Evaluation of Automatic Speech Recognition Approaches

Regis Pires Magalhães[1], Daniel Jean Rodrigues Vasconcelos[1], Guilherme Sales Fernandes[1], Lívia Almada Cruz[1], Matheus Xavier Sampaio[2], José Antônio Fernandes de Macêdo[1], Ticiana Linhares Coelho da Silva[1]

[1]Insight Data Science Lab - Universidade Federal do Ceará (UFC)
{regis, daniel.jean, guilherme.sales, livia, jose.macedo, ticianalc}@insightlab.ufc.br
[2]Instituto de Computação - Universidade Estadual de Campinas (Unicamp)
m220092@dac.unicamp.br

Abstract. Automatic Speech Recognition (ASR) is essential for many applications like automatic caption generation for videos, voice search, voice commands for smart homes, and chatbots. Due to the increasing popularity of these applications and the advances in deep learning models for transcribing speech into text, this work aims to evaluate the performance of commercial solutions for ASR that use deep learning models, such as Facebook Wit.ai, Microsoft Azure Speech, Google Cloud Speech-to-Text, Wav2Vec, and AWS Transcribe. We performed the experiments with two real and public datasets, the Mozilla Common Voice and the Voxforge. The results demonstrate that the evaluated solutions slightly differ. However, Facebook Wit.ai outperforms the other analyzed approaches for the quality metrics collected like WER, BLEU, and METEOR. We also experiment to fine-tune Jasper Neural Network for ASR with four datasets different with no intersection to the ones we collect the quality metrics. We study the performance of the Jasper model for the two public datasets, comparing its results with the other pre-trained models.

Categories and Subject Descriptors: Computing methodologies [**Artificial intelligence**]: Natural language processing—*Speech recognition*; Human-centered computing [**Human computer interaction (HCI)**]: Interactive systems and tools; Computing methodologies [**Machine learning**]: Machine learning approaches—*Neural networks*

Keywords: automatic speech recognition, speech translation, speech to text

1. INTRODUCTION

Automatic Speech Recognition (ASR) techniques to transform speech-to-text [Reddy 1976] have gained increased importance in recent years and have applications in many problems, such as screen readers, automatic video and music captioning, etc [Graves et al. 2013]. One of those applications is chatbots, which gained popularity due to the adoption of messaging services and advances in Artificial Intelligence and Deep Learning.

This work proposes to evaluate the performance of the commercial APIs of ASR Facebook Wit.ai, Microsoft Azure Speech Services, and Google Cloud Speech-to-Text on two public Portuguese datasets, called Mozilla Common Voice and Voxforge. The most used metric to evaluate ASR is the Word Error Rate (WER) [Këpuska and Bohouta 2017], however, it is limited to determine the rate of incorrect words in the transcription. In this work, we applied other NLP metrics to also validate if the models keep the original sentence structure and organization and if they generate transcriptions with similar vectorial representation. We also investigate to fine tune the Jasper model for ASR instead of using pretrained a model. Only Jasper and Wav2Vec offer the model architecture to train. We chose Jasper to fine-tune since it outperforms the others sequence-to-sequence ASR models as shown in [Li et al. 2019]. From the authors' knowledge, there is no Jasper pre-trained model in Portuguese language or a multilingual available to use as a pre-trained. In this paper, we **fine-tune** Jasper from an English pre-trained version [1].

Other works evaluated ASR models [Këpuska and Bohouta 2017; Filippidou and Moussiades 2020], but they aimed at evaluating performance in English and using only WER or WER in combination with the precision and recall of words [Filippidou and Moussiades 2020]. [de Lima and Da Costa-Abreu 2020] presents a survey of techniques and data sets for ASR in Portuguese. However, it does not offer an experimental evaluation of these techniques. The main contribution of this work is to offer a comparison of different ASR models according to different metrics. This may help data scientists to choose one of these available models.

This paper is an extension of [Sampaio et al. 2021]. The paper [Sampaio et al. 2021] proposes the following contributions: (i) offers a comparison between three commercial solutions for ASR, as Facebook Wit.ai, Microsoft Azure Speech, and Google Cloud Speech-to-Text, and (ii) adopts the metrics WER, BLEU and METEOR and cosine similarity on embedding space to gain insight on the word translation and the contextual reflected during translation. This work substantially extends the work conducted in [Sampaio et al. 2021] the following ways: (i) we expand the related work section; (ii) we include another commercial API: AWS Transcribe; additionally, we included the publicly available architecture and model Wav2Vec 2.0 that has a pre-trained deep learning model for ASR in Brazilian Portuguese; (iii) we fine-tune and include in the comparison the efficient ASR model called Jasper; and (iv) we further expand the experimental evaluation to better assess the quality of the ASR models regarding the quality metrics like WER, BLEU and METEOR for two public datasets.

The remaining sections of this article are organized as follows. Section 2 provides a brief review of the main concepts used in this paper. Section 3 presents the related works. Section 4 shows the ASR models used throughout this work. Section 5 explains the evaluation metrics used on the comparisons. Section 6 presents the experimental setup and Section 7 shows the analysis performed compared to the ASR models. Finally, Section 8 summarizes this work and proposes future developments.

2. PRELIMINARIES

This section introduces briefly some concepts related to the usage and the layers of the ASR approaches investigated in this paper.

Speech recognition proposals usually require massive training data to reach acceptable performance, which is challenging. The solutions for ASR investigated in this work provide pre-trained models, which can be helpful in another dataset different from their training set. We can directly use the pre-trained model in a new dataset or train and fine-tune it. In the second alternative, the knowledge gained while training a model for a dataset can be applied to a different but related dataset.

In this paper, we investigate both approaches. We analyze the quality metrics for the output from the pre-trained models and we investigate the output for a fine-tuned model. Instead of learning the parameters from scratch along with a new dataset, pre-trained ASR models have previously trained weights and can be trained and tweaked in two ways: static or dynamic. The model layers are started from the pre-trained model weights in dynamic form, and the update is propagated through the neural network during training. One or more layers can be frozen in static form, preventing weights from updating. Both strategies are called fine-tuning. The dynamic form is the one we investigate to train the Jasper model.

The models we study in this paper present different layers, such as convolutional, recurrent neural networks (RNNs), attention mechanisms, among others. We provide a brief description of the most relevant in what follows.

A typical **Convolution Neural Network** is usually composed of three main types of layers:

[1] `https://nvidia.github.io/OpenSeq2Seq/html/speech-recognition.html`

Convolutional Layer, Pooling Layer, and Softmax Layer. The convolutional layers generate feature maps by applying consecutive convolution between a group of trainable convolutional kernels and the input. From this, the network can learn filters that activate when some feature or a specific pattern appears. The pooling layers are utilized to progressively reduce the spatial size of the representation to reduce the dimension of feature maps and computation in the network, and hence to also control overfitting. The softmax layers output the normalized probability of each label and in general, are used at the end of a CNN [CS231n: Convolutional Neural Networks for Visual Recognition 2022]. It is worth to mention Jasper uses 1D Convolutional layers, i.e., the kernel slides along one dimension. It is commonly used on time series data and text. Different from 2D Convolutional layers the kernel moves in 2 dimensions, commonly used on image data.

The **Recurrent Neural Network** (RNN) or more specifically, **Long Short-Term Memory** (LSTM) has been extensively used in NLP tasks since it processes variable-length input and can allow highly non-trivial long-distance dependencies to be easily learned. Bi-directional LSTM model can take into account an effectively infinite amount of context on both sides of a word and eliminates the problem of limited context that applies to any feed-forward model [Chiu and Nichols 2016].

Transformers were introduced in [Vaswani et al. 2017]. A simple mechanism called "neural attention" could be used to build powerful sequence models that did not feature any recurrent layers or convolution layers [Chollet 2021]. The attention layer is often used in an encoder, allowing the model to automatically search for parts of an input sentence that are relevant to predict the label of a target word, without having to assume that these parts form a rigid segment [Bahdanau et al. 2015], that is, it does not try to encode the entire input sentence into a single fixed-length vector. It is especially useful for long sentences, but improvements can be seen with sentences of any length. Rather than generating the same context vector, or phrase representation, for all words in the input text, the attention engine allows the encoder to calculate a different context vector for each word based on a model or alignment function. This model provides a score of how appropriate the relationship between the word and the encoder output is.

Activation functions are a crucial part of deep learning models as they add the non-linearity to neural networks. There is a great variety of activation functions in the literature, and some are more beneficial than others. Two of the "oldest" activation functions are still commonly used for various tasks: sigmoid and tanh. Another popular activation function that has allowed the training of deeper networks, is the Rectified Linear Unit (ReLU). Despite its simplicity of being a piecewise linear function, ReLU has one major benefit compared to sigmoid and tanh: a strong, stable gradient for a large range of values. Based on this idea, a lot of variations of ReLU have been proposed [MSc program in Artificial Intelligence of the University of Amsterdam 2021].

Another technique used in deep learning models is **dropout**. Dropout is applied to a layer and consists of randomly dropping out (setting to zero) a number of output features of the layer during training. Dropout works as a regularization technique, different from weight regularization that is typically for smaller deep learning models, dropout tends to be applied in large deep learning models [Chollet 2021].

3. RELATED WORKS

This section details some related works which evaluate different ASR strategies or services.

The Speech Recognition Benchmark[2] is a lightweight, open-source framework that assesses the performances of automated speech recognition (ASR) APIs by comparing the predicted transcriptions with the reference transcriptions [Dernoncourt et al. 2018]. The framework supports the following 7 ASR APIs: Google Speech Recognition, Google Cloud Speech API, Houndify API, IBM Speech to

[2] `https://github.com/Franck-Dernoncourt/ASR_benchmark`

Text, Microsoft Bing Speech-to-Text, Speechmatics, and Wit.ai. It is easily extendable to more APIs. The work compares the ASR APIs with five datasets. Speechmatics reaches lower WER for three datasets, while Google was the best model for the other two datasets.

Libri-light [Kahn et al. 2020] is a benchmarking for ASR systems under limited or no supervision. Libri-light includes a large open-source corpus of audio in English derived from the LibriVox project, a common set of evaluation metrics and baseline systems. The data and metrics were segmented to evaluate unsupervised, semi-supervised, and distant supervision settings. They used the supervised systems from LibriSpeech as baselines and compared them to a CPC trained with unlabeled speech and fine-tuned with limited data and an MFSC TDS trained on limited labeled data. For unsupervised settings, CPC reached good scores when compared to the baselines. The unsupervised training using more data could improve the performance of phoneme recognition tasks in the semi-supervised setting and word recognition tasks in a distant-supervision setting.

[Karita et al. 2019] compare Transformers and conventional Recurrent Neural Networks (RNN) for ASR, speech translation, and text-to-speech tasks and also compare their results to reports for LibriSpeech ASR benchmark. The ASR experiments covered English, Japanese, Mandarin, Chinese, Spanish and Italian languages on fifteen datasets. They report ASR results in character and word rates (CER/WER). In their experiments, Transformer has outperformed RNN-based in mono-language and multi-language systems.

[Likhomanenko et al. 2020] analyzed the domain transfer in ASR models. They used a single Transformer-based acoustic model(AM) architecture and conducted their experiments on public datasets restricted to English. They provided the results for baselines trained on each dataset and a unique model trained on the integrated dataset. The authors evaluated each model on all validation and test sets to estimate how the models transfer to ”out-of-domain” data. Their results show that: i) in general, AM trained on a single dataset performs poorly on other datasets; ii) For single dataset, training the transfer quality can vary a lot; iii) The joint model performs well, and iv) joint model with noise improved the robustness.

The work [Chiu et al. 2018] proposed and explored many improvements of the encoder-decoder Listen, Attend and Spell (LAS) architecture for ASR systems. They used word piece models (WPM), incorporated multi-head attention (MHA) mechanism [Vaswani et al. 2017], explored training the model to minimize the number of expected word errors, using synchronous SGD as an optimizer, schedule sampling, and incorporated a language model. Their experimental evaluation combined different strategies that have yielded significant improvements in WER.

4. ASR MODELS

This section introduces the ASR models investigated in this work.

Facebook Wit.ai[3] is a service, which is a natural language interface for applications capable of turning sentences into structured data. It is a free service, including for commercial use. Wit.ai has a Wit Speech API[4] for converting speech to text using state-of-the-art Natural Language Processing techniques and many speech recognition engines in order to achieve low latency and high robustness to both surrounding noise and paraphrastic variations [Mitrevski 2018].

Wav2Vec bypasses the problem of training a model with huge datasets [Baevski et al. 2020]. It is trained with limited amounts of labeled data. Wav2Vec jointly learns discrete speech units with contextualized representations. The model architecture comprises a feature encoder that receives the raw waveform as input and feeds several blocks containing a temporal convolution followed by layer

[3] https://wit.ai/faq

[4] https://wit.ai/docs/http/20210928/\#post__speech_link

Table I. Comparison of models used by the ASR APIs

Paper	Architecture	Training Corpus	Test WER
Facebook Wit.ai [Mitrevski 2018]	Encoder-Decoder built with fully connected CNN	1041 hours of audio combining the Wall Street Journal and Librispeech datasets in English	2.4% in the Wall Street Journal dataset
Microsoft Azure Speech Services[Xiong et al. 2018]	CNN Encoder and BiLSTM Decoder	2000 hours of audio from the Switchboard dataset and 25 hours of audio from the CallHome dataset, both in English	5.1 % in the SwitchBoard dataset and 9.8 % in the CallHome dataset
Google Cloud Text-to-Speech[Chiu et al. 2018]	LAS with Multi-headed Attention	12500 hours of audio consisting of 15 million phrases taken from Google Voice Search in English	5.6% in Google Voice Search dataset
Wav2Vec 2.0 [Baevski et al. 2020]	CNN encoder and Linear projection to output representations	Fine-tuned the original model with Brazilian Portuguese audios: 145 hours of audio from CETUC, 284 hours of audio from Multilingual Librispeech, 50 hours of audio from Common Voice	12.9 % in Common Voice Portuguese Test dataset
Jasper [Li et al. 2019]	1D Convolution, batch normalization, ReLU dropout and residual connections	LibriSpeech and Wall Street Journal (WSJ)	3.74% on LibriSpeech dev-clean and 10.21% on dev-other

normalization and a GELU activation function. The output of the feature encoder is fed to a context network that follows a Transformer architecture.

Microsoft Azure Speech Services. The Microsoft AI and Research team proposed a speech recognition system composed of a combination of Convolutional Neural Networks (CNN) architectures Residual Network (ResNet) and Layer-wise Context Expansion with Attention (LACE), and Bi-directional Long Short Term Memory (Bi-LSTM) layers [Xiong et al. 2017; Xiong et al. 2018]. In addition, it adds language models based on LSTM at the word and character levels responsible for reclassifying the output at the end of the model.

Google Cloud Text-to-Speech. The Google AI team proposed the Listen, Attend, and Spell (LAS) [Chan et al. 2016], that uses an Encoder-Decoder with Recurrent Neural Network (RNN). They improved LAS by adding a Multi-headed Attention layer, a new training metric based on the minimum rate of word errors, and the use of an external language model during inference [Chiu et al. 2018].

Amazon Transcribe [Amazon Transcribe Site 2021] is part of Cloud Computing Services from Amazon Web Services (AWS). Amazon Transcribe uses a deep learning model to perform ASR to quickly and accurately convert speech to text. In this conversion, the data needs to be first uploaded to Amazon Simple Storage Service (Amazon S3). Then Transcribe calls the objects from S3 for transcription. The model provided automatically adds punctuation and number formatting, so that the output closely matches the quality of manual transcription at a fraction of the time and expense. Numbers are also transcribed into digits or "normal form" instead of words.

Jasper [Li et al. 2019] is an end-to-end ASR model that uses a stack of 1D Convolution layers, batch normalization, ReLU, dropout, and residual connections. The paper [Li et al. 2019] also proposes a new residual connection topology Dense Residual. Instead of having dense connections within a block, the output of a convolution block is added to the inputs of all the following blocks. In the paper, the authors show that residual connections are necessary to converge during the training.

Table I presents a comparison between the architectures, training data, and test results of the models. The papers show the WER of the audio transcriptions on test sets selected for each work, except for Amazon Transcribe, since we could not find any research papers providing information about its architecture, training corpus, and WER performance. We also cannot assure that all architectures from Table I were not updated overtime.

5. ASR EVALUATION METRICS

This section presents the evaluation metrics for ASR systems used in the experimental of this work. These metrics assess the transcript quality by evaluating the difference between two sentences and their contexts. Given a reference text T^* consisting of the correct transcription of an audio and the text T consisting of the ASR transcription of T^*, the goal of the evaluation metric is to estimate an error based on the comparison between T^* and T.

5.1 Word Error Rate

The word error rate (WER) is the most used evaluation metric for ASR systems. The percentage of incorrect words gives the WER of a transcription concerning the number of input words. The incorrect words were erroneously inserted, replaced, or deleted by the system transcription. WER is defined as in Equation 1.

$$WER = \frac{I + R + D}{H + R + D} \tag{1}$$

where I is the number of inserted words, R is the number of replaced words, D is the number of deleted words, and H is the number of hits. Despite its popularity, WER is limited to the accuracy at the word level.

5.2 BLEU

Different from WER, BLEU [Papineni et al. 2002] can evaluate whether the transcription maintains the context and organization of the sentence. BLEU was originally proposed for neural machine translation and it claims to be highly correlated with human assessment.

BLEU is based on the precision of n-grams, which compares the n-grams of reference text T^* with the n-grams of its transcription T. Let be $NG(n, t)$ the set of n-grams of text t, the n-gram precision P_n (Equation 2) between texts T^* and T.

$$P_n = \frac{|NG(n, T^*) \cap NG(n, T)|}{NG(n, T)} \tag{2}$$

BLEU is calculated as the geometric mean of P_n, for $n = 1, 2, 3, 4$ multiplied by a factor that penalizes transcriptions shorter than the referenced text. The $bleu_{penalty}$ factor is 1 if $|T| > |T^*|$ and $e^{1-|T^*}|/|T|$, otherwise. BLEU is defined in Equation 3.

$$BLEU = \sqrt[4]{P_1 P_2 P_3 P_4} \times bleu_{penalty} \tag{3}$$

5.3 METEOR

METEOR was proposed by [Banerjee and Lavie 2005] to fix limitations of BLEU, such as the fact that it does not require explicit word-to-word matching. Another limitation is that its score results in zero whenever one of the n-gram precision is zero, which means the score at sentence level can be meaningless.

METEOR is based on the harmonic mean of unigram precision and recall, multiplied by a penalty factor.

The n-gram recall is defined in Equation 4 and METEOR in Equation 5.

$$R_n = \frac{|NG(n, T^*) \cap NG(n, T)|}{NG(n, T^*)} \tag{4}$$

$$METEOR = \frac{10P_1R_1}{R_1 + 9P_1} \times meteor_{penalty} \tag{5}$$

To calculate the penalty ($meteor_{penalty}$) in METEOR (Equation 6), the unigrams in $NG(n, T^*) \cap NG(n, T)$ are grouped in chunks, such as each chunk has the maximum number of unigrams in adjacent positions in both T^* and T. The fewer the chunks, the better system transcription matches with the reference transcription.

$$meteor_{penalty} = 0.5 \times \frac{\#chuncks}{|NG(1, T^*) \cap NG(1, T)|} \tag{6}$$

5.4 Cosine Similarity

At the same time, the **Cosine Similarity** allows determining how close the two sentences are in a defined vector space. For the Cosine Similarity, we use the Word Embedding vectors produced in [Hartmann et al. 2017] by using the Word2Vec approach in both Continuous Bag of Words (CBOW) and Skip-Gram variations, with 50 dimensions. The Cosine Similarity is defined as in Equation 7, where A and B are vectors of attributes; A_i and B_i are components of vector A and B, respectively; $\|\mathbf{A}\|$ is the Euclidean norm of vector A. Similarly, $\|\mathbf{B}\|$ is the Euclidean norm of vector B.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i\mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{7}$$

6. EXPERIMENTAL SETUP

Datasets. The experiments use public and collaborative audio datasets in Portuguese, the Mozilla Common Voice[5] and the Voxforge[6] datasets. Mozilla Common Voice collaborators can record the audio and evaluate the available data quality. Voxforge is composed of sentences from audiobooks of the public domain. It is frequently updated but not versioned, so its official site publishes only its most recent version. For reproducibility of our results, the VoxForge dataset release that we use is available at a repository available on Zenodo[7] Table II presents the characteristics of the used datasets such as the number of recorded sentences, the size of vocabulary, the average time of audios, average length of sentences in terms of the number of characters, the original audio format, and frequency. We conduct the experiments on over 10,000 sentences selected from these datasets, accounting for more than 12 hours of audio. We chose all sentences with female or not informed voices and randomly selected from the remaining male voices to reduce the data imbalance, with the resulting distribution presented in Table III.

Data preparation. From the dataset, we remove the characters that are not recognized by Portuguese language and punctuation, like an exclamation mark. We also separate the dataset into files by gender to collect the API WER results.

ASR models. We compare in this paper some of the widely used commercial ASR models, like Facebook Wit.ai, Google Could Text-to-Speech, AWS Transcribe, and Microsoft Azure. We also investigated two other models that are not commercial solutions, as Wav2Vec[8] and Jasper. The former was fine-tuned using some Brazilian Portuguese datasets, like CETUC, VoxForge, and Common Voice. The pre-trained model is available on Hugging Face, and we collected the results reported in the next section from it. Jasper model was fine-tuned from a pre-trained model[9].

[5] https://commonvoice.mozilla.org/pt

[6] http://www.voxforge.org/pt

[7] https://zenodo.org/record/6077607.

[8] https://huggingface.co/lgris/wav2vec2-large-xlsr-open-brazilian-portuguese

[9] https://nvidia.github.io/OpenSeq2Seq/html/speech-recognition.html\#speech-recognition

Run-time parameters. The Jasper model is the only ASR model we need to set up the parameters' values since it is the only model we train in this paper as Jasper provides an open model architecture. For the parameters to fine-tune Jasper, we experimented with batch sizes of 32 and 300 epochs. The remaining parameters and hyper-parameters are kept the same as proposed by the authors. The only significant change was to replace the English for the Brazilian vocabulary with its Latin characters, such as 'ã', 'ô', etc.

Evaluation Overview. Our evaluation first presents an analysis concerning the quality results yielded by the ASR models. We present the results according to WER, BLEU, METEOR, and Cosine Similarity metrics for two public datasets. The evaluation then concludes by analyzing the performance of the ASR models by gender. Table III shows the gender distribution over the sentences from the two datasets.

Table II. Characteristics of the datasets used in our experiments

Corpus	Number of sentences	Size of vocabulary	Average audio duration in seconds	Average sentence size in characters	Format and frequency of the original audio
Mozilla Common Voice	4586	6310	4.45(±1.44)	35.19(±16.73)	mp3 48KHz
Voxforge	4115	566	3.67(±1.21)	22.27(±9.73)	wav 48KHz

Table III. Gender distribution over the sentences on the datasets used in our experiments

Voice gender	Mozilla Common Voice	Voxforge
Male	3350 (73.03%)	2800 (70%)
Female	509 (11.10%)	200 (5%)
Not informed	728 (15.87%)	1000 (25%)

7. RESULTS

In this paper, we aim at evaluating the quality of Wit.ai, Azure Speech Services, Google Cloud Speech to Text, and Amazon Transcribe APIs transcriptions when applied to an extensive dataset in Portuguese. In addition to these services, we also evaluated the models Wav2Vec and Jasper that have their architecture and weights open to the community and are free to use. The metrics used in these experiments are WER, BLEU, METEOR, and Cosine Similarity. Tables IV and V present a summary of the experimental results for two datasets. We omit to compare Wav2Vec for VoxForge Corpus since its model was trained using that dataset.

The experiments show a result for WER between 6.69% and 50.77%. We remind that the lower WER, the better the associated ASR model. Facebook Wit.ai and Microsoft Azure Speech Services consistently outperform the other models for the WER metric. However, due to the nature of the WER metric, even one character error makes a whole word incorrect. We performed Wilcoxon signed-rank test [Wilcoxon 1992] since it is a non-parametric test. From the statistical point of view, the test is safer since it does not assume normal distributions [Demšar 2006]. Our null hypothesis (H_O) states that the models perform equally well for WER results. We reject the null hypothesis at a 0.05 significance level, indicating that the WER results for the analyzed models for both datasets are statistically different. We chose to analyze only WER results since this measure is the most utilized to assess ASR models.

We use BLEU and METEOR metrics to obtain a more accurate picture of the models and APIs' ability to maintain sentence structure during transcriptions. The higher the values for BLEU and METEOR, the better the associated ASR model. From Tables IV and V, we observe that all ASR models reported slightly different values for BLEU and METEOR metrics – indeed, the approaches yield high values for both metrics on both datasets, except for the Jasper model. BLEU and METEOR are more confident metrics than WER since they measure if the words are recognized in the sentences and maintained in a sequence compared to the original text.

Except for the Jasper results, the cosine similarity values are slight the same for the compared approaches and close to 1 (better). We have computed the embeddings of the transcribed sentences from the average of the multidimensional vector of each word from a pre-trained Word2Vec model [Hartmann et al. 2017] with 50 dimensions. In general, as cosine similarity abstracts the word order, and except for Jasper results, we can conclude that the APIs and ASR models produced texts that are pretty similar to the original sentences. The Skip-Gram variations obtained a marginally better result compared to CBOW. Facebook Wit.ai is consistently superior to the other APIs and models for these metrics.

From the paper [Li et al. 2019], we expected Jasper to achieve better results after performing the dynamic fine-tuning using a Brazilian Portuguese dataset. However, during the fine-tuning process, we realized that the amount of GPU memory resources demanded to execute the process far surpassed our local infrastructure. To bypass this problem, we leveraged the resources of Google Colaboratory Pro. Nonetheless, we were only able to train for about 15 hours each day for one month. Our best model reached around 50% in the WER metric within this time frame. Therefore, we conclude that the excellent results of the Jasper approach reported on [Li et al. 2019] come with the cost of thousands of training/fine-tuning hours. By contrast, the inference time is about 100ms on average using the dataset described in this work.

Table IV. API results on Mozilla Common Voice Corpus

API	WER(%)	BLEU	METEOR	Word2Vec CBOW	Word2Vec SKIP
Facebook Wit.ai	**6.69**	**0.871**	**0.923**	**0.959**	**0.964**
Microsoft Azure Speech Services	7.97	0.864	0.920	0.944	0.950
Google Cloud Text-to-Speech	12.71	0.779	0.877	0.913	0.923
Wav2Vec 2.0	10.80	0.793	0.888	0.921	0.931
AWS Transcribe	14.94	0.734	0.857	0.904	0.917
Jasper	24.11	0.577	0.755	0.813	0.835

Table V. API results on Voxforge Corpus

API	WER(%)	BLEU	METEOR	Word2Vec CBOW	Word2Vec SKIP
Facebook Wit.ai	**7.05**	**0.832**	**0.915**	**0.953**	**0.952**
Microsoft Azure Speech Services	8.14	0.825	0.899	0.942	0.943
Google Cloud Text-to-Speech	11.58	0.748	0.852	0.911	0.913
AWS Transcribe	10.27	0.799	0.881	0.930	0.931
Jasper	50.77	0.228	0.474	0.547	0.553

We also investigate the influence of voice gender on transcription quality as shown in Figure 1. We can observe that all approaches recognize male voices better than female voices, with WER variations less than 3% for Wit.ai and Azure Speech Services and almost 4% for Google Cloud Speech-to-Text. This is already expected, according to Table III, the majority of sentences are composed by male voice(s). This disparity in results by gender of voice is repeated in the other metrics, as shown in Figure 1.

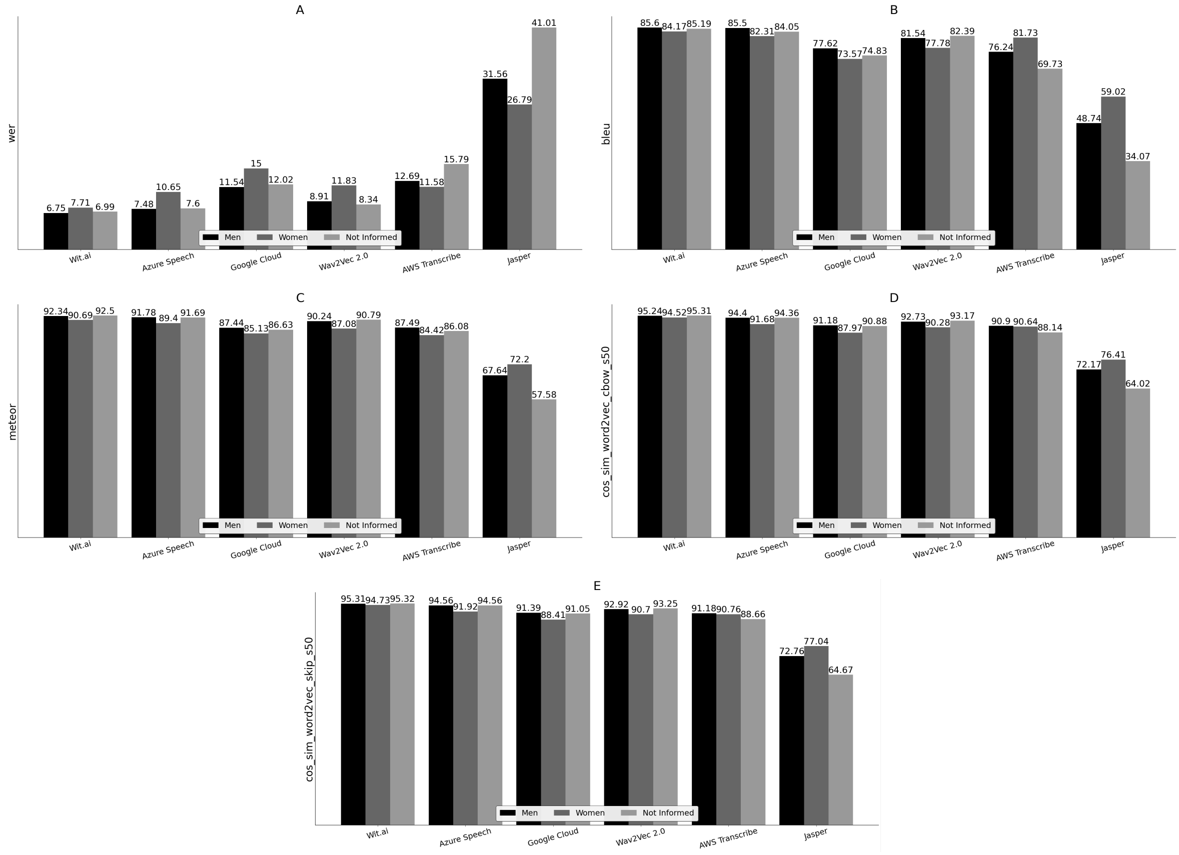

Fig. 1. API and Model results by Gender. **(A)** shows the WER metric; **(B)** shows the BLEU Score; **(C)** shows the METEOR Score; **(D)** and **(E)** show the Cosine Similarity with CBOW and SKIP variations, respectively

8. CONCLUSION

This work proposes an evaluation of speech recognition systems to interact with chatbots through voice. In order to achieve such an objective, we have explored tools to execute the ASR tasks, obtained datasets, and studied metrics for carrying out tests and experiments.

After analyzing the techniques used by the APIs, we carried out experiments to assess the quality of speech transcription in Portuguese. We used WER, the primary metric for analyzing voice-to-text transcription, in addition to metrics that calculate the similarities between sentences, with text translation evaluation metrics BLEU and METEOR, and Cosine Similarity using Word Embeddings. For this, we used the datasets Mozilla Common Voice and Voxforge.

We can also observe the impact that the voice gender has on the accuracy of the transcriptions. The results showed similar performances between the tools in all metrics, with an advantage to Facebook Wit.ai.

For future works, the comparison between the accent of different regions of Brazil could evaluate if it influences the quality of transcriptions. Additionally, we plan to use a larger dataset to train the Jasper model with more extended periods in the fine-tuning phase to obtain results that are on par with those reported by the paper [Li et al. 2019]. We also aim at investigating the inverse path, i.e., from text to speech models.

Acknowledgments. This work is partially supported by the FUNCAP projects 04772314/2020, 04772420/2020, and 04772551/2020.

REFERENCES

Amazon Transcribe Site. Amazon Transcribe. `https://aws.amazon.com/transcribe/?nc=sn&loc=0`, 2021. [Online; accessed 11-January-2021].

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020*. Curran Associates, Inc., Virtual-only Conference, pp. 12449–12460, 2020.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*. ICLR, San Diego, CA, USA, 2015.

Banerjee, S. and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, Ann Arbor, Michigan, pp. 65–72, 2005.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, IEEE, Shanghai, China, pp. 4960–4964, 2016.

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 *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, IEEE, Calgary, Alberta, Canada, pp. 4774–4778, 2018.

Chiu, J. P. and Nichols, E. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics* vol. 4, pp. 357–370, 2016.

Chollet, F. *Deep learning with Python*. Manning Publications, Shelter Island, NY, 2021.

CS231n: Convolutional Neural Networks for Visual Recognition. Convolutional neural networks for visual recognition. http://cs231n.github.io/, 2022. Accessed: 2022-01-12.

de Lima, T. A. and Da Costa-Abreu, M. A survey on automatic speech recognition systems for portuguese language and its variations. *Computer Speech & Language* vol. 62, pp. 101055, 2020.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* vol. 7, pp. 1–30, 2006.

Dernoncourt, F., Bui, T., and Chang, W. A framework for speech recognition benchmarking. In *Proc. Interspeech 2018*. ISCA, Hyderabad, pp. 169–170, 2018.

Filippidou, F. and Moussiades, L. A benchmarking of ibm, google and wit automatic speech recognition systems. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer International Publishing, Cham, pp. 73–82, 2020.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, Vancouver, BC, Canada, pp. 6645–6649, 2013.

Hartmann, N. S., Fonseca, E. R., Shulby, C. D., Treviso, M. V., Rodrigues, J. S., and Aluísio, S. M. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*. SBC, Porto Alegre, RS, Brasil, pp. 122–131, 2017.

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., et al. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Barcelona, Spain, pp. 7669–7673, 2020.

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. E. Y., Yamamoto, R., Wang, X., et al. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, Singapore, pp. 449–456, 2019.

Këpuska, V. and Bohouta, G. Comparing speech recognition systems (microsoft api, google api and cmu sphinx). *Int. J. Eng. Res. Appl* 7 (03): 20–24, 2017.

Li, J., Lavrukhin, V., Ginsburg, B., Leary, R., Kuchaiev, O., Cohen, J. M., Nguyen, H., and Gadde, R. T. Jasper: An End-to-End Convolutional Neural Acoustic Model. In *Proc. Interspeech 2019*. ISCA, Graz, Austrian, pp. 71–75, 2019.

Likhomanenko, T., Xu, Q., Pratap, V., Tomasello, P., Kahn, J., Avidov, G., Collobert, R., and Synnaeve, G. Rethinking evaluation in ASR: are our models robust enough? *CoRR* vol. abs/2010.11745, pp. arXiv:2010.11745, 2020.

Mitrevski, M. Getting started with wit.ai. In *Developing Conversational Interfaces for iOS: Add Responsive Voice Control to Your Apps*. Apress, Berkeley, CA, pp. 143–164, 2018.

MSc program in Artificial Intelligence of the University of Amsterdam. Deep Learning Tutorials. `https://uvadlc.github.io/`, 2021. [Online; accessed 12-January-2021].

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Association for Computational Linguistics, USA, pp. 311–318, 2002.

Reddy, D. R. Speech recognition by machine: A review. *Proceedings of the IEEE* 64 (4): 501–531, 1976.

Sampaio, M., Magalhães, R., Silva, T., Cruz, L., Vasconcelos, D., Macêdo, J., and Ferreira, M. Evaluation of automatic speech recognition systems. In *Anais do XXXVI Simpósio Brasileiro de Bancos de Dados*. SBC, Porto Alegre, RS, Brasil, pp. 301–306, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *"31st Conference on Neural Information Processing Systems*. (NIPS'17). Curran Associates Inc., Long Beach, CA, USA, pp. 6000–6010, 2017.

Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*, S. Kotz and N. L. Johnson (Eds.). Springer New York, New York, NY, pp. 196–202, 1992.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M. L., Stolcke, A., Yu, D., and Zweig, G. Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (12): 2410–2423, 2017.

Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., and Stolcke, A. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, IEEE, Nova Orleans, EUA, pp. 5934–5938, 2018.