UNIVERSIDADE FEDERAL DE LAVRAS

ALINE RODRIGUES GUIMARÃES DE OLIVEIRA

EVALUATION OF AUTOMATIC SPEECH RECOGNITION TOOLS FOR PORTUGUESE LANGUAGE

LAVRAS – MG

2022

ALINE RODRIGUES GUIMARÃES DE OLIVEIRA

EVALUATION OF AUTOMATIC SPEECH RECOGNITION TOOLS FOR PORTUGUESE LANGUAGE

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para a obtenção do título de Bacharel.

Prof. Dr. Luiz Henrique de Campos Merschmann
Orientador

LAVRAS – MG
2022

ALINE RODRIGUES GUIMARÃES DE OLIVEIRA

EVALUATION OF AUTOMATIC SPEECH RECOGNITION TOOLS FOR PORTUGUESE LANGUAGE

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para a obtenção do título de Bacharel.

APROVADA em 28 de Abril de 2022.

Prof. Dr. Luiz Henrique de Campos Merschmann	UFLA
Prof. Dr. Paula Christina Figueira Cardoso	UFLA
Dr. Erick Galani Maziero	

Prof. Dr. Luiz Henrique de Campos Merschmann
Orientador

LAVRAS – MG
2022

AGRADECIMENTOS

Agradeço aos meus pais, Ariane e Ney Roberto, pelo exemplo, apoio e influência ao longo da vida.

Agradeço ao meu orientador, Luiz Merschmann, pelos conhecimentos.

Agradeço ao meu irmão e aos meus companheiros de graduação, Miguel e Alice, Álvaro, Leonardo e Luiz Otávio, pela amizade.

Agradeço à Lorena pelo suporte emocional e apoio incondicional.

Só sei que nada sei
(Sócrates)

Sumário

RESUMO

Ao longo dos anos, o reconhecimento automático de fala (ASR) tem recebido cada vez mais foco e interesse da indústria e da academia. Consequentemente, as ferramentas com o objetivo de transcrever dados de áudio, em vários idiomas e suas variantes, foram desenvolvidos. Além disso, conjuntos de dados contendo dados de áudio junto com seu texto transcrito são criados para treinar e avaliar os modelos utilizados por essas ferramentas de ASR. Este trabalho tem como objetivo comparar várias ferramentas de ASR em termos de desempenho de precisão, precificação e tempo de execução, para língua portuguesa em diferentes domínios. Além disso, mapeamos o ambiente de dados existente para a língua portuguesa e discutimos métricas para avaliação de ASR.

Palavras-chave: Processamento de áudio . Análise de dados . Machine learning . Reconhecimento de voz . Inteligência artificial.

Evaluation of Automatic Speech Recognition Tools for Portuguese Language

Aline Rodrigues Guimarães de Oliveira[1] **, Luiz Henrique de Campos Merschmann**[2]

[1]Instituto de Ciências Exatas – Universidade Federal de Lavras (UFLA)

aline.oliveira2@estudante.ufla.br[1], luiz.hcm@ufla.br[2]

Abstract. *Over the years, automatic speech recognition (ASR) has received increasing attention and interest from industry and academia. Consequently, tools aiming to transcribe audio data, in multiple languages and their variants, have been developed. Furthermore, datasets containing audio data along with their transcribed text are created to train and evaluate the models used by these ASR tools. This work aims to compare multiple ASR tools in terms of accuracy performance, pricing, and execution time, for Portuguese language in different domains. In addition, we map the existing data environment for the Portuguese language and discuss metrics for ASR evaluation.*

Resumo. *Ao longo dos anos, o reconhecimento automático de fala (ASR) tem recebido cada vez mais foco e interesse da indústria e da academia. Consequentemente, as ferramentas com o objetivo de transcrever dados de áudio, em vários idiomas e suas variantes, foram desenvolvidos. Além disso, conjuntos de dados contendo dados de áudio junto com seu texto transcrito são criados para treinar e avaliar os modelos utilizados por essas ferramentas de ASR. Este trabalho tem como objetivo comparar várias ferramentas de ASR em termos de desempenho de precisão, precificação e tempo de execução, para língua portuguesa em diferentes domínios. Além disso, mapeamos o ambiente de dados existente para a língua portuguesa e discutimos métricas para avaliação de ASR.*

1. Introduction

Tools that use voice resources, such as Siri [Apple 2022] and Google Voice [Google 2009], available on smartphones and other electronic devices are more present every day. That tools are changing the way people interact with devices present in their cars, homes and jobs. Therefore, academia and industry have become increasingly interested in the development of computational techniques that interprets as accurately as possible what the people are saying - be it a command or a conversation. Consequently, a lot of tools have been developed by industry and academia.

Automatic Speech Recognition (ASR), or Speech to Text (STT), is an interdisciplinary research area involving Computer Science and Computational Linguistic that focus on development of technologies to recognize and translate spoken language into text.

Machine learning techniques and statistical models (such as Hidden Markov Models - HMMs) have been widely used in many speech recognition systems for different languages. However, in addition to the works that propose new machine learning based techniques [Quintanilha et al. 2017],[Quintanilha et al. 2018] and statistical methods [Carvalho and Abad 2021] for ASR, due to the particularities of each language, some works in the literature has as purpose to improve computational techniques and to provide specialized resources for a particular language. [Batista et al. 2018b], [Neto et al. 2011], [Oliveira et al. 2012] are examples of researches carried out for speech recognition in Portuguese language, the focus of this work.

The vast amount of research on ASR conducted in the last years gave rise to several speech recognition tools/services. Although some of them have extensive documentation (eg., Google Cloud Speech API), there is a lack of works that compare the performance of that tools for Portuguese language in different domains and point out some aspects of each tool.

Therefore, this work intends to contribute to fill this gap by providing experimental results of a comparative study involving five well-known ASR tools provided either as service through an Application Programming Interface (Google Cloud, Microsoft Azure, Amazon Transcribe, Wit) or as an offline library (Vosk). This comparative study used 15,000 audio instances in Portuguese language obtained from four datasets related to different domains.

The remainder of this paper is organized as follows. Section 2 presents a brief review regarding automatic speech recognition, introduces the tools adopted in the comparative study, and details the three evaluation metrics used to assess the performance of each tool. Section 3 synthesizes the literature review research, focusing on works that evaluate ASR tools. Section 4, in turn, describes the methodology employed in this work. Details about data used in the experiments and the comparative analysis of experimental results are presented in Section 5. Finally, Section 6 draws conclusions and provides directions for future work.

2. Background

2.1. Automatic Speech Recognition

Automatic speech recognition can be seen as the technology that enables a speech-to-text conversion, i.e., a computer program generates text corresponding to the recognized speech [Ghai and Singh 2012].

Figure 1 illustrates a typical speech recognition process of probabilistic ASR systems, which is composed of the following components: feature extractor, acoustic model, language model and decoder.

In brief, an input speech signal is processed by a feature extractor to transform it into feature vectors that describe the characteristics of that input signal. Then, these feature vectors are mapped to phonemes by using an acoustic model, which represents the relationship between audio signals and phonemes of a language. After, using the acoustic and language models, the decoder searches for the sequence of words that best match the input feature vector sequence. A language model corresponds to a statistical model that provides a probability distribution over sequences of words and, therefore, it adds context by discarding unlikely word sequences given the language grammar rules and the subject of conversation. In the end, the output is a word sequence with the largest probability from acoustic and language models.

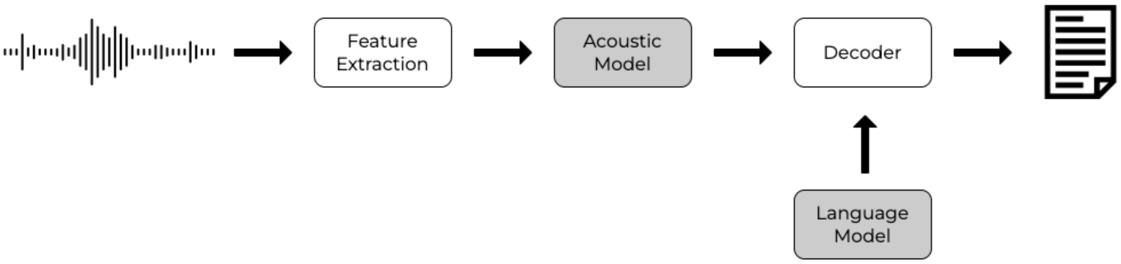

Figure 1. Automatic speech recognition pipeline

Despite of the historical success of ASR probabilistic systems, from 2012 on, with the increasing computational power in conjunction with the development of powerful graphics processing units (GPUs), several researchers started to apply only one neural based system, named end-to-end solution, to perform the speech recognition task.

2.2. ASR Tools

There are multiple automatic speech recognition tools available - engines that require an audio input and returns its transcription. The ones applied in this work are listed and briefly described below.

- Google Cloud Speech API: the Google Cloud Platform contains many resources, such as the Google Cloud Speech API. This resource supports 125 languages and variants [Google 2022b]. Its documentation is available at [Google 2022c] and a specific tutorial [Google 2022a], that was essential for the understanding and applying this tool.

- Azure Speech Service: Azure is Microsoft's cloud service, and it contains several resources, including Azure Cognitive Services [Microsoft 2022b] and, more specifically to this project, Azure Speech Service (Azure's speech to text tool). This resource supports 120 languages and variants [Microsoft 2022a].

- Amazon Transcribe: Amazon Web Services, or AWS, is Amazon's cloud service. This cloud also provides a transcription service (Amazon Transcribe) [Amazon Web Services 2022b], that supports 37 languages and variants [Amazon Web Services 2022d].

- Wit.ai: Wit.ai is a Natural Language service, owned by Facebook, that provides a speech-to-text service [Wit.ai 2022a]. This is a free service, even for corporate use, that supports 31 languages and their variants.

- Vosk API: Vosk API [Cephei 2022b] is an free open-source service, and works offline, that supports over 20 languages.

2.3. Evaluation metrics

Different evaluation metrics can be used to evaluate the performance of an ASR system. In this work, the goal of ASR systems evaluation is to provide a comparison between different systems for distinct domains.

All evaluation metrics presented in this section are calculated in function of hits and errors. A hit (H) happens whenever a word in the automatic transcription and in the reference get matched. Otherwise, we have an error. The three kinds of errors that can occur in automatic speech recognition, namely substitution, deletion and insertion, are detailed as follows:

- Substitution (*S*): when a word in the reference is transcribed as a different word (see Figure 2).

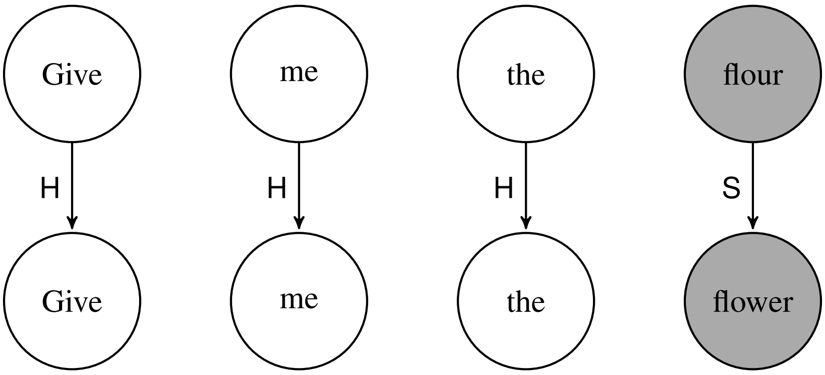

Figure 2. Example of substitution. Upper sentence: reference / Lower sentence: automatic transcription.

- Deletion (*D*): when a word in the reference is missed in the automatic transcription (see Figure 3).

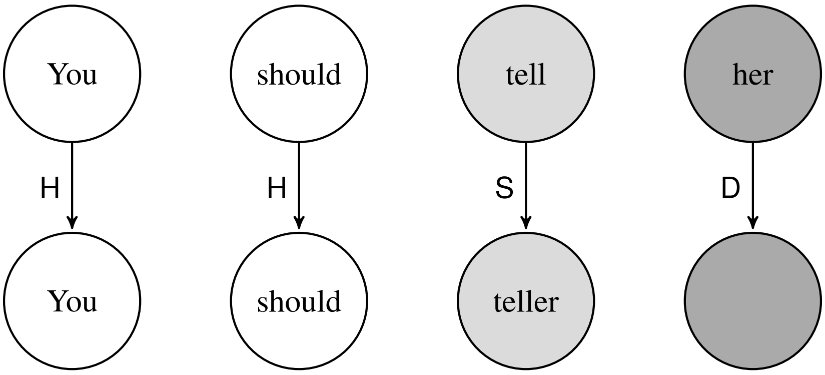

Figure 3. Example of deletion. Upper sentence: reference / Lower sentence: automatic transcription.

- Insertion (*I*): when a word, that does not exist in the reference, appears in the automatic transcription (see Figure 4).

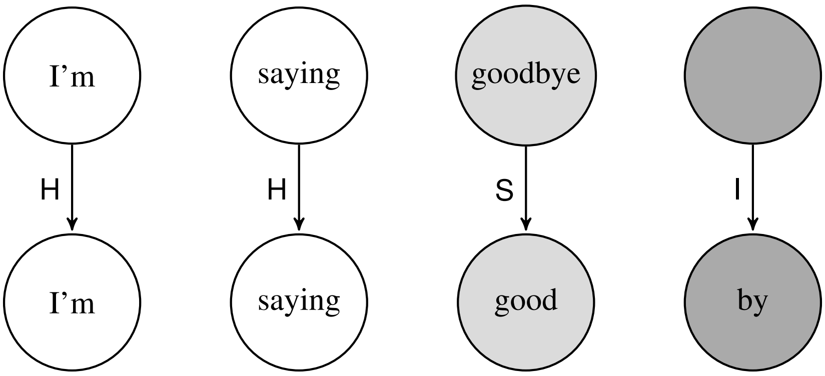

Figure 4. Example of insertion. Upper sentence: reference / Lower sentence: automatic transcription.

It is important to note that a primary issue related to ASR evaluation metrics calculation is obtaining a proper word alignment between the reference and the automatic transcription to count the number of hits and/or errors. To do so, usually an algorithm selects the sequence alignment for which the error score is minimized. Next, we present three metrics adopted in this work to evaluate ASR systems.

2.3.1. Word error rate

Word Error Rate (*WER*), the most popular evaluation metric for ASR assessment [Morris et al. 2004], is defined as:

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

where S is the total number of substitutions, D is the total number of deletions, I is the total number of insertions, and H is the total number of hits. Note that $H + S + D$ corresponds to the total number of words in the reference and $S + D + I$ corresponds to the total of errors produced by the transcription tool.

While this metric is useful for ASR performance systems comparisons, it is not appropriate to tell us how good a system is, since it has no upper bound, i.e., *WER* can surpass 100% in noisy conditions. Aiming at solving the limitations of *WER*, some alternative evaluation metrics proposed in the literature are presented below.

2.3.2. Match error rate

Considering N is the number of matched I/O word pairs (matches between the reference and the automatic transcription), Match Error Rate (*MER*) corresponds to the proportion of I/O word matches which are errors [Morris et al. 2004]. It is given by:

$$MER = \frac{S+D+I}{H+S+D+I} = 1 - \frac{H}{N}, \tag{2}$$

where S is the total number of substitutions, D is the total number of deletions, I is the total number of insertions, and H is the total number of hits. Note that $N = H+S+D+I$.

2.3.3. Word information lost

Word Information Lost (*WIL*) is a probabilistic metric that computes the proportion of word information lost due to errors. To do so, it takes into account the proportion of hits to the number of words in the reference and the proportion of hits to the number of words in the automatic transcription. This metric is defined [Morris et al. 2004] as:

$$WIL = 1 - \frac{H}{H+S+D} * \frac{H}{H+S+I}, \tag{3}$$

where S is the total number of substitutions, D is the total number of deletions, I is the total number of insertions, and H is the total number of hits.

3. Related work

Some ASR tools were evaluated by some papers, using datasets for their testing and metrics for their evaluation. The Table 1 summarises the quantity that each work features. Also, a brief description is elucidated below.

[Neto et al. 2011] implements and tests an automatic speech recognition traditional pipeline, constituted of an acoustic model (HTK), a statistical language model (SRI Language Modeling Toolkit), and a decoder (Julius and HDecode). This ASR pipeline is tested based on five datasets, three developed by this work (LapsNews, LapsBenchmark, and LapsStory) and two other domains (West Point and CETUC). Finally, WER was the only metric applied in this paper. The work also features many elucidations on audio instances and phoneme concepts.

[Oliveira et al. 2012] implements and tests an automatic speech recognition traditional pipeline, with a single tool for the process: CMU Sphinx. Four databases were used, three as input to build the acoustic model (West Point, LapsStory and CETUC) and one for evaluation of the model (LapsBenchmark). One metric (WER) was used for evaluation of results.

[Lima et al. 2021] is focused on experimenting with one dataset (verbal communication of the sector operation electric) on multiple automatic speech recognition (Vosk, IBM and Azure natively; Wit.ai, Google and Azure using Speech Recognition python library) tools for Portuguese and evaluating to present the best result in the context. The evaluation metrics used were WER, MER and WIL. The dataset used on this work was no longer available (January to April 2022).

Table 1. Related work summary

Reference	Tools	Datasets	Metrics
[Neto et al. 2011]	2 *	5	1
[Oliveira et al. 2012]	1	4	1
[Lima et al. 2021]	5	1	3
This work	**5**	**4**	**3**

Different to previous work, this study addresses five well-known ASR tools, four datasets, and three evaluation metrics, to evaluate each ASR tool for a variety of data domains.

4. Methodology

As aforementioned, this work aiming at carrying out a comparative study using five ASR tools. The evaluation of each ASR tool followed the pipeline presented in Figure 5. Shortly, given a dataset containing audio data along with their transcribed texts, here named references, the audio file is converted to WAV file format (if not already) before to be sent to an ASR tool for transcription. After transcription, two scenarios were adopted to evaluate the ASR tool's performance according to different evaluation metrics. In the first one, the raw references and the raw automatic transcriptions are compared to compute the evaluation metrics. In a different way, in the second scenario, the references and automatic transcriptions are post-processed before to be used to compute the evaluation metrics.

The post-processing step was carried out since some ASR tools do not include neither punctuation nor capital letters in their transcriptions, potentially increasing the error rate indicated by the evaluation metrics. Then, in the second scenario, the following post-processing tasks were conducted on the references and automatic transcriptions:

- The texts were converted to lower case;
- Multiple whitespaces between the words were replaced by a single whitespace;
- All punctuation was removed.

As the focus of this work is the evaluation of ASR tools for Portuguese language, 15,000 audio instances (along with their transcribed texts) were selected from four datasets related to different domains to provide a heterogeneous environment to evaluate the ASR tools. The characteristics of the adopted datasets and the criteria chosen for select the audio instances are described in Section 5.1.

Finally, to evaluate and compare the five ASR tools adopted in this work, named Google Cloud Speech API, Azure Speech Service, Amazon Transcribe, Wit.ai, and Vosk API, we used the evaluation metrics WER, MER, and WIL (see Section 2.3).

5. Experiments

5.1. Data

As stated before, we used the following four different datasets to select 15,000 instances produced in Portuguese language in our comparative experiment: Laps Benchmark [Falabrasil 2018a], Laps Mail [Falabrasil 2018b], VoxForge, and Common Voice Portuguese. The FalaBrasil group from the University of Pará [FalaBrasil 2020] provided the Laps Benchmark and the Laps Mail datasets. The VoxForge [VoxForge 2006] is a publicly available corpus proposed to be used in open source speech recognition tools. The last dataset, Common Voice [Foundation 2017], is made available by Mozilla as an open-source corpus for many languages.

VoxForge and Common Voice datasets were created and can be augmented by users interested in volunteering using a web platform. Both have a system where volunteers can record themselves reading some text and upload their recordings to the platform

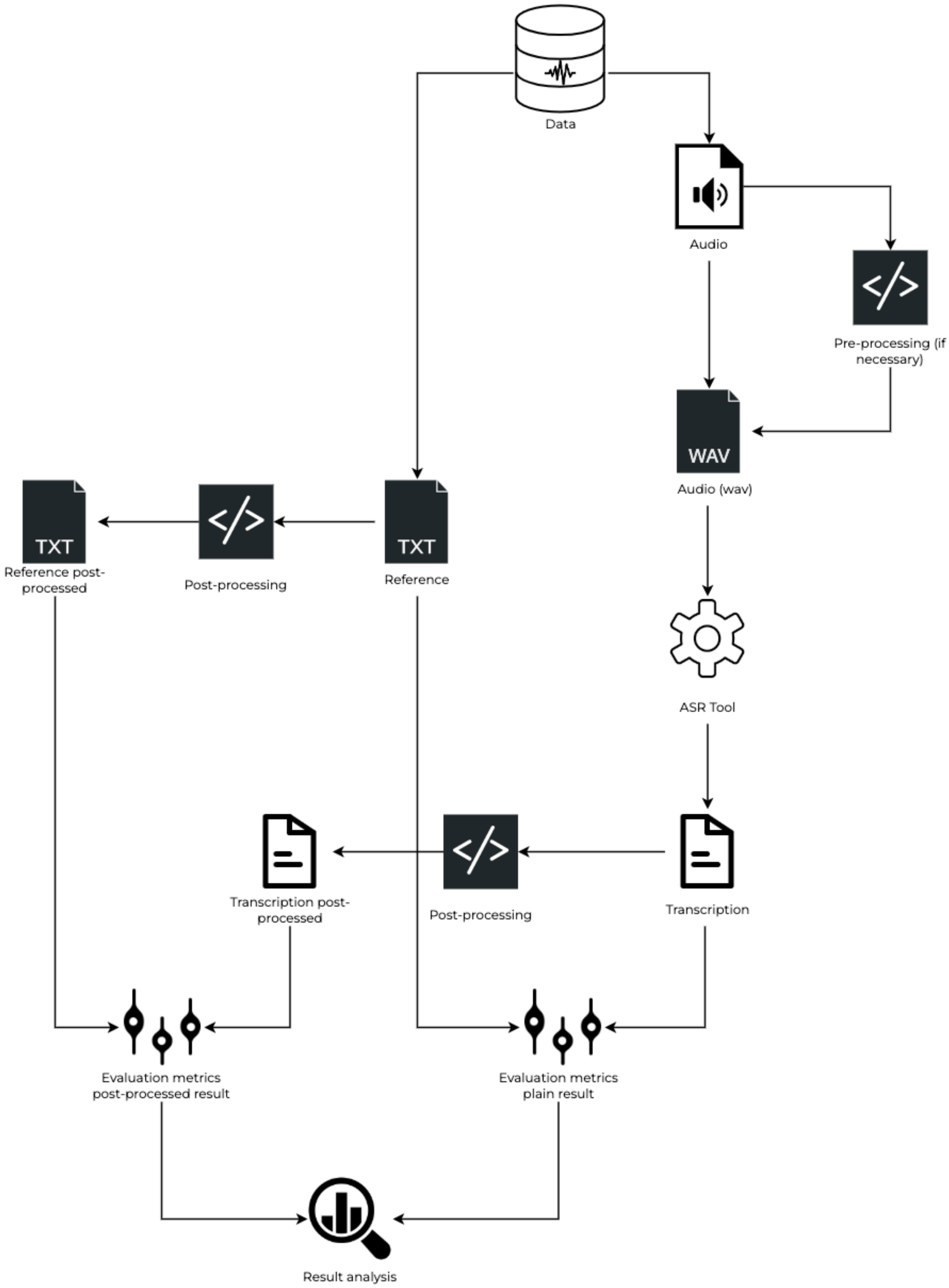
Data
Audio
Pre-processing (if necessary)
WAV
Audio (wav)
ASR Tool
TXT
Reference post-processed
Post-processing
TXT
Reference
Transcription post-processed
Post-processing
Transcription
Evaluation metrics post-processed result
Evaluation metrics plain result
Result analysis

Figure 5. Diagram demonstrating the work's processes

to be validated by other volunteers. In a different way, Laps Benchmark and Laps Mail datasets were created by a limited set of people from FalaBrasil's group.

The datasets characteristics regarding the production of the audios is shown in Table 2. The column ”Environment” presents the audio recording environment, which can be either controlled (with background noise controlled in the recording environment) or not controlled. The column ”Microphone” specifies the type of microphone used to record the audios. The number of different voices contained in the dataset is presented in the ”Number of speakers” column. Lastly, the ”Encoding” column depicts the original encoding of the audio data in each dataset.

Table 2. Datasets characteristics in January 2022

Dataset	Environment	Microphone	Number of speakers	Encoding
Common Voice	Not controlled	Computer	Not possible to estimate	mp3
Laps Benchmark	Not controlled	Computer	35	wav
Laps Mail	Not controlled	Shure PG30	25	wav
VoxForge	Not controlled	Computer	199 (estimate)	mp3

Other interesting metadata is the number of files and the duration (in seconds) of each dataset (considering the slice used). The number of files, seconds, and metrics related to seconds of the instances can be noticed in Table 3. Also, another interesting metadata regarding the datasets is that all audios have sample widths of 2 bytes and are mono-channel (the representation of sound is coming from or going to a single point). The last metadata is the frame rate, except for the Common Voice dataset, all of the audios have the value of 16000 Hz. About the Common Voice specifically, there are 4631 audio files with the frame rate of 32000 Hz, 3361 with 48000 Hz, and 1 with 44100 Hz.

In this case, considering computational power and running time all tests, 15,000 instances were selected. In consequence of that, a slice of the Common Voice dataset was selected - that is because, at the date of the extraction, the database was constituted of 95429 instances validated (the concept of validated, in this dataset, means that at least two users validated the instance and the maximum of one user invalidating the instance). In this case, to achieve maximum of database diversity, the priority conditions of the instances to constitute the slice of 7993 instances were:

1 - Gender (all female data was included, being 3751 instances);

2- Validation made by more than 2 users (all male data validated by more than 2 users was included);

3 - Age (the male data included, as far as possible, even data between ages of teens, twenties, until seventies).

Table 3. Total of files and seconds in each database in january 2022

	Duration (seconds)				Files
	Total	Max	Min	Median	Total
Dataset					
Common Voice	33370.29	10.48	0.90	3.89	7993
Laps Benchmark	3240.16	7.85	2.72	4.44	700
Laps Mail	5142.27	5.38	0.77	2.30	2176
VoxForge	15287.39	20.75	0.88	3.58	4131

5.1.1. Data inconsistency

It was expected each instance to have audio and transcription with the content of the unique language (in this case, Portuguese). But some inconsistencies were found as the analysis started and further evaluation of each instance was necessary as a consequence. First of all, were found 122 instances without transcription in the Laps Mail dataset. Secondly, VoxForge's dataset for Portuguese contains 40 instances with content in English. Table 4 has some examples with, respectively, the name of the files and the respective reference. Lastly, Common Voice's dataset for Portuguese also contains instances (a total of 6) with English content (files and respective reference in Table 5).

Although the instances may be spoken by Portuguese native speakers, the content is not in Portuguese - causing high error rate transcriptions by the tools. Noisy data was removed, changing the total of instances to 14,832.

Table 4. Sample of files with english content in VoxForge Portuguese's dataset

File	Sentence
voxforge_anonymous-20140619-wcy_ar-07.wav	One rainy day the rats heard a great noise in the loft.
voxforge_anonymous-20140619-wcy_ar-08.wav	The pine rafters were all rotten, so that the barn was rather unsafe.
voxforge_anonymous-20131016-uzv_ar-01.wav	Once there was a young rat named Arthur who never could make up his mind.

Table 5. Files with english content in Common Voice Portuguese's dataset

File	Sentence
common_voice_pt_30419254.wav	Orleans
common_voice_pt_28689859.wav	Selbach
common_voice_pt_19364113.wav	Alabama, Montgomery.
common_voice_pt_19426374.wav	Youtube Rewind.
common_voice_pt_28744887.wav	Major Sales
common_voice_pt_28925439.wav	General Maynard

5.2. Data processing

For this comparative work, there was the necessity to process the data and organize it in a certain pattern, since each dataset has its patterns - code available in [Oliveira 2021a]. The organization of the dataset consisted in standardizing the audio distribution in folders and the transcriptions in the same types of files, with the same types of structures. In this case, it was used a *csv* file containing two columns (name of the script and official transcription) for each dataset, and all of its files in one single folder. This organization also aimed to improve the iteration overall data, using only one generic script with parameters to specify the dataset.

Also, another interesting thing to point out is that both Common Voice and VoxForge datasets had their files transformed to *wav* files - because most tools had ease of usage when the files were in *wav* format rather than *mp3* format.

5.3. ASR Tools

The tools used in the experiments to transcribe audio into text were briefly introduced in Section 2.2. The Google Cloud Speech API, Azure Speech Service, and Amazon Transcribe were chosen considering that they are resources provided by great technology companies. The Wit.ai is also made available by a great company, but also, another reason for its inclusion in this work is [Lima et al. 2021], which shows the high performance of the tool in the data domain in which the work encompassed. About Vosk API, the advantages evaluated were that the tool can be used offline, is based on Kaldi models (that have a noticeable visibility [Povey et al. 2011], [Batista et al. 2018a]). Also, this tool was used in [Kolobov et al. 2021] with reasonable results for French, Spanish, Arabic, and Turkish.

Another resource used in this work was the Speech Recognition [Zhang 2014] library. The library supports several automatic speech recognition tools, online and offline. For this work, this library was applied with four ASR tools: Google Cloud Speech API, Azure Speech Service, Wit.ai, Vosk API. This library, other than centralizing tools in a unique library, also transforms the audio data into a *flac* format, being a factor to possibly differentiate the metrics result when compared to the native libraries. The contribution made by this work to the open-source library in [Zhang 2022] - fix of a bug related to the application of Google Cloud Speech API.

To apply Google Cloud Speech API in its native form it was necessary to create an account (the platform provides a free trial of three months), create a bucket, create an access key and use the python library [APIs 2022] to execute the tool. A specific requirement about the tool is that it was necessary to upload the audio to be transcribed. When applying Google Cloud Speech API using Speech Recognition, it was also necessary to create an account and an access key. Then, it was necessary to make use of the Speech Recognition specific function for Google passing the access key as a parameter. The script created to obtain the transcriptions using this ASR tool is available in [Oliveira 2022b].

About Azure Speech Service, this tool was used only with Speech Recognition. To use this resource, it was necessary to create an account (the platform provides credits for new users or students), create the Cognitive Services resource, and generate its key. Then, it was necessary to employ the Speech Recognition specific function for Azure passing the key as a parameter.

To use Amazon Transcribe, it was necessary to create an account (the platform provides a free tier [Amazon Web Services 2022a] - in this case, the free tier was used only regarding the storage), create a bucket, create an access key and use the python library [Amazon Web Services 2022c] to execute the tool. A specific requirement about this tool is that for the usage of this service is that the audio to be transcribed has to be in a bucket. For that reason, the audio needs to be uploaded. The script created to obtain the transcriptions using this ASR tool is available in [Oliveira 2022a].

Regarding Wit.ai, it was necessary to create a Facebook account, create an access key and use the python library [Wit.ai 2022b] to execute the tool. When applying Wit.ai using Speech Recognition, it was also necessary to create an Facebook account and an access key. Then, it was necessary to make use of the Speech Recognition specific function for Wit.ai passing the access key as a parameter. The script created to obtain the transcriptions using this ASR tool is available in [Oliveira 2022c].

Finally, concerning Vosk API, it was necessary to use the python library [Cephei 2022a] and to use a language model. In the case of Portuguese, there is a model available on the tool's website [Cephei 2022c], but the tool supports other adapted models. When applying Vosk API using Speech Recognition, it was also required a model for the language (the same previously described, in this case). Then, it was necessary to make use of the Speech Recognition specific function for Vosk passing the model folder as a parameter. The script created to obtain the transcriptions using this ASR tool is available in [Oliveira 2021b].

5.4. Results

For each tool transcribing the data of each dataset, the metrics (described in Section 2.3) were calculated (using the python library Jiwer [Jitsi 2022]) and presented as a violin plot [Hintze and Nelson 1998] - Figs. 7 to 30. The order of presentation take into account primarily the tool and secondarily the metric (Word Error Rate, Match Error Rate, and Word Information Lost respectively). Each violin plot has the x-axis as the dataset and is grouped by the plain result (comparison of the plain result returned by the tool and the transcription of the instance) and the post-processed data (described in - Section 5.4.2).

Although not as popular as the boxplot and very similar to it, the violin plot was chosen considering the abundance of significant information in this analysis (that both intercept) and the data's distribution. The information's interpretation of the violin plot compared to the boxplot can be seen in Figure 6. Some important data presented in this plot is the median (white dot), the interquartile range (black wider bar in the center of the plots), outliers (distributed data outside the narrow centered line) and entire data distribution (the violin plot around the centered line).

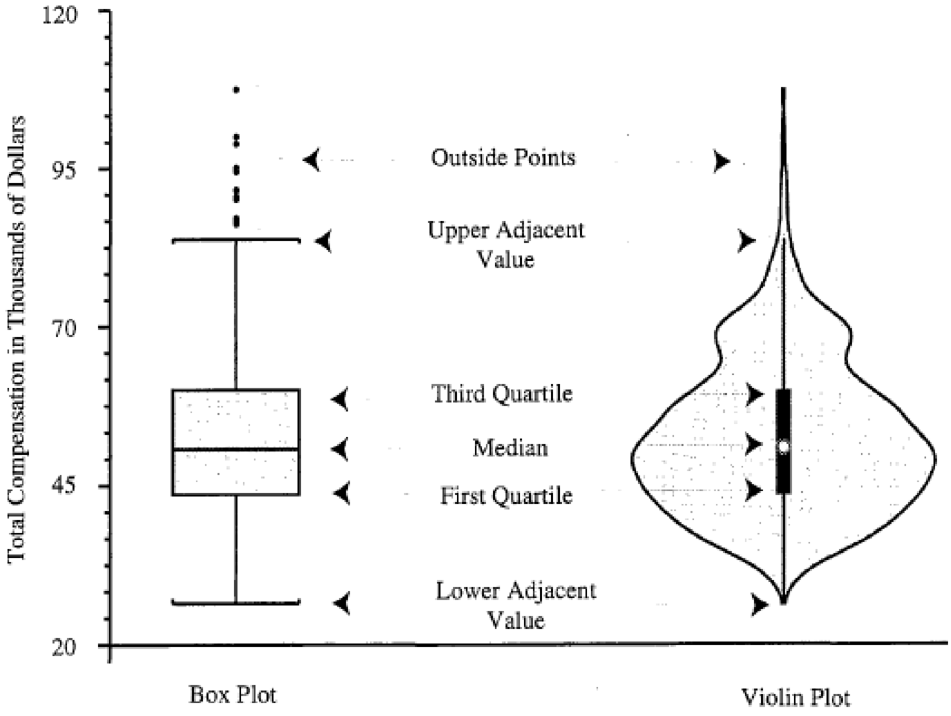

Figure 6. Reading box plot vs violin plot [Hintze and Nelson 1998]

Also, to summarise the results, Table 6 represents the performance of each tool. The value was calculated as the average of the order that which each tool appeared considering its average WER, MER, or WIL value resulting from the transcription of each dataset (see Appendix to see each average of each metric per tool per dataset).

The result of Table 6 is not only considering the quality of the tool's transcriptions - tools with lower error rates would appear high on the ranking of the averages, but also consistency - the order is taken into account was by each dataset. For example, Google Cloud Speech API (Speech Recognition) tool appeared in second place for Common Voice, LaPS Mail, LaPS Benchmark, and fifth for VoxForge - with this result constant for all three metrics considered. In this case, the value in the ranking is 2,75 because (2 + 2 + 2 + 5) / 4 - as the result was the same for each metric.

Table 6. Ranking representing the performance of each ASR tool

ASR Tool	Ranking
Google Cloud Speech API (Speech Recognition)	2.75
Vosk API (Speech Recognition)	3.50
Microsoft Azure Speech (Speech Recognition)	4.00
Wit.ai	4.08
Wit.ai (Speech Recognition)	4.42
Amazon Transcribe	4.50
Google Cloud Speech API	4.75
Vosk API	8.00

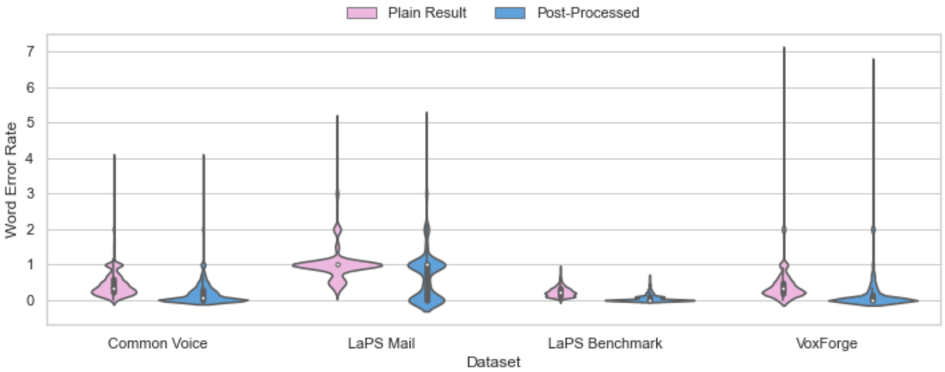

Figure 7. Word Error Rate considering Amazon Transcribe tool

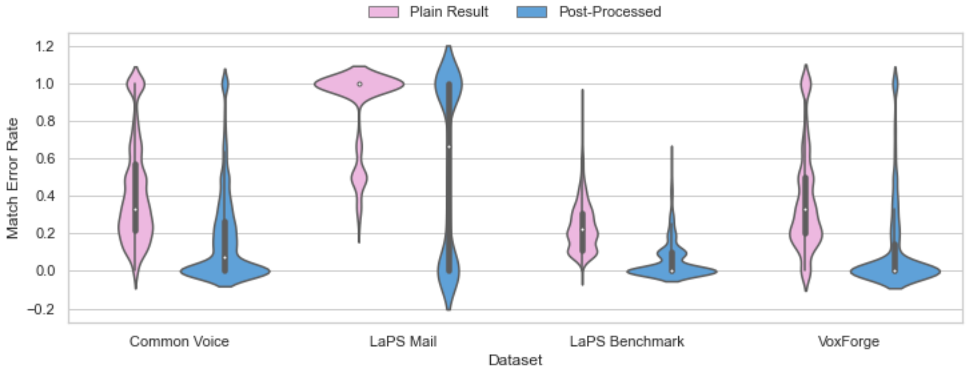

Figure 8. Match Error Rate considering Amazon Transcribe tool

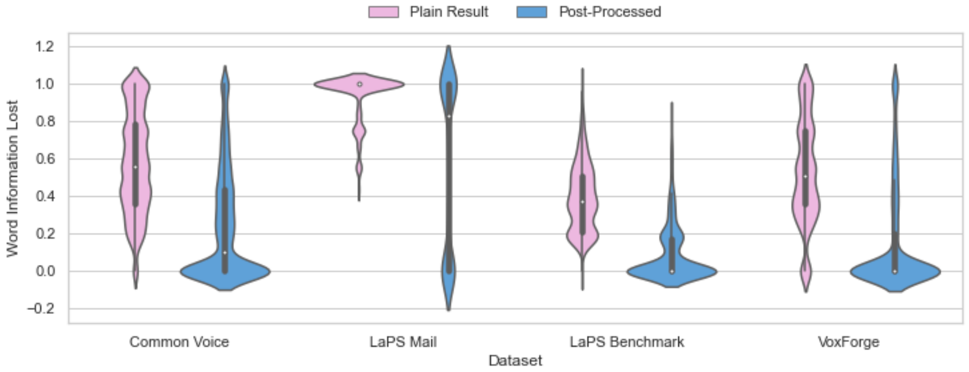

Figure 9. Word Information Lost considering Amazon Transcribe tool

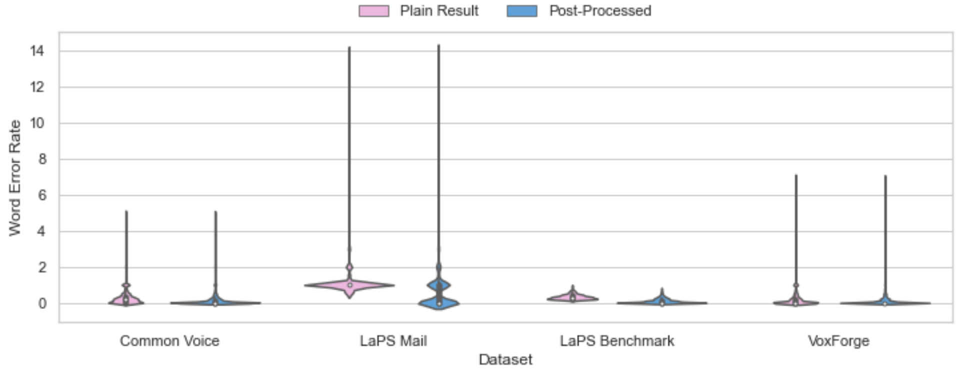

Figure 10. Word Error Rate considering Azure Speech Service (Speech Recognition) tool

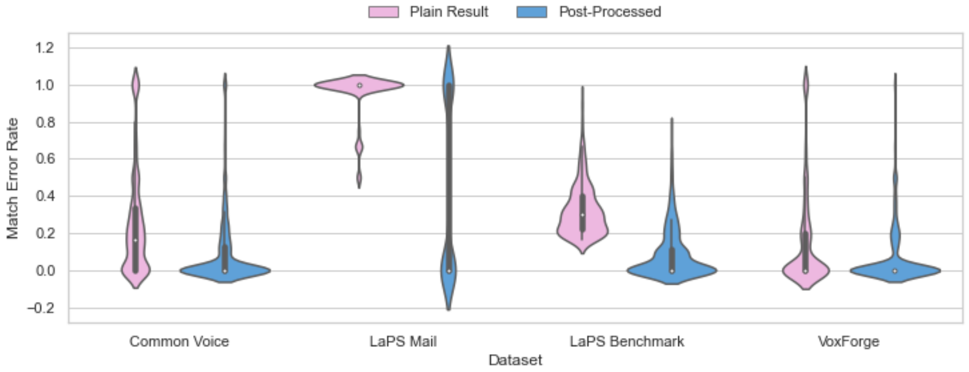

Figure 11. Match Error Rate considering Azure Speech Service (Speech Recognition) tool

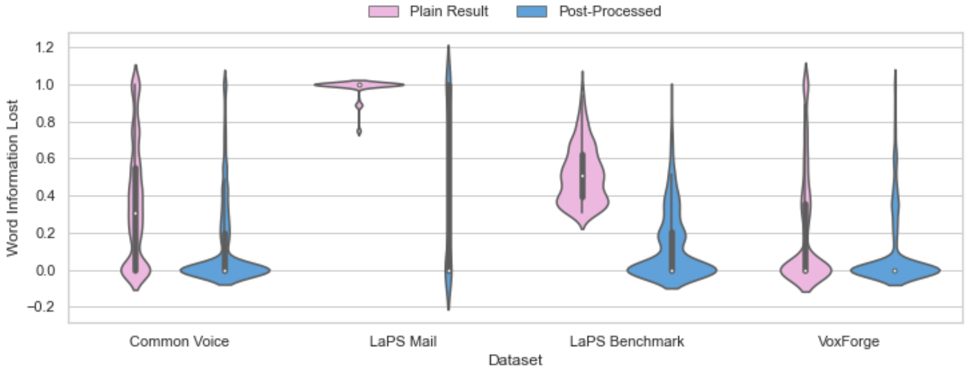

Figure 12. Word Information Lost considering Azure Speech Service (Speech Recognition) tool

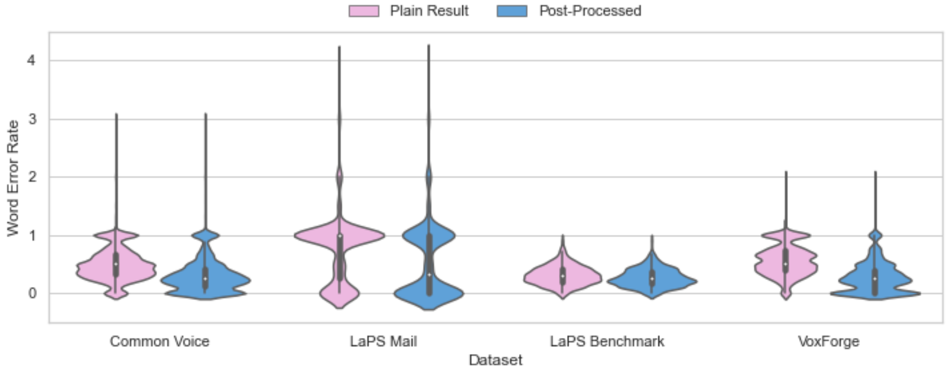

Figure 13. Word Error Rate considering Google Cloud Speech API tool

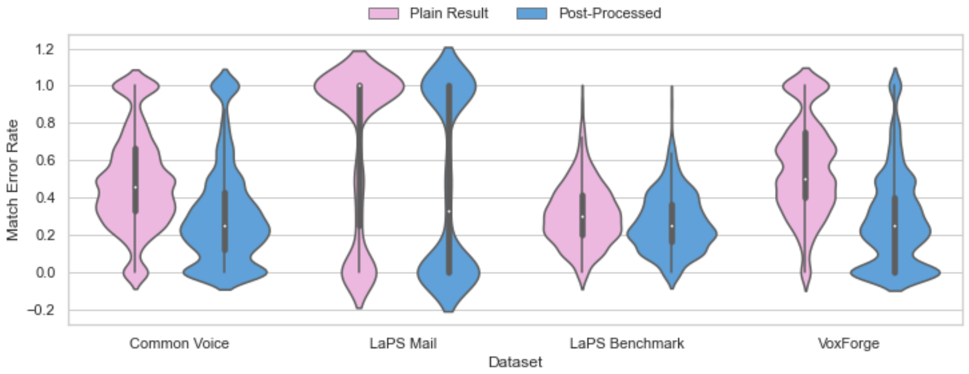

Figure 14. Match Error Rate considering Google Cloud Speech API tool

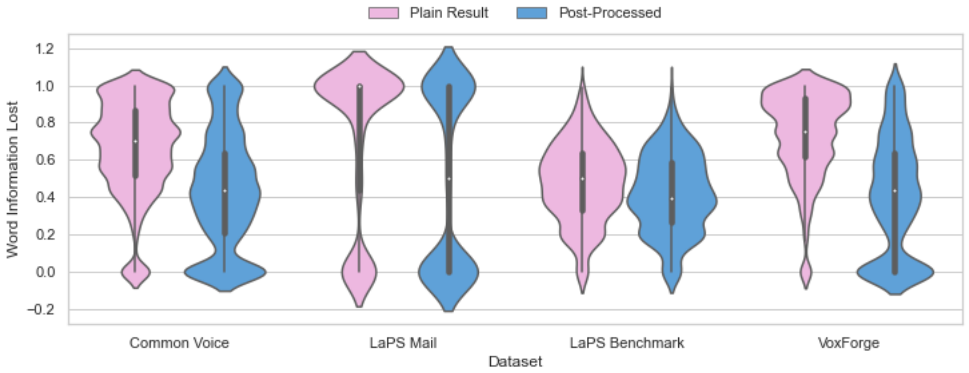

Figure 15. Word Information Lost considering Google Cloud Speech API tool

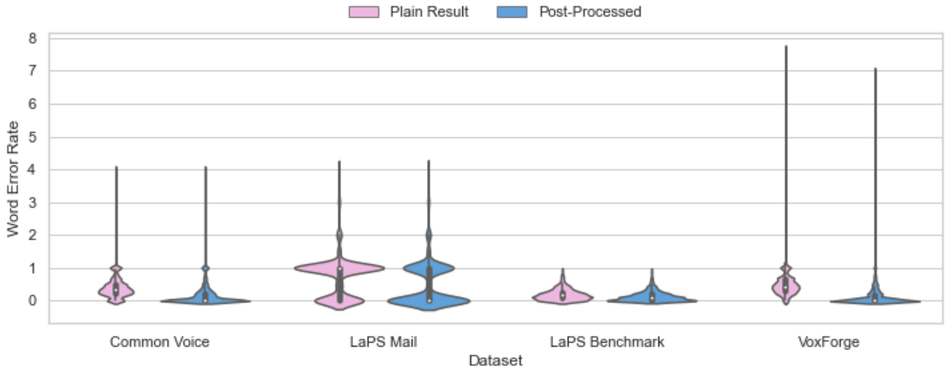

Figure 16. Word Error Rate considering Google Cloud Speech API (Speech Recognition) tool

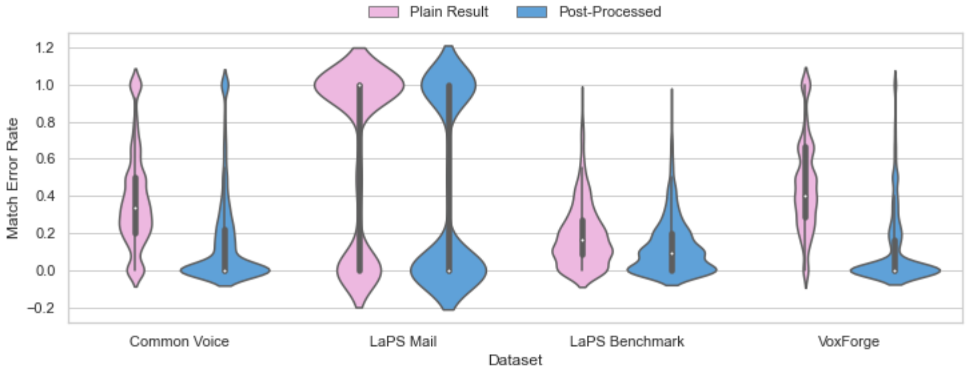

Figure 17. Match Error Rate considering Google Cloud Speech API (Speech Recognition) tool

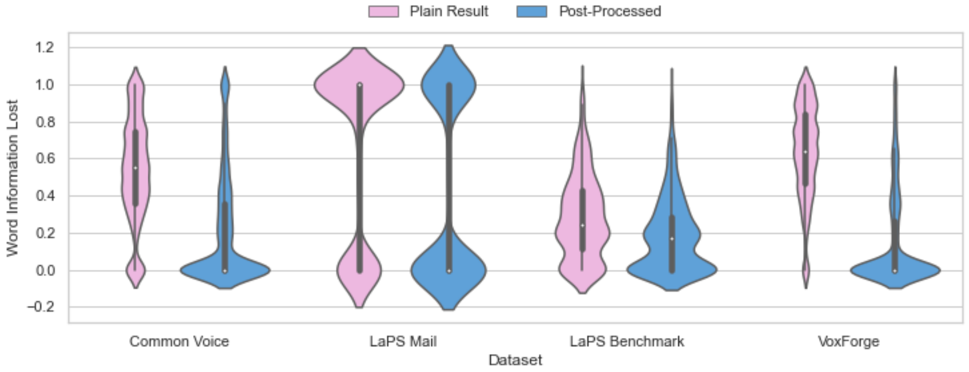

Figure 18. Word Information Lost considering Google Cloud Speech API (Speech Recognition) tool

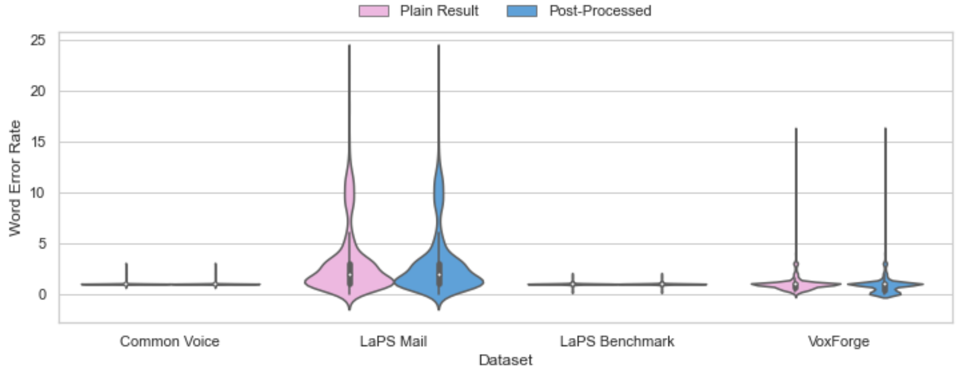

Figure 19. Word Error Rate considering Vosk API tool

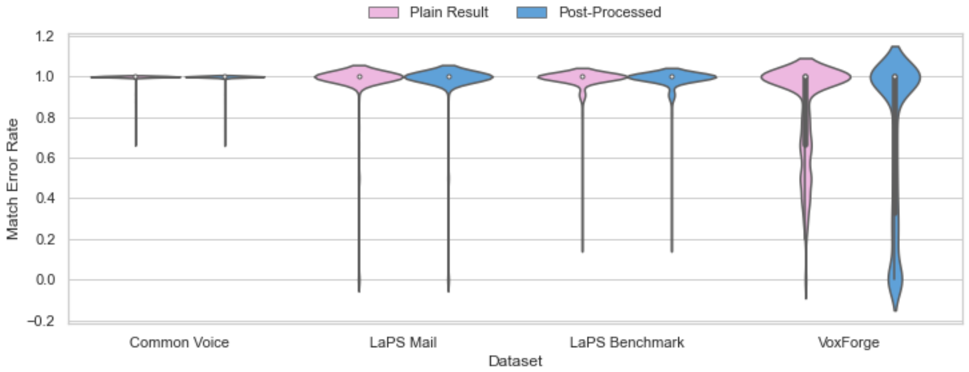

Figure 20. Match Error Rate considering Vosk API tool

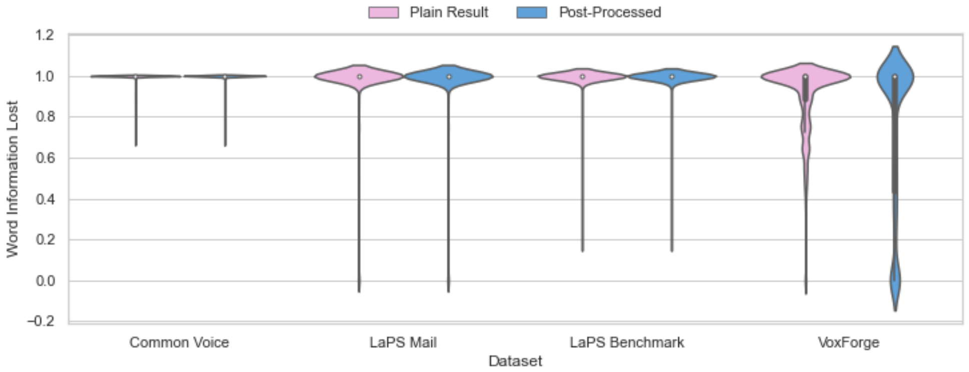

Figure 21. Word Information Lost considering Vosk API tool

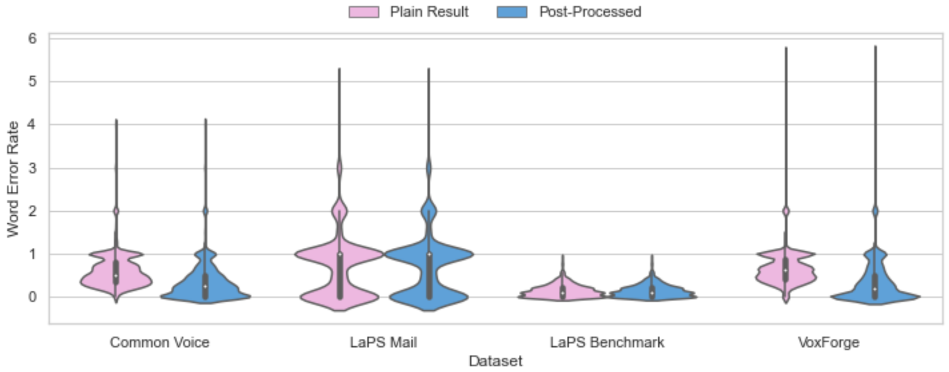

Figure 22. Word Error Rate considering Vosk API (Speech Recognition) tool

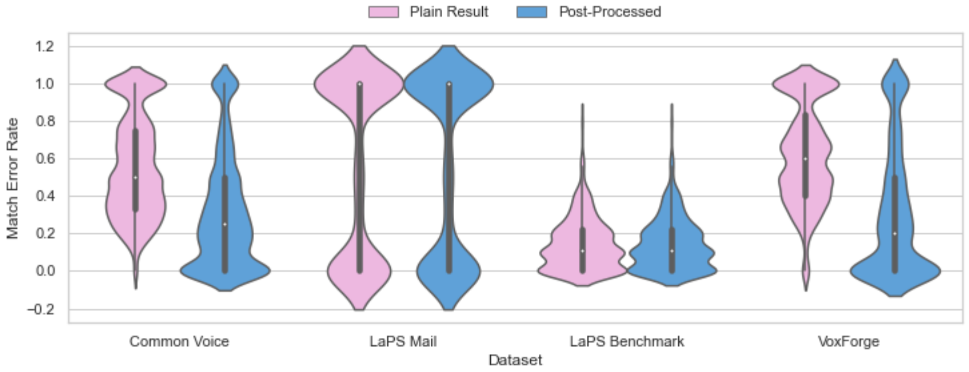

Figure 23. Match Error Rate considering Vosk API (Speech Recognition) tool

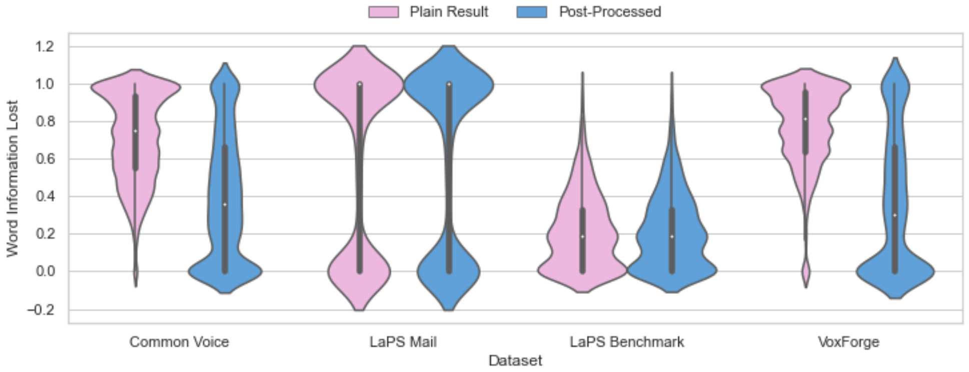

Figure 24. Word Information Lost considering Vosk API (Speech Recognition) tool

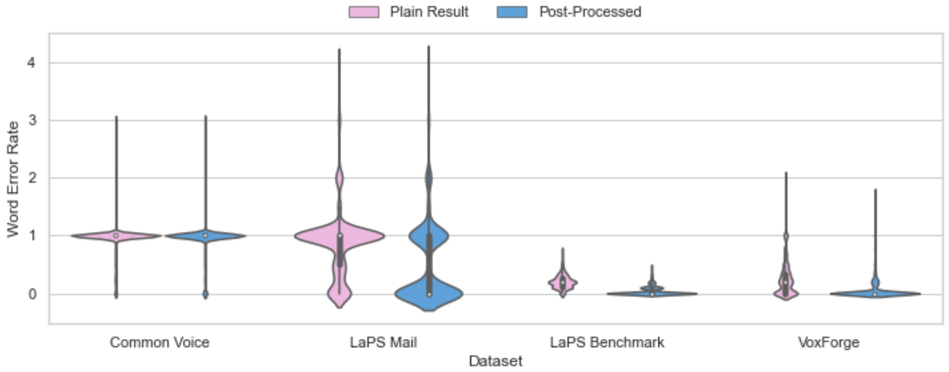

Figure 25. Word Error Rate considering Wit.ai tool

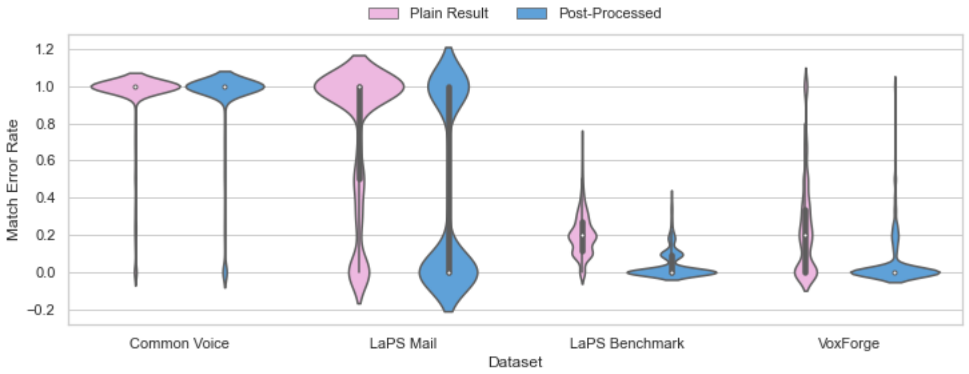

Figure 26. Match Error Rate considering Wit.ai tool

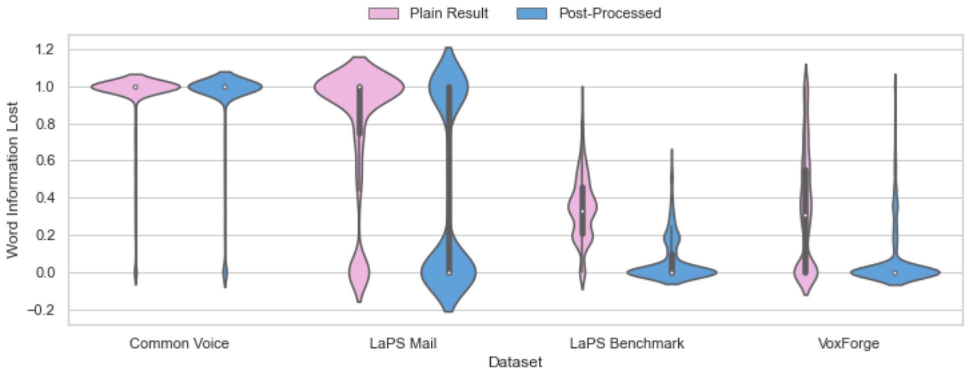

Figure 27. Word Information Lost considering Wit.ai tool

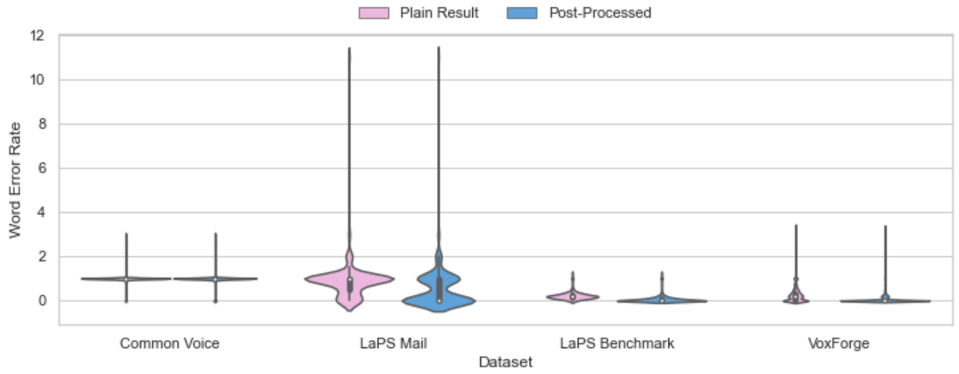

Figure 28. Word Error Rate considering Wit.ai (Speech Recognition) tool

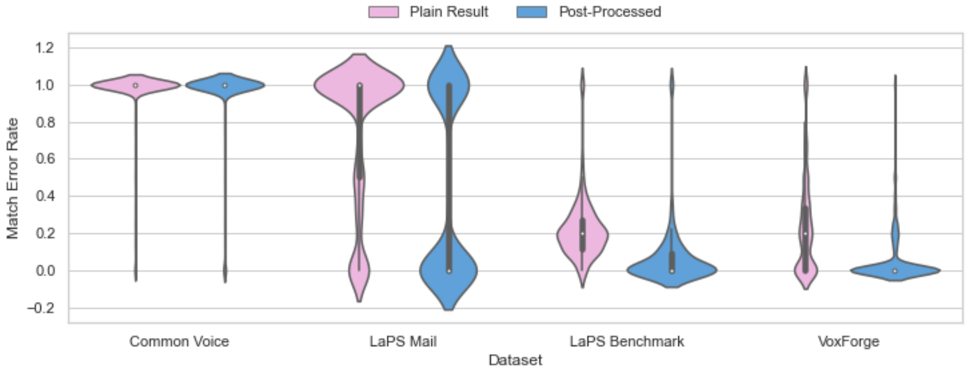

Figure 29. Match Error Rate considering Wit.ai (Speech Recognition) tool

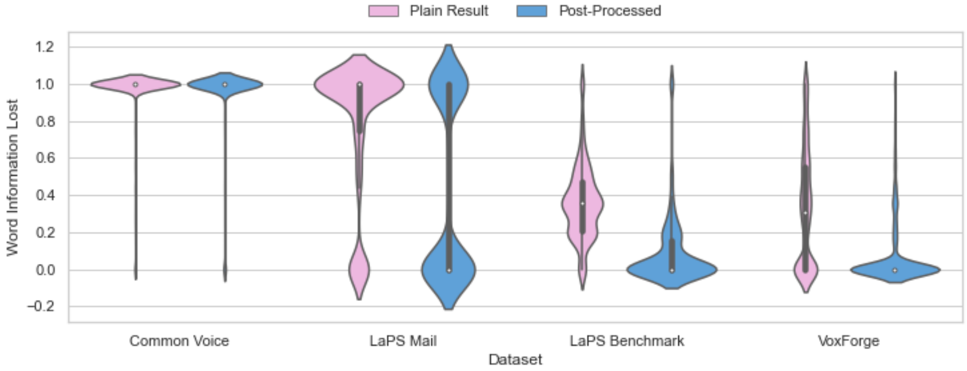

Figure 30. Word Information Lost considering Wit.ai (Speech Recognition) tool

5.4.1. Further evaluation regarding Wit.ai tool

This evaluation was instigated considering that the Wit.ai tool did not appear in the beginning of the ranking on Table 6 - diverging from a similar study [Lima et al. 2021] for Portuguese. Was noticed a large amount of null returns from this tool. To understand the quality better the not null results were isolated and plotted for each metric considering the native tool (Figure 31, Figure 32 and Fig 33) and the Speech Recognition tool (Figure 34, Figure 35 and Figure 36).

A great difference in the whole data plot and the not null data can be noticed in the data distribution of all metrics regarding the Common Voice dataset. The tool has a relatively good result but not for all instances - possibly not instances recorded in computers by ordinary speakers being those strong characteristics of Common Voice's dataset.

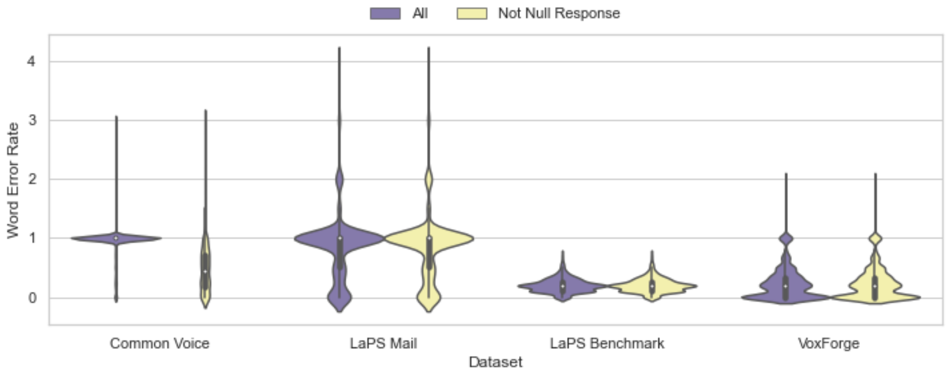

Figure 31. Word Error Rate considering Wit.ai tool - isolating not null data

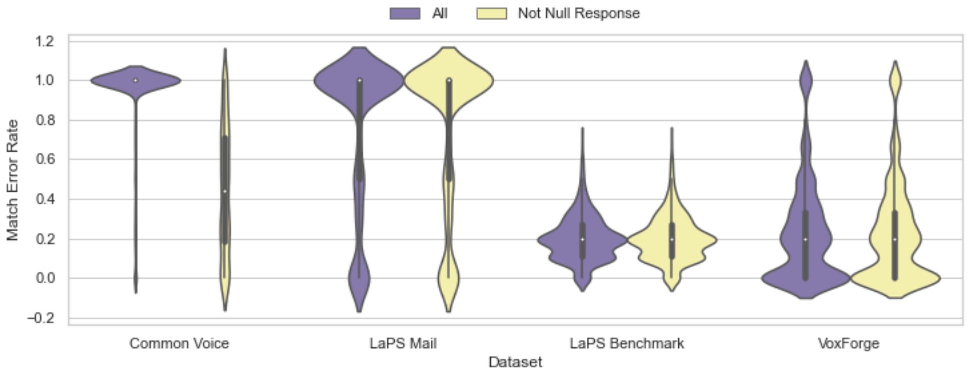

Figure 32. Match Error Rate considering Wit.ai tool - isolating not null data

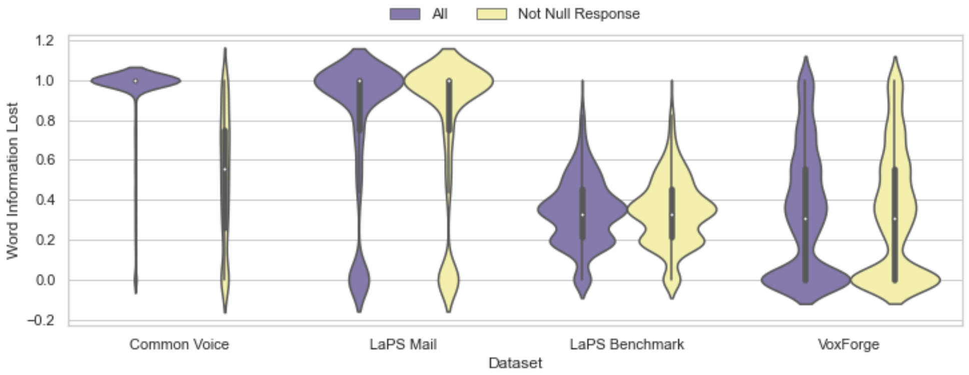

Figure 33. Word Information Lost considering Wit.ai tool - isolating not null data

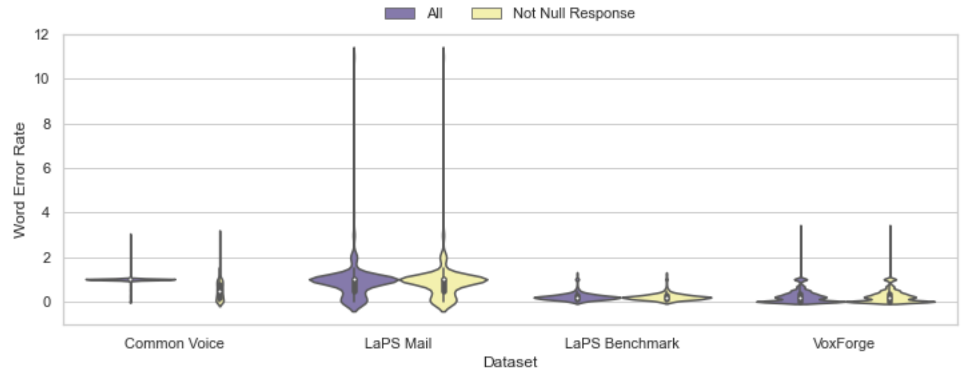

Figure 34. Word Error Rate considering Wit.ai (Speech Recognition) tool - isolating not null data

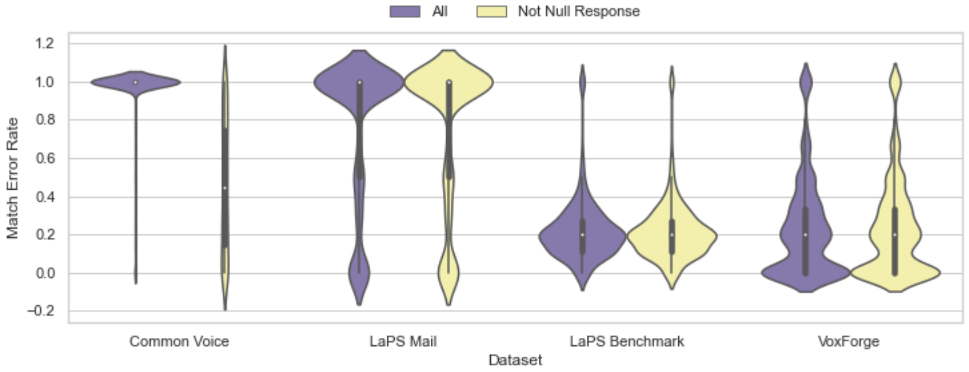

Figure 35. Match Error Rate considering Wit.ai (Speech Recognition) tool - isolating not null data

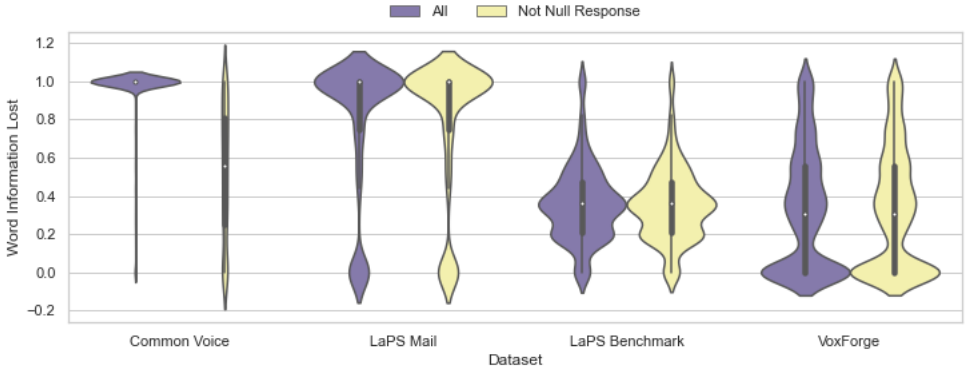

Figure 36. Word Information Lost considering Wit.ai (Speech Recognition) tool - isolating not null data

5.4.2. Pre-processing

Google Cloud Speech API, Vosk API, and Wit.ai were used with its native library and with the Speech Recognition library. The results showed some difference, having a lower error with the Speech Recognition library.

It was possible to further investigate the code - as it is open source - because no documentation about the library had any enlightenment on the transformations. It was noted that, in all cases, some kind of pre-processing on the audio is executed - as the input is transformed in a class where the object is evaluated regarding its sample rate and sample width. Even though is understandable that a pre-processing is present, it is not clear what exactly causes the difference in the results - there was no response about it as the author was contacted.

Vosk API's results were the most impacted by the pre-processing. The metrics can be seen in the Figure19, Figure20 and Figure20 for respectively WER, MER, and WIL metrics with the native Vosk API tool. In the Figure22, Figure23 and Figure23, Speech Recognitions Vosk API's tool is exhibited with lower medians and higher distribution of the metric - whereas in the previous graphs the result is focused on a 1.0 value of error.

5.4.3. Post-processing

As previously mentioned (see Section 4), the post-processing step applied in the references and in the automatic transcriptions aimed to remove all punctuation, substitute multiple whitespaces with single whitespace, and convert the texts to lower case.

The consequence of this specific result is explicit in all Figs. 7 to 30, as the Post-Processed label. In some cases, such as Google Cloud Speech (native approach) - Figure 13, Figure 14 and Figure 15 - particularly on VoxForge's and Common Voice's datasets as the distribution of the data reaches lower values and also the median, consequently.

5.5. Execution Time

This section presents the information on the execution time for each tool transcribing each dataset. Is presented in Table 7 the total of hours that each tool took to transcribe all of the instances described in 5.1. Also, this Table is presented the average time (in seconds) that the tool took to process each file. In Table 8 is presented for each tool the execution time of all data used in this paper from each dataset - as the columns.

All experiments were carried out on a PC with a 3.3GHz Intel Xeon E-2136 processor and 64 GB of RAM.

Table 7. Execution time for each tool

ASR Tool	Total	Avg por file (seconds)
Amazon Transcribe	73:10:14	17.56
Azure Speech Service (Speech Recognition)	8:55:28	2.14
Vosk API	2:28:21	0.59
Vosk API (Speech Recognition)	3:35:48	0.86
Wit.ai	9:36:27	2.31
Wit.ai (Speech Recognition)	10:02:13	2.41
Google Cloud Speech API	9:22:04	2.25
Google Cloud Speech API (Speech Recognition)	8:08:02	1.95

Table 8. Execution time for each dataset by each tool

ASR Tool	LaPS (both)	VoxForge	Common Voice
Amazon Transcribe	13:40:06	19:31:00	39:59:07
Azure Speech Service (Speech Recognition)	1:33:12	2:18:31	5:03:44
Vosk API	0:13:42	0:28:55	1:45:43
Vosk API (Speech Recognition)	0:30:20	0:53:28	2:11:59
Wit.ai	1:15:55	1:57:42	6:22:49
Wit.ai (Speech Recognition)	1:53:56	2:02:35	6:05:41
Google Cloud Speech API	1:36:28	2:27:15	5:18:21
Google Cloud Speech API (Speech Recognition)	1:14:14	2:11:15	4:42:32

5.6. Pricing

Both Vosk API and Wit.ai tools are free regardless of the sample's size. The cloud services (Microsoft, Google, and Amazon) charge for the transcription of audios. Table 9 contains the specific price of this experiment - executed in march of 2022. Each tool is followed by the total charge and the charge for an hour of audio transcribed.

Table 9. Pricing of all tools used in the experiments (march 2022)

ASR Tool	Total	Per hour
Amazon Transcribe	USD 101.74	USD 6.42
Azure Speech Service (Speech Recognition)	USD 17.96	USD 1.13
Vosk API	0.0	0.0
Vosk API (Speech Recognition)	0.0	0.0
Wit.ai	0.0	0.0
Wit.ai (Speech Recognition)	0.0	0.0
Google Cloud Speech API	USD 104.08	USD 6.57
Google Cloud Speech API (Speech Recognition)	USD 104.09	USD 6.57

6. Conclusions

It is not possible to conclude that one tool is better than the other but to conclude that there are some key points to evaluate when applying in a context. Also, it is important to point out that this study only focused on Portuguese, and even though aimed for heterogeneity in the data, there are contexts not approached that can have a significant difference in the results.

Regarding the quality of the transcription, both Google Cloud Speech API (Speech Recognition) and Vosk API (Speech Recognition) had great results (Table 6) regarding lower errors measured by the three evaluation metrics and consistency in the transcriptions of four datasets with different audio content and pattern.

Concerning the execution time, Vosk API (native and with Speech Recognition) had the best time, with an average of less than one second to transcribe each audio. Lastly, regarding the pricing, both Vosk API and Wit.ai are free, considering any context. In the cloud context, Azure Speech Service had the best pricing, with USD 1,13 per hour of audio transcribed.

Also, a detail to be considered while applying automatic speech recognition is the pre-processing of the audio and the post-processing of the reference and transcription. Both processings have decreased significantly the error rates in all WER, MER, and WIL metrics for all datasets. Pre-processing techniques can be applied to all data, while post-processing can be more difficult to apply in contexts where punctuation and case sensitive is necessary.

Regarding the data, the lack of pattern between datasets could be a problem while running tests, so a repository aiming to organize the data in a unique pattern was essential. Also, the reference of the data's instances appeared to have a significant number of null data and data in other languages - showing the importance of ensuring data quality.

In respect of adding value to the comparative work, a future work would be to include more ASR tools in the comparative evaluation. Also, would be interesting to incorporate data aiming even more heterogeneity taking into consideration domains, gender, age and accent.

References

Amazon Web Services, I. (2022a). Amazon free tier. https://aws.amazon.com/free. Accessed on 26.03.2022.

Amazon Web Services, I. (2022b). Amazon transcribe documentation. https://docs.aws.amazon.com/transcribe/index.html. Accessed on 26.03.2022.

Amazon Web Services, I. (2022c). The aws sdk for python. https://pypi.org/project/boto3/. Accessed on 26.03.2022.

Amazon Web Services, I. (2022d). Supported languages. https://docs.aws.amazon.com/transcribe/latest/dg/supported-languages.html. Accessed on 26.03.2022.

APIs, G. (2022). Google cloud speech api client library. https://pypi.org/project/google-cloud-speech/. Accessed on 26.03.2022.

Apple (2022). Siri. https://www.apple.com/br/siri/. Accessed on 24.03.2021.

Batista, C., Dias, A., and Neto, N. (2018a). Baseline acoustic models for brazilian portuguese using kaldi tools. pages 77–81.

Batista, C., Dias, A. L., and Neto, N. S. (2018b). Baseline acoustic models for brazilian portuguese using kaldi tools.

Carvalho, C. and Abad, A. (2021). Tribus: An end-to-end automatic speech recognition system for european portuguese.

Cephei, A. (2022a). Offline open source speech recognition api based on kaldi and vosk. https://pypi.org/project/vosk/. Accessed on 30.12.2021.

Cephei, A. (2022b). Vosk offline speech recognition api. https://alphacephei.com/vosk/. Accessed on 21.12.2021.

Cephei, A. (2022c). Vosk offline speech recognition api - models. https://alphacephei.com/vosk/models. Accessed on 30.12.2021.

Falabrasil, G. (2018a). Laps benchmark 16k. https://gitlab.com/fb-audio-corpora/lapsbm16k. Accessed on 12.09.2021.

Falabrasil, G. (2018b). Laps mail 16k. https://gitlab.com/fb-audio-corpora/lapsmail16k. Accessed on 12.09.2021.

FalaBrasil, G. (2020). Grupo falabrasil. https://ufpafalabrasil.gitlab.io/. Accessed on 12.09.2021.

Foundation, T. M. (2017). Common voice by mozilla. https://commonvoice.mozilla.org/pt/datasets. Accessed on 12.09.2021.

Ghai, W. and Singh, N. (2012). Literature review on automatic speech recognition. *International Journal of Computer Applications*, 41:42–50.

Google (2009). Google voice. https://voice.google.com/u/0/about. Accessed on 24.03.2021.

Google (2022a). Language support — cloud speech-to-text documentation — google cloud. https://cloud.google.com/speech-to-text/docs/languages. Accessed on 12.09.2021.

Google (2022b). Quickstart: Use client libraries — cloud speech-to-text documentation — google cloud. https://cloud.google.com/speech-to-text/docs/quickstart-client-libraries. Accessed on 12.09.2021.

Google (2022c). Speech-to-text: Automatic speech recognition — google cloud. https://cloud.google.com/speech-to-text. Accessed on 12.09.2021.

Hintze, J. L. and Nelson, R. D. (1998). Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184.

Jitsi, I. (2022). Jiwer. https://pypi.org/project/jiwer/1.2/. Accessed on 12.09.2021.

Kolobov, R., Okhapkina, O., Omelchishina, O., Platunov, A., Bedyakin, R., Moshkin, V., Menshikov, D., and Mikhaylovskiy, N. (2021). Mediaspeech: Multilanguage asr benchmark and dataset.

Lima, M., Coelho, B., and Takigawa, F. (2021). Ferramentas e recursos disponíveis para reconhecimento de fala em português brasileiro. pages 475–479.

Microsoft (2022a). Language and voice support for the speech service. `https://docs.microsoft.com/pt-br/azure/cognitive-services/speech-service/language-support#speech-to-text`. Accessed on 26.03.2022.

Microsoft (2022b). What are azure cognitive services? `https://docs.microsoft.com/en-us/azure/cognitive-services/what-are-cognitive-services`. Accessed on 26.03.2022.

Morris, A., Maier, V., and Green, P. (2004). From wer and ril to mer and wil: improved evaluation measures for connected speech recognition.

Neto, N., Patrick, C., Klautau, A., and Trancoso, I. (2011). Free tools and resources for brazilian portuguese speech recognition. *Journal of the Brazilian Computer Society*, 17.

Oliveira, A. R. G. (2021a). Processing audio files. `https://github.com/alinerguio/processing-data`. Accessed on 12.09.2021.

Oliveira, A. R. G. (2021b). Vosk api usage for portuguese. `https://github.com/alinerguio/vosk-transcript-tool`. Accessed on 31.12.2021.

Oliveira, A. R. G. (2022a). Aws transcribe usage for portuguese. `https://github.com/alinerguio/aws-transcript-tool`. Accessed on 26.03.2022.

Oliveira, A. R. G. (2022b). Google cloud speech api usage for portuguese. `https://github.com/alinerguio/google-transcript-tool`. Accessed on 26.03.2022.

Oliveira, A. R. G. (2022c). Wit.ai usage for portuguese. `https://github.com/alinerguio/wit-ai-transcript-tool`. Accessed on 26.03.2022.

Oliveira, R., Batista, P., Neto, N., and Klautau, A. (2012). Baseline acoustic models for brazilian portuguese using cmu sphinx tools. volume 7243 LNAI.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., and Vesel, K. (2011). The kaldi speech recognition toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

Quintanilha, I., Biscainho, L., and Netto, S. (2017). Towards an end-to-end speech recognizer for portuguese using deep neural networks.

Quintanilha, I. M., Biscainho, L. W. P., and Netto, S. L. (2018). A new automatic speech recognizer for brazilian portuguese based on deep neural networks and transfer learning. In *Congreso Latinoamericano de Ingeniería de Audio*, Montevideo, Uruguay. (Submitted).

VoxForge (2006). Voxforge. http://www.voxforge.org/pt. Accessed on 12.09.2021.

Wit.ai, I. (2022a). Build natural language experiences. https://wit.ai/. Accessed on 26.03.2022.

Wit.ai, I. (2022b). Python sdk for wit.ai. https://github.com/wit-ai/pywit. Accessed on 26.03.2022.

Zhang, A. (2014). Speech recognition (version 3.8) [software]. https://github.com/Uberi/speech_recognition#readme. Accessed on 28.03.2022.

Zhang, A. (2022). Speech recognition (version 3.8) [software]. https://github.com/Uberi/speech_recognition/tree/f149affeac095268c22443218bf7697d515fbe5a. Accessed on 28.03.2022.

A. Appendix

Table 10. Word Error Rate ascending average values and grouped by datasets

WER	Dataset	ASR Tool
0.27	Common Voice	Azure Speech Service (Speech Recognition)
0.39	Common Voice	Google Cloud Speech API (Speech Recognition)
0.45	Common Voice	Amazon Transcribe
0.50	Common Voice	Google Cloud Speech API
0.60	Common Voice	Vosk API (Speech Recognition)
0.93	Common Voice	Wit.ai
0.96	Common Voice	Wit.ai (Speech Recognition)
1.01	Common Voice	Vosk API
0.71	LaPS Mail	Vosk API (Speech Recognition)
0.73	LaPS Mail	Google Cloud Speech API (Speech Recognition)
0.76	LaPS Mail	Google Cloud Speech API
0.85	LaPS Mail	Wit.ai
0.93	LaPS Mail	Wit.ai (Speech Recognition)
1.02	LaPS Mail	Amazon Transcribe
1.08	LaPS Mail	Azure Speech Service (Speech Recognition)
3.24	LaPS Mail	Vosk API
0.15	Laps Benchmark	Vosk API (Speech Recognition)
0.19	Laps Benchmark	Google Cloud Speech API (Speech Recognition)
0.20	Laps Benchmark	Wit.ai
0.23	Laps Benchmark	Wit.ai (Speech Recognition)
0.24	Laps Benchmark	Amazon Transcribe
0.32	Laps Benchmark	Google Cloud Speech API
0.33	Laps Benchmark	Azure Speech Service (Speech Recognition)
1.01	Laps Benchmark	Vosk API
0.16	VoxForge	Azure Speech Service (Speech Recognition)
0.24	VoxForge	Wit.ai (Speech Recognition)
0.24	VoxForge	Wit.ai
0.41	VoxForge	Amazon Transcribe
0.47	VoxForge	Google Cloud Speech API (Speech Recognition)
0.57	VoxForge	Google Cloud Speech API
0.66	VoxForge	Vosk API (Speech Recognition)
1.05	VoxForge	Vosk API

Table 11. Match Error Rate ascending average values and grouped by datasets

MER	Dataset	ASR Tool
0.25	Common Voice	Azure Speech Service (Speech Recognition)
0.38	Common Voice	Google Cloud Speech API (Speech Recognition)
0.42	Common Voice	Amazon Transcribe
0.49	Common Voice	Google Cloud Speech API
0.56	Common Voice	Vosk API (Speech Recognition)
0.93	Common Voice	Wit.ai
0.96	Common Voice	Wit.ai (Speech Recognition)
1.00	Common Voice	Vosk API
0.57	LaPS Mail	Vosk API (Speech Recognition)
0.66	LaPS Mail	Google Cloud Speech API (Speech Recognition)
0.69	LaPS Mail	Google Cloud Speech API
0.76	LaPS Mail	Wit.ai (Speech Recognition)
0.76	LaPS Mail	Wit.ai
0.88	LaPS Mail	Amazon Transcribe
0.96	LaPS Mail	Azure Speech Service (Speech Recognition)
0.98	LaPS Mail	Vosk API
0.15	Laps Benchmark	Vosk API (Speech Recognition)
0.18	Laps Benchmark	Google Cloud Speech API (Speech Recognition)
0.20	Laps Benchmark	Wit.ai
0.22	Laps Benchmark	Wit.ai (Speech Recognition)
0.24	Laps Benchmark	Amazon Transcribe
0.31	Laps Benchmark	Google Cloud Speech API
0.32	Laps Benchmark	Azure Speech Service (Speech Recognition)
0.98	Laps Benchmark	Vosk API
0.15	VoxForge	Azure Speech Service (Speech Recognition)
0.23	VoxForge	Wit.ai (Speech Recognition)
0.23	VoxForge	Wit.ai
0.37	VoxForge	Amazon Transcribe
0.46	VoxForge	Google Cloud Speech API (Speech Recognition)
0.56	VoxForge	Google Cloud Speech API
0.63	VoxForge	Vosk API (Speech Recognition)
0.84	VoxForge	Vosk API

Table 12. Word Information Lost ascending average values and grouped by datasets

WIL	Dataset	ASR Tool
0.36	Common Voice	Azure Speech Service (Speech Recognition)
0.54	Common Voice	Google Cloud Speech API (Speech Recognition)
0.58	Common Voice	Amazon Transcribe
0.66	Common Voice	Google Cloud Speech API
0.71	Common Voice	Vosk API (Speech Recognition)
0.94	Common Voice	Wit.ai
0.97	Common Voice	Wit.ai (Speech Recognition)
1.00	Common Voice	Vosk API
0.59	LaPS Mail	Vosk API (Speech Recognition)
0.67	LaPS Mail	Google Cloud Speech API (Speech Recognition)
0.71	LaPS Mail	Google Cloud Speech API
0.79	LaPS Mail	Wit.ai
0.80	LaPS Mail	Wit.ai (Speech Recognition)
0.93	LaPS Mail	Amazon Transcribe
0.98	LaPS Mail	Azure Speech Service (Speech Recognition)
0.98	LaPS Mail	Vosk API
0.22	Laps Benchmark	Vosk API (Speech Recognition)
0.28	Laps Benchmark	Google Cloud Speech API (Speech Recognition)
0.34	Laps Benchmark	Wit.ai
0.36	Laps Benchmark	Wit.ai (Speech Recognition)
0.40	Laps Benchmark	Amazon Transcribe
0.48	Laps Benchmark	Google Cloud Speech API
0.51	Laps Benchmark	Azure Speech Service (Speech Recognition)
0.99	Laps Benchmark	Vosk API
0.20	VoxForge	Azure Speech Service (Speech Recognition)
0.34	VoxForge	Wit.ai (Speech Recognition)
0.34	VoxForge	Wit.ai
0.53	VoxForge	Amazon Transcribe
0.64	VoxForge	Google Cloud Speech API (Speech Recognition)
0.74	VoxForge	Google Cloud Speech API
0.77	VoxForge	Vosk API (Speech Recognition)
0.91	VoxForge	Vosk API