that are suitable for processing any grid-like data. Moreover, artificial neural networks do not require hand engineered features, as they build hierarchical representations of features during the training; the learning of better features is achieved by updating the networks' weights through a number of epochs, using an algorithm like stochastic gradient descent. Also, with the huge amounts of data that is available today, neural networks are able to achieve state of the art results in many different NLP tasks. However, complex neural networks can be computationally expensive to train; fortunately, and because of how promising this machine learning technique is, many affordable, computational resources are made available through a variety of cloud computing services, that provide different options for GPU and TPU usage, to train deep learning models.

A lot of work has been done to build automatic speech recognition systems for different languages, especially the most spoken ones such as English, Mandarin and French, for which data is highly available [8, 13, 14].

Moroccan dialect (Darija), which is mainly derived from standard Arabic, in addition to other languages (French, Spanish and Tamazight), is spoken daily by millions of Moroccans every day; it is also adopted in many businesses and organizations to communicate with customers and users. However, the presence of this dialect in technology is really low, almost non-existent, as it is assumed that the Moroccan population is fluent in either standard Arabic or French, which is in many cases a wrong assumption. Thus, the purpose of this work is to provide ground research for speech recognition in Moroccan dialect which happens to be one of the most poorly resourced dialects.

II. RELATED WORK

Speech recognition has known a great start with the Hidden Markov Models. A Hidden Markov Model (HMM) is a natural and highly robust statistical technique for automatic speech recognition. The parameters of HMMs have been essential in describing the behavior of the utterance of speech segments [1]. HMMs were always supported by the Gaussian Mixture Model (GMM) and Mel Frequency Cepstral Coefficients (MFCCs). Using a GMM for clean speech and a mean vector for noise, the GMM estimates the expectation of the mismatch factor between clean and noisy speech at each step [1]. Such models were only tested on small sequences of speech (e.g., words), and they provided good results at the first stages of automatic speech recognition.

Nevertheless, ASR systems have known great enhancement with artificial neural networks and the variety of architectures they come with. For example, Longfei Li et al. [2] have been able to combine the Markovian model and deep neural networks (DNN) to give birth to a hybrid DNN-HMM model that is able to achieve good speech-to-text results with the support of a Gaussian Mixture Model (GMM-HMM). In another study, that was conducted by Facebook AI [4], a fully ASR system for English was built using deep bi-directional recurrent neural networks with skip connections and the connectionist temporal classification training method that allows to train RNNs for sequence labelling where the input-output characters' alignment is unknown. A similar study [5] enhanced the previous architecture by implementing the attention component that optimizes such models and makes them able to handle longer sequences of text/speech. Moreover, the authors of [3] tackled the need for a huge amount of transcribed data to perform speech recognition using a method, named Masked Predictive Coding (MPC), for pre-training. The training in their work is composed of two steps: a pre-training step on unsupervised data and a fine-tuning stage on supervised data. Once the unsupervised pre-training is done, the MPC layer is removed, and a transformer decoder is plugged for fine-tuning. The MPC method is inspired by the Bidirectional Encoder Representations from Transformers (BERT), and it performs predictive coding on Transformer-based models using a Masked Language Model like structure.

Today, we can talk about even more advanced, complex models that make use of Transformers and even Conformers (a combination of the convolutional operation and transformers [22]). Wav2vec [6, 7], which is considered a state of the art model for automatic speech recognition nowadays, is built based on of the most complex architectures, which requires huge amounts of resources to train, combining CNNs with Transformers. This model allows training ASR systems with raw speech data that is not necessarily annotated, using self-supervised learning to generate speech representation. In other words, a model is first trained to generate transcriptions and label raw speech audio data that would be used to train a second model for ASR. Furthermore, and with the evolution of CNNs in natural language processing (NLP), especially in speech recognition, the Google authors of [8] suggest a new solution that combines both the use of a CNN and the transformer based RNN architecture to give birth to a CNN-RNN transducer architecture, that is called ContextNet. The proposed architecture includes a fully convolutional encoder that adds squeeze-and-excitation modules to convolution layers to integrate global context information. The proposed solution achieves a good trade-off between the computational cost and accuracy. Hence, the paper's key contributions are an upgraded CNN architecture for ASR with global context and a progressive downsampling/model scaling approach to achieve great accuracy and model size trade-off.

Research efforts were not only put into the widely spoken languages like English. Many studies have been tackling the challenges of developing robust ASR systems for low-resource languages and dialects. For instance, in Zhao Yue et al. [9], the authors propose a simple model that tackles speech content recognition and dialect classification for the Tibetan language. The proposed solution makes use of the WaveNet architecture [10], which is a deep neural network that is used for generating raw audio waveforms, the authors used WaveNet to model the distribution of speech data from different speakers and dialects. Also, Xu Jin et al. [11] worked on an ASR model for an extremely low-resource language that was tested on the Lithuanian language. This work consists of three main steps, which are pre-training on a rich language like English, fine-tuning on low-resource languages, dual transformation between text-to-speech and ASR to enhance the accuracy of each other. Another study [12] explored the conformer architecture to train an ASR model for Indian. The model consists of one conformer encoder and two parallel transformer decoders (a phoneme decoder and a grapheme one), the neural network is then optimized using Joint CTC-Attention training; the results of this model outperformed many models that existed for Indian dialects.

In this work, we opt for Deep Speech 2 [13], which is an enhanced version of Deep Speech [14] that makes use of a CNN to extract features, mainly because of its simplicity and flexibility in terms of scalability and maintainability; the architecture can easily be optimized and fine-tuned, and it also allows the merging of a language model and the acoustic model, supported by beam search [13, 15, 16], which enhances the ASR system performance. The architecture of Deep Speech 2, and more specifically the first version of Deep Speech [14], has been adopted to implement various ASR systems for low-resource languages. Two studies that are similar to ours have already proven the good performance that the Deep Speech architecture achieves. Namely, the authors in [23] were able to achieve a word error rate of 6%, adopting this architecture for a Tibetan automatic speech recognition system. Additionally, Messaoudi Abir et al. [21] implemented an ASR system for the Tunisian dialect based on the first version of Deep Speech, achieving a word error rate of 24.4%.

III. Materials and Methods

A. Deep Speech 2: Overall Architecture

The general idea behind Deep Speech 2 [13] is ingesting spectrograms to a number of convolutional layers to extract features and feed them to a number of consecutive RNN layers that leverage those features for the CTC decoder to generate text transcriptions. An utterance or a speech/audio file is a time-series of a certain duration, with each time-slice containing a vector of audio features.

The RNN part consists of several bidirectional recurrent layers, followed by one or more fully connected layers, followed by the output layer which is a Softmax that computes the probability distribution over the characters in the language's vocabulary. The model is trained using the CTC loss function. The CTC algorithm, briefly explained in the next section, helps with identifying alignments between inputs and outputs [17].

To be able to scale the Deep Speech 2 model as the training set increases, the depth of the network is increased by adding more hidden layers instead of making the existing layers larger and more complex, meaning more bidirectional recurrent layers are added. However, adding more layers make the model prone to the problem of covariate shift which reduces the generalization performance as the activations in previous layers keep changing all the time. For this issue, DeepSpeech 2 uses one of the most exciting innovations in optimizing deep neural networks which is Batch Normalization (BN) [18]. BN applies the normalization process even for the hidden layers in the network, which enhances the speed of convergence and helps with regularization.

B. Connectionist Temporal Classification (CTC)

The Connectionist Temporal Classification (CTC) [17] network includes a Softmax output layer in which the activations of the given labels are interpreted as the probabilities of observing the corresponding labels at different, specific time steps, as well as an extra unit whose activation represents the probability of observing a blank/no label.

The trick that CTC uses to tackle the problem of silences and repeated characters is inserting a blank character; this blank character should separate the repeated characters in the actual transcript, and the CTC operation explores all the paths or alignments, using the vocabulary tokens that include the blank character, that can lead to the actual ground truth transcript. The CTC gets rid of any repeated characters that are not separated using the blank characters and keeps all the other different tokens in addition to only one token in the repeated sequence that is not separated by the blank character (e.g., only one H and one O would be kept in a generated sequence like "HHH-E-L-L-OOO" for the label "Hello"). The CTC scores every path from the different Softmax vectors that are generated in every time step; this scoring is done by considering all the provided combinations of characters provided in the vectors of probabilities, and the score of one path is the product of the probabilities that constructs it as shown in Eq (1). Moreover, to reduce the number of considered paths, all the paths that start with a token, that is neither a blank character nor the first character in the actual transcript, are ignored.

$$P(C\,|x) = \prod_{i=1}^{N} P(c_i|x) \tag{1}$$

where C is our sequence "HHH-E-L-L-OOO", c_i is every token that constructs that path/sequence, and N is the total number of tokens constructing that path.

Then, an addition of all the scores of paths that yield the same transcript is performed; The CTC loss is the negative logarithm of that.

In fact, the number of different paths that can be generated is exponential which makes it computationally hard to explore all the paths using a brute force algorithm. Hence, the authors of [17] made use of an algorithm that is based on dynamic programming, called Forward-Backward algorithm. So, instead of only selecting the most likely alignment, the expectation over all possible alignments is generated during training which allows to exploit the existence of various sub-paths in the graph of vectors.

The CTC is a good approach for languages, such as the Moroccan dialect, which do not have a clear linguistic structure. It predicts sequences of characters that yield the ground truth transcripts by using a moving window that explores the feature maps which are generated by the CNN.

IV. Proposed Model

A. Dataset

To create the speech corpus presented in this paper, we collected several sentences in Darija as part of a previous work for learning word embedding in Darija [19]. We chose to scrape the needed text from a website that publishes stories in Darija (with Arabic script). We have gathered approximately 377MB of text in Darija, which represents approximately 40 million words. We have used a portion of the text scraped, chosen carefully, to build our Dataset. Then, we segmented the sentences at the level of the period. We mostly selected sentences with dialogues between different characters to make our speech corpus more vivid and expressive which can be useful in tasks such as speech sentiment classification. Fig. 1 shows a sample of the data.

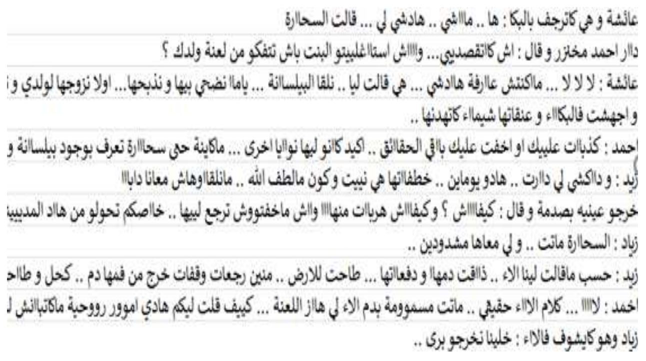
عائشة و هي كاترجف بالبكا : ها .. مااشي .. هادشي لي ... قالت السحارة
دار احمد مخازر و قال : اش كاتقصديي... وااااش استاغليتو البنت باش تتفكو من لعنة ولدك ؟
عائشة : لا لا لا ... ماكنتش عاارفة هادشي ... هي قالت ليا .. نلقا البيلسانة ... ياما نضحي بيها و نذبحها... اولا نزوجها لولدي و
و اجهشت فالبكااء و عنقاتها شيماء كاتهدنها ..
احمد : كذبات عليك او اخفت عليك بااقي الحقائق .. اكيد كانو ليها نواايا اخرى ... ماكاينة حتى سحاارة تعرف بوجود بيلسانة و
زيد : و دااكشي لي داارت .. هادو يومين .. خطفااتها هي نبيت وكون مالطف الله .. مانلقااوهاش معانا داباا
خرجو عينيه بصدمة و قال : كيفاااش ؟ وكيفاااش هربات منهاا واش ماخفتووش ترجع ليها .. خاصكم تحولو من هاد المدي
زياد : السحاارة ماتت .. و لي معاها مشدودين ..
زيد : حسب ماقالت لينا الاء .. ذااقت دمهاا و دفعااتها ... طاحت للارض .. منين رجعات وقفات خرج من فمها دم .. كحل و طاح
احمد : لاااا ... كلام الااء حقيقي .. ماتت مسموومة بدم الاء لي هااز اللعنة ... كيف قلت ليكم هادي امور رووحية ماكاتبانش ل
زياد وهو كايشوف فالااء : خلينا نخرجو برى ..

Fig. 1. Sample of the Annotated Transcriptions

We built the Annrabic Annotation Tool (AAT) whose main objective is to facilitate the recording task for the annotators. This helped speed up the process of annotation by almost 10 times. AAT was greatly inspired by the Common Voice tool. Fig. 2 shows a snapshot of the recording interface of AAT.

To provide audio clips for the sentences in our dataset, we hired 38 annotators to do the task. This established diversity and heterogeneity in our speech corpus which makes it suitable for a wide range of NLP applications as the dataset will help avoid high variance. All the annotators were remunerated for the hours of speech they have provided to ensure the quality of the work done and incentivize them to do their best since annotation is a tedious and exhausting task. Concerning the demographics of our annotators, 52.6% of them are female and 47.4% are male annotators. This balance is crucial since males and females have different voice pitches. Also, since most of the annotators are students attending the university, their ages vary between 18 and 25 years old making our population of annotators consists of young adults only. The accents of our annotators also vary depending on their hometown. The majority of our annotators come from the big 5 Moroccan cities, namely Casablanca, Rabat, Tangier, Fez, and Marrakech. Hence, minor variations in the way certain words are said are captured which makes our dataset more homogenous. This diversity in voices and gender allows training a model that can generalize better and that is less prone to overfitting.

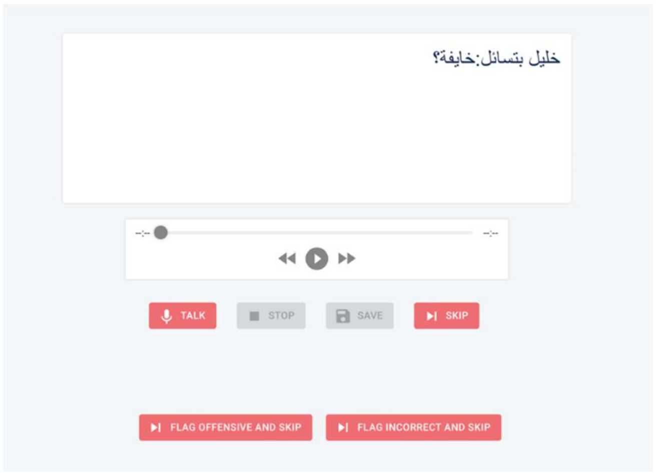
خليل بتسائل:خايفة؟

Fig. 2. Recording interface for the Annarabic Annotation Tool

In order to ensure consistency and data quality, the annotators were provided with a set of guidelines including: i) reading the sentences as they are, ii) using one's own accent, and iii) perform the recording in a normal environment (e.g., background noise is permitted). AAT allows also verifying the correctness of recorded audios by other annotators, only the upvoted clips were qualified to show up in our speech Dataset.

We collected 36 hours of annotated speech. This represents approximately 20000 sentences with lengths varying from 2 seconds to 1 minute with an average of 6.43 seconds per audio clip. However, in this work we are using only a total of 8 hours of speech due to limitation of computational resources.

B. Data Pre-processing

In this step, we first normalize the transcriptions by removing punctuation characters (e.g., !?:,…) and all the noisy

Arabic special characters that are not necessary, such as Arabic diacritics. For the audio files, a conversion from mp3 to single channel 16-bit PCM wav with a sample rate of 22,050 Hz is performed for the sake of cohesion with [13].

Next, a vocabulary is defined using Arabic letters, digits and some special characters; in this step, we use the Tensorflow pre-processing layer, StringLookUp. This layer constructs a table-based vocabulary lookup by translating a set of arbitrary strings into integer output. The vocabulary consists of the following 46 tokens plus the CTC blank character : ['', ' ','خ' ,'ح' ,'ج' ,'ث' ,'ت' ,'ب' ,'ض' ,'ص' ,'ش' ,'س' ,'ز' ,'ر' ,'ذ' ,'د' ,'ل' ,'ك' ,'ق' ,'ف' ,'غ' ,'ع' ,'ظ' ,'ط' ,',' ,':' ,'،' ,'ي' ,'و' ,'ه' ,'ن' ,'م' ,'5' ,'4' ,'3' ,'2' ,'1' ,'!' ,'ا' ,'?' 0' ,'9' ,'8' ,'7' ,'6'', '.', ' '].

Then, the raw audio waveform signals are converted into log-spectrograms. We use a frame length of 256, a frame step of 160 and an Fast Fourier Transform (FFT) length of 384.

We perform online data augmentation on the original audio clips with additive Gaussian noise and slight time stretching (resampling). For noise injection, a random value is added into the data using a noise factor that can be defined. On the other hand, the time shifting technique is nothing but shifting the audio either to the left or to the right by a random number of seconds.

C. *Architecture of DarSpeech*

The proposed model is a less complex version of Deep Speech 2 [13], it consists of two convolutional layers and three bi-directional RNN layers with Long-Short Term Memory (LSTM) units; the training is performed using the CTC loss with Batch Normalization. Fig. 3 illustrates the overall architecture of our model.

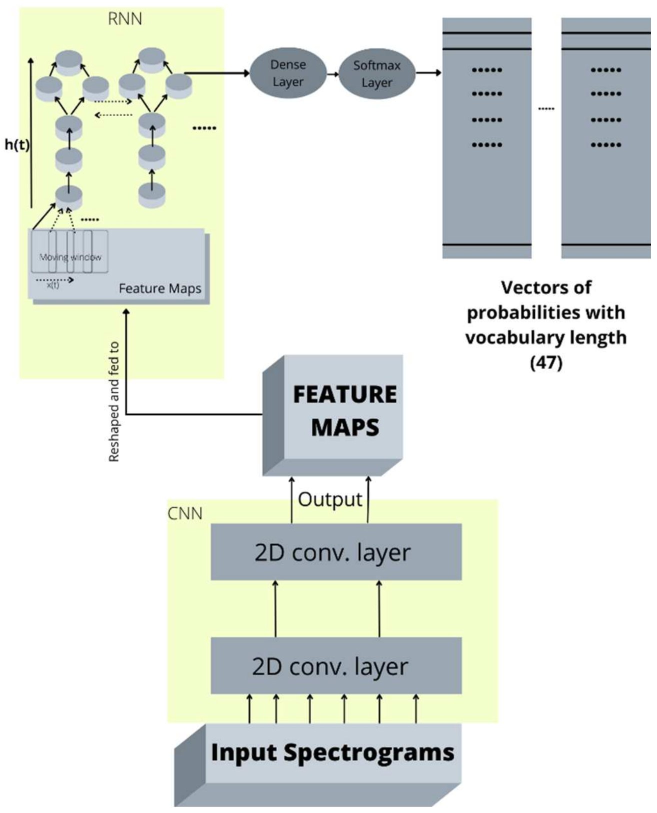

Fig. 3. Overall architecture of DarSpeech

For the first 2D convolutional layer, 32 filters are used with a kernel size of (11, 41), a stride of (2, 2), and same padding. The second 2D convolutional layer also has 32 filters with a kernel size of (11, 21), a stride of (1, 2), and same padding. Both convolutional layers are activated using ReLU.

In the recurrent network, we define three bi-directional RNN layers; each layer has 512 hidden LSTM units.

Before the output layer, a fully connected layer with 1024 units is added. The output layer is a Softmax with 47 units representing the number of characters in our defined vocabulary in addition to the CTC blank character. This output layer generates a vector of probabilities for every token in the vocabulary at each timestep. For every bidirectional LSTM, we apply dropout with a rate of 0.5 to achieve regularization.

The model's training was achieved using the stochastic gradient descent algorithm with Adam as optimizer [20]. For hyperparameters, such as the sizes of filters and strides, we adopted values used in Deep Speech 2 [13], except for the batch size that we had to decrease to 4 training examples in order to speed up the training process. Table I. summarizes our model.

TABLE I. SUMMARY OF DARSPEECH MODEL

Type	Output shape	Number of parameters
Input layer	(None, None, 193)	0
Reshaping	(None, None, 193, 1)	0
Conv2D layer 1	(None, None, 97, 32)	14,432
Conv2D layer 1 BatchNorm	(None, None, 97, 32)	128
Conv2D layer 2	(None, None, 49, 32)	236,544
Conv2D layer 2 BatchNorm	(None, None, 49, 32)	128
Reshaping	(None, None, 1568)	0
Bi-LSTM layer 1	(None, None, 1024)	8,523,776
Applied dropout 1	(None, None, 1024)	0
Bi-LSTM layer 2	(None, None, 1024)	6,295,552
Applied dropout 2	(None, None, 1024)	0
Bi-LSTM layer 3	(None, None, 1024)	6,295,552
Applied dropout 3	(None, None, 1024)	0
Dense layer	(None, None, 1024)	1,049,600
Applied dropout 4	(None, None, 1024)	0
Output layer	(None, None, 47)	48,175

V. Results

A. Experimental Setup

Both experiments in this section are conducted using the Microsoft Azure platform as service for cloud computing. The process to perform distributed training on this platform starts by creating scripts in notebooks to use for training, in an Azure Machine Learning workspace for which Python 3.6 and Tensorflow 2.6.0 are available. We used two options from the available compute ones. For the first experiment, we train the model using a single GPU (Nvidia Tesla K80 with 18 cores), and for the second one, a distributed cluster of 4 GPUs is used for training (4 Nvidia Tesla M60 with 24 cores each).

The evaluation is done using two main metrics: the word error rate (WER) and the character error rate (CER). The word error rate is one common evaluation metric for automatic speech recognition, and it basically represents the number of errors divided by the total number of words as shown in Eq (2).

$$WER = \frac{S+I+D}{Nw} \qquad (2)$$

where *S*, *I* and *D* are the types of errors that respectively stand for Substitutions, Insertions and Deletions. Substitution is the error of having a replaced word, insertion is the error of having one or more extra words that are not part of the actual transcription, and deletion is the error of having missing/omitted words in the transcription. The character error rate is generated using the same formula for WER, except that the errors are detected on a character-level, meaning that we divide the number of character errors in S, I, and D by the total number of characters.

B. Experiment 1

We train the model on only one hour and a half of speech originally, which becomes approximately 4.5 hours of speech after online data augmentation; around 54 minutes were used for validation (with a splitting of 80% for training and 20% for validation). Training runs for about 100 epochs on a single GPU (approximately 5 minutes/epoch). Table II summarizes the results of experiment 1.

TABLE II. Summary of the Results of Experiment 1

Performance metrics	**Training**	**Validation**
WER	80%	92%
CER	74.3%	86.7%
CTC Loss	10	386.3

We notice that the difference is huge for all the considered metrics, especially for the CTC loss, indicating that the model in this experiment is overfitting, it is always not performing very well as we can infer from both the word error rate and the character error rate.

C. Experiment 2

We increase the input data to 8 hours of speech in total and scale it to 24 hours using online data augmentation, and we use around 4.8 hours for validation (80% for training and 20% for validation). Then, the initialization of the model's parameters in this experiment is done using the ones that were learned in the first experiment in order to make the training more efficient and faster; this process saves us the time to learn features from the ground up. Next, we train the model for around 60 epochs using a distributed cluster of 4 GPUs without changing the model's structure or modifying its hyperparameters. In this experiment, the model was able to achieve much better results as we can see in Table III.

TABLE III. Summary of the Results of Experiment 2

Performance metrics	**Training**	**Validation**
WER	20.9%	22.7%
CER	5.55%	6.03%
CTC Loss	32.38	34.88

We can observe that the difference in the model's performance between the two experiments is significant, meaning that scaling training examples and data contributes to the improvement of the model's generalization.

The results of this work can also be compared to those achieved in [21] for the Tunisian dialect ASR system that is based on the first version of Deep Speech [14]. The authors trained their model for 40 epochs on approximately 65 hours of speech that combines both the Tunisian dialect and modern standard Arabic (15 hours of Tunisian and 50 hours of modern standard Arabic). Their best achieved performance is 24.4% of word error rate and 18.7% of character error rate, which is outperformed by our model, especially on the CER metric with a difference of 12.67%. This difference can certainly be attributed to the several enhancements brought by DeepSpeech 2, and particularly to the use of CNNs to extract word level features.

VI. Conclusion and future work

This paper presents a ground for automatic speech recognition for dialectical Arabic in general, and the Moroccan dialect specifically. Even though the current DarSpeech model provides decent results with only around 8 hours of speech, which is considerably small compared to the number of speech hours that is used in speech recognition research. Further scaling of input data should provide better results, especially that now around 36 hours of speech is already annotated and available.

In further research, better and stronger architectures could be explored for low-resource ASR systems, such as Transformers and Conformers. Another axis of future research is to improve the performance of our ASR system by taking a different approach that combines pre-trained on a rich language, such as Modern Standard Arabic (MSA), and fine-tuning on the Moroccan dialect with low data cost.

Acknowledgment

This work was supported by Al Akhawayn University seed money grant under contract number 92567.

We would like to thank all annotators who contributed to the elaboration of the Darija speech dataset.

References

[1] Patel, Ibrahim, and Y. Srinivas Rao. "Speech Recognition Using HMM with MFCC-An Analysis Using Frequency Specral Decomposion Technique." Signal & Image Processing : An International Journal, vol. 1, no. 2, Dec. 2010, pp. 101–10. DOI.org (Crossref), https://doi.org/10.5121/sipij.2010.1209.

[2] L. Li et al., "Hybrid Deep Neural Network--Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition," 2013 Humaine

Association Conference on Affective Computing and Intelligent Interaction, 2013, pp. 312-317, doi: 10.1109/ACII.2013.58.

[3] Jiang, Dongwei, et al. “Improving Transformer-Based Speech Recognition Using Unsupervised Pre-Training.” ArXiv:1910.09932 [Cs, Eess], Oct. 2019. arXiv.org, http://arxiv.org/abs/1910.09932.

[4] Graves, Alex, et al. “Speech Recognition with Deep Recurrent Neural Networks.” ArXiv:1303.5778 [Cs], 1, Mar. 2013. arXiv.org, http://arxiv.org/abs/1303.5778.

[5] Chorowski, Jan, et al. “Attention-Based Models for Speech Recognition.” ArXiv:1506.07503 [Cs, Stat], 1, June 2015. arXiv.org, http://arxiv.org/abs/1506.07503.

[6] Zhang, Yu, et al. “Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition.” ArXiv:2010.10504 [Cs, Eess], 1, Oct. 2020. arXiv.org, http://arxiv.org/abs/2010.10504.

[7] Chung, Yu-An, et al. “W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training.” ArXiv:2108.06209 [Cs, Eess], 2, Sept. 2021. arXiv.org, http://arxiv.org/abs/2108.06209.

[8] Han, Wei, et al. “ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context.” ArXiv:2005.03191 [Cs, Eess], May 2020. arXiv.org, http://arxiv.org/abs/2005.03191.

[9] Zhao, Yue, et al. “Tibetan Multi-Dialect Speech and Dialect Identity Recognition.” Computers, Materials & Continua, vol. 60, no. 3, 2019, pp. 1223–35. DOI.org (Crossref), doi:10.32604/cmc.2019.05636. Access : https://www.ecse.rpi.edu/~cvrl/Publication/pdf/Zhao2021.pdf

[10] Oord, Aaron van den, et al. “WaveNet: A Generative Model for Raw Audio.” ArXiv:1609.03499 [Cs], Sept. 2016. arXiv.org, http://arxiv.org/abs/1609.03499.

[11] Xu, Jin, et al. “LRSpeech: Extremely Low-Resource Speech Synthesis and Recognition.” ArXiv:2008.03687 [Cs, Eess], Aug. 2020. arXiv.org, http://arxiv.org/abs/2008.03687.

[12] N, Krishna D. “Multilingual Speech Recognition for Low-Resource Indian Languages Using Multi-Task Conformer.” ArXiv:2109.03969 [Cs, Eess], Sept. 2021. arXiv.org, http://arxiv.org/abs/2109.03969.

[13] Amodei, Dario, et al. “Deep Speech 2: End-to-End Speech Recognition in English and Mandarin.” ArXiv:1512.02595 [Cs], Dec. 2015. arXiv.org, http://arxiv.org/abs/1512.02595.

[14] Hannun, Awni, et al. “Deep Speech: Scaling up End-to-End Speech Recognition.” ArXiv:1412.5567 [Cs], Dec. 2014. arXiv.org, http://arxiv.org/abs/1412.5567.

[15] Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2013). Audio Chord Recognition with Recurrent Neural Networks. Paper presented at the ISMIR.

[16] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Paper presented at the Advances in neural information processing systems.

[17] Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In ICML, pages 369–376. ACM, 2006.

[18] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[19] Aghzal, Mohamed, and Asmaa Mourhir. “Distributional Word Representations for Code-Mixed Text in Moroccan Darija.” Procedia Computer Science, vol. 189, Jan. 2021, pp. 266–73. ScienceDirect, https://doi.org/10.1016/j.procs.2021.05.090.

[20] Kingma, Diederik P., and Jimmy Ba. Adam: A Method for Stochastic Optimization. Dec. 2014. arxiv.org, https://arxiv.org/abs/1412.6980v9

[21] Messaoudi, Abir, et al. “Tunisian Dialectal End-to-End Speech Recognition Based on DeepSpeech.” Procedia Computer Science, vol. 189, Jan. 2021, pp. 183–90. ScienceDirect, https://doi.org/10.1016/j.procs.2021.05.082.

[22] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Lukasz; Polosukhin, Illia (2017-06-12). "Attention Is All You Need". arXiv:1706.03762 [cs.CL].

[23] Wang, Weizhe, et al. “End-to-End Low-Resource Speech Recognition with a Deep CNN-LSTM Encoder.” *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)*, 2020, pp. 158–62. *IEEE Xplore*, https://doi.org/10.1109/ICICSP50920.2020.9232119.