Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет

«Высшая школа экономики»

Факультет гуманитарных наук

Образовательная программа

«Фундаментальная и компьютерная лингвистика»

КУРСОВАЯ РАБОТА

На тему «Применение алгоритмов обработки речи к полевым данным: диаризация говорящих и распознавание речи»

Тема на английском: Applying speech processing to field data: speaker diarization and speech recognition

Студентка 2 курса

группы № 203

Неминова Екатерина Сергеевна

Научный руководитель

Сериков Олег Алексеевчи,

Старший перподаватель

Москва, 2022 год

Contents

Introduction 2

Theoretical background 2

Language Identification 3

Speaker diarization 3

Speech recognition 3

Diarization Error Rate (DER) 3

Character Error Rate (CER) 4

K-means clustering 4

Related works 4

Speaker diarization 4

ASR 5

Data 6

Data preprocessing 6

Hypothetical problems 7

Noises 7

Too much attention to the respondent 7

An influence of extralinguistic factors 7

Speaker Diarization Baseline 7

Preliminary tests 8

System description 10

System evaluation 11

Results 12

ASR Baseline 13

System description 13

System evaluation 13

Results 16

Further Investigation 17

References 18

The experiments codebase as well as the up-to-date term paper text can be found at: https://github.com/Ne-minus/termpaper_diar_asr

1. Introduction

Field linguistics is a vast part of linguistic science. Its aim is to describe various languages around the world and preserve all the information collected, thereby saving data on endangered languages. In addition, according to Serikov O. et al.[1], field linguistics is open-minded about languages, thus not paying attention to language prestige and collecting data from non-privileged languages. The work of a field linguist is extremely challenging and time-consuming, since after recording all the data, a scientist should also transcribe it for further use (creating grammars, writing articles, etc.)

Before first speech processing technologies were built, one couldn't even imagine the possibility of using different automatic tools for field speech recognition. Nowadays, when quite numerous of such devices have been invented, researchers are increasingly thinking about how to adapt them to field work in order to reduce the time spent on purely mechanical activity, such as manual transcription. In the present paper I rely on my personal experience of fieldwork and, subsequently, performing this kind of activity. So, I also became interested in this issue, and therefore the purpose of my term paper was to try to apply existing solutions for automatic speech processing to field recordings, analyze how they perform and, if possible, improve them.

2. Theoretical background

This section is dedicated to providing some general information.

Speech processing is a conversion of audio signal/speech into various kinds of digital information. Depending on the type of received digital information, there are different kinds of speech processing, such as speaker separation, language identification, speaker diarization and speech recognition[1]. The latter three are the most applicable for field matters.

[1] These are only the most common types.

2.1. Language Identification

Language identification is determining which language is spoken on the recording. I will not dive into this problem, as several groups of scientists have already dealt with it in the framework of a workshop dedicated to the identification of the spoken language. For more information, see [2].

2.2. Speaker diarization

> "Diarize" means making a note or keeping an event in a diary. Speaker diarization, like keeping a record of events in such a diary, addresses the question of "who spoke when" by logging speaker-specific salient events on multiparticipant (or multispeaker) audio data. [3]

In the same paper authors wrote that during the process of diarization, a recording is divided and clustered in groups. Each group gains a label of a particular speaker (audio segments from the same speaker are combined). Besides this, speaker diarization also helps to detect and further filter non-speech segments.

2.3. Speech recognition

Speech recognition is an automatic transcription of audio recordings. As an output one gets a text recording of what was said on the audio.

It is also possible to receive text notes of different formats. For example, it is much more convenient for phonetists to work with transcripts in the IPA[2] system, so there are some models that convert WAV-files to IPA text.

2.4. Diarization Error Rate (DER)

Anguera Miró X. noted: "The main metric that is used for speaker diarization experiments is the DER <...> It is measured as the fraction of time that is not attributed correctly to a speaker or to non-speech."[4].

He also suggested the formula:

$$DER = E_{spkr} + E_{MISS} + E_{FA} + E_{ovl}$$

E_{spkr}– speaker error (fraction of time that speaker's ID is assigned to another speaker)

[2] International Phonetic Alphabet

E_{MISS} – missed speech (fraction of time that speaker's label is assigned to non-speech segment)

E_{FA} – false alarm (fraction of time that speaker is labeled as a non-speech segment)

E_{ovl} – overlap speaker (fraction of time with several speakers in a segment)

2.5. Levenshtein distance (LD)

This is an algorithm that measures the discrepancy of two string utterances. According to this method, there are three types of mismatches between strings:

- deletion (D) ($cat \rightarrow \_at$)
- insertion (I) ($cat \rightarrow cat\underline{s}$)
- substitution (S) ($cat \rightarrow c\underline{u}t$)

For example, LD for strings *kitten* and *cutting* equals 4
(k → s, i → u, e → i, _ → g).

2.6. Character Error Rate (CER)

A metric based on LD.

The formula is:

$CER = \frac{S+D+I}{N}$, where N – the number of characters in source string

2.7. K-means clustering

This is an extremely popular clustering algorithm that is, as Kardi Teknomo defined, "an algorithm to classify or to group your objects based on attributes/features into K number of groups" [6], where K is a positive integer.

3. Related works

3.1. Speaker diarization

There are a lot of papers dedicated to speaker diarization. Before I started working on my own solution, I studied some of them. For example, an article called "The Third DIHARD Diarization Challenge" [6]. It describes the results of the diarization

challenge. The goal of the challenge was to improve the quality of speaker diarization and create more robust baselines. For assessment, they used recordings of different types of speech activity (diarization with reference annotation and diarization from scratch) in 11 variants of external conditions, including recordings with noises and interference. Some of these variants were audiobooks, meeting speech, conversational telephone speech, etc.

As a result, they stated that almost all the participants were able to improve the baseline to 4-5%, and the winner's solution performed 7% better. According to one of the boxplots shown in this paper, a lower level of processing quality was noted for diarization without a reference. Moreover, recording from different domains was diarized differently. The audiobooks had the lowest DER value, which was expected due to the good quality of the recording, while recordings from a restaurant had the highest.

The results of the competition described in a current paper showed that speech processing is a developing sphere. Over time, scientists are getting better and better at filtering out speech from everything else on recordings, which is especially important, for instance, in field linguistics.

3.2. ASR

Automatic speech recognition has great potential to simplify field work, although it is not used very often for this purpose. The authors of the work, called "ASR for documenting acutely under-resourced indigenous languages" [7], devoted it to the application of speech recognition models to the describing and preservation of low-resource languages. They argued that it is the small number of resources that prevents the widespread use of speech recognition for field recordings. Therefore, researchers decided to create 3 models for recognizing the endangered Seneca language found in North America. They used the spontaneous speech of 5 adult native speakers as data, and Kaldi as the base for their ASR system. Corpus and dictionaries were also utilized as sources for transcription and model training. For performance three evaluation metrics were used – OOV (the number of new words in a new sample), Accuracy (assess how close an output to ground truth) and WER (quite similar to CER but operates with words). The third model showed the best results: OOV = 31%, Accuracy

= 42% and WER = 65%. Also, over time, scientists expected a decrease in the OOV score.

4. Data

To test existing speaker diarization and ASR tools and make possible improvements, I used two types of data. The first one was recorded by me using the built-in microphone of my smartphone, and the materials of various expeditions were taken as the second type of records. At the first stages of the work, I used data from the Abaza [8] and Even [9] languages, and after creating the baselines, towards the end of the work, I got access to datasets containing the Evenki [10], Nenets[3], Chukot [11], Yakut [12] and Meadow Mari [13] languages. These datasets were used to assess how my baselines cope with their tasks.

Table 1. Distribution of the amount of material by language

language	the amount of material (min)
Evenki	439.414
Nenets	129.764
Chukot	74.427
Yuakut	109.169
Meadow Mari	139.10
Total	**891, 874**

4.1. Data preprocessing

Dataset for speech recognition consists of paths to files, time codes indicating when a certain string was pronounced and the transcription of this string. First of all, I unified the paths to the files (since they were in different formats) and converted all files to WAV-format with Sample Rate = 16000 Hz. Then I cut out the sections

[3] Data from the expedition transmitted by personal request

corresponding to the timecodes, and finally cleaned the transcriptions from traces of morphological annotation and other non-IPA symbols.

Dataset for speaker diarization also consists of paths to files, time codes indicating who spoke when and labels of the speakers. Firstly, I again unified the paths to the files and converted all files to WAV-format with Sample Rate = 16000 Hz. After that I made some dictionaries that join all speaker's segments to one value (where the speaker's labels are the keys of the dictionaries).

4.2. Hypothetical problems

This section is dedicated to potential issues that one may face while working with field data.

4.2.1. Noises

Taking into account that it is not always possible to conduct an interview in a quiet room or to use a good microphone, there may be a problem of poor recording quality due to extraneous noise or interference. This may affect the operation of algorithms and, consequently, degrade the quality of processing.

4.2.2. Too much attention to the respondent

Since field work is devoted to the description and preservation of language, the interviewer often pays more attention to the respondent (pushes the microphone towards him to achieve better recording quality), and therefore the interviewer himself is practically not heard on the audio. Such a difference in the volume of speakers can interfere with successful processing, since there is a possibility that different speech recognition tools are sensitive to volume.

4.2.3. An influence of extralinguistic factors

Occasionally extralinguistic factors, such as sex and age, may have an impact on the way a person speaks. That is why, the speech of respondents who differ in these parameters can be interpreted in divergent ways. For instance, children's or older people's speech may be recognized worse.

5. Speaker Diarization Baseline

To start with, there are not many solutions for diarization task in comparison with

ASR issue. In the present work I use the *Pyannote.audio*[4] model that obtains low DER scores and also is quite simple to utilize. That is why, I decided to apply it to field speech and evaluate its performance.

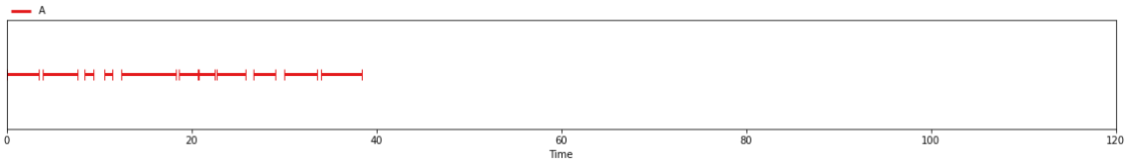
5.1. Preliminary tests[5]

As was proved by Ryant N. et al. [6], recording conditions and types of interaction influence the quality of diarization, so I became interested in how *Pyannote.audio* would cope with such sort of impact, which is unavoidable when processing field data. Hence, I conducted various experiments with different types of recordings, duration (D), volume (V) and sample rate (SplR).

Initially, I realised that field recordings were recognised worse than ones I recorded myself. Moreover, based on the results of my tests, I found out that duration affects the quality of diarization.

Figure 1. Dependence between duration (D) and diarization quality

a) D = 00:00:40



b) D = 00:05:00

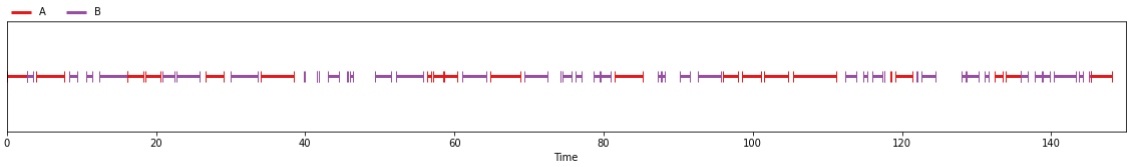


Figure 1 (where the labels of the speakers are indicated in capital letters and line segments indicate segments of the time during which a particular speaker talks) shows that duration influences the number of identified speakers. Although there are actually two speakers on the recording, as a result of the diarization of the short file, only one

[4] For documentation and relevant DER scores, see https://github.com/pyannote/pyannote-audio

[5] Since during the preliminary tests I did not have access to files annotated specifically for diarization, I based my conclusions in this section on comparing the graphs I received with what I heard when manually listening to the files.

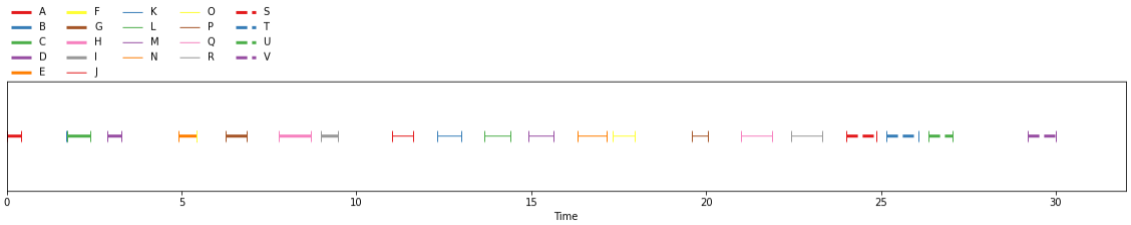
was identified. Therefore, since performance is worse when working with short files, I decided to concentrate on them and perhaps find a way to fix the incorrect identification of the speakers.

Concerning sample rate, Figure 2 shows results that I received by changing the values of this parameter for the same file. The file contained a conversation between an interviewer and a respondent, and the model of interaction was "question – answer".

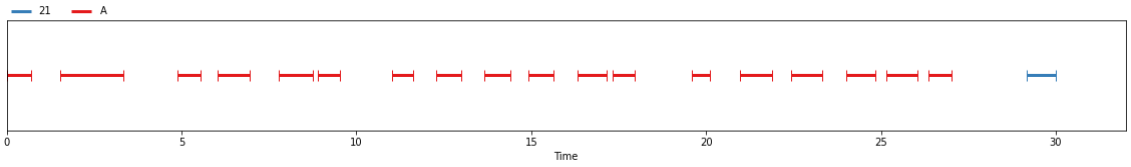
These plots (Figure 2.(a)-(b)) show how many speakers were found on the recording and when they talked. As it can be seen, the file with SplR = 16000Hz was diarized better since it identified only two speakers and not many (when there were only two speakers on recording). That is why, I decided to use SplR = 16000Hz for all further tests.

Figure 2. Dependence between Sample Rate (SplR) and diarization quality

a) SplR = 44100Hz, SplR = 48000Hz



b) SplR = 16000 Hz



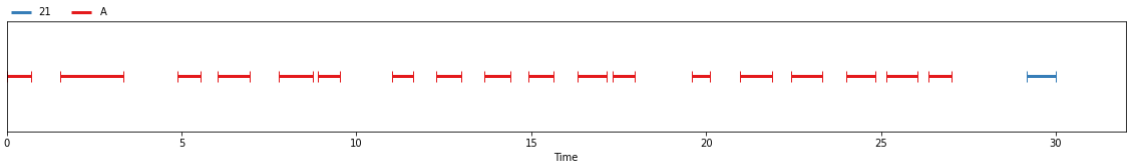
Although in this case the "number of speakers" problem was solved successfully, the "who speaks when" task was performed poorly, because, according to Figure 2(b), there is no dialogue between the interviewer and his interlocutor. To my mind, here I faced the issue described in section 4.2.2. So, I decided to normalize the volume of the recording (first manually via Audacity, then automatically) to make both speakers "equal" participants of the conversation. However, this did not lead to a strong improvement, and I got almost the same result.

The next variant of the solution to the problem discussed above was the K-means clustering method. Considering the purpose of speaker diarization and variation in the volume level of different speakers, it was reasonable to apply clustering by volume.

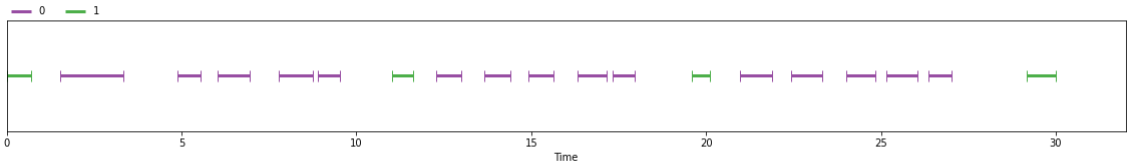
Figure 3 indicates the changes that occurred after applying the K-means clustering algorithm. Figure 3(b) is an exact representation of what was on the recording. That is why, K-means clustering seems to be a superior way to achieve proper diarization.

Figure 3. Diarization before (a) and after (b) utilizing K-means clustering

a)



b)



5.2. System description

My approach to building successful diarization contains the following steps:

- primary diarization, the result of which is a list of segments
- selecting a vector in each of these segments that sets the maximum volume of the segment (same speakers would have almost similar volume in different segments, so they would be incentives for K-means algorithm to merge them to one cluster)
- K-means clustering with subsequent assignment of new labels to segments

It took 1,5 hours to process the whole dataset consisted of 213 files (66 minutes of audio material).

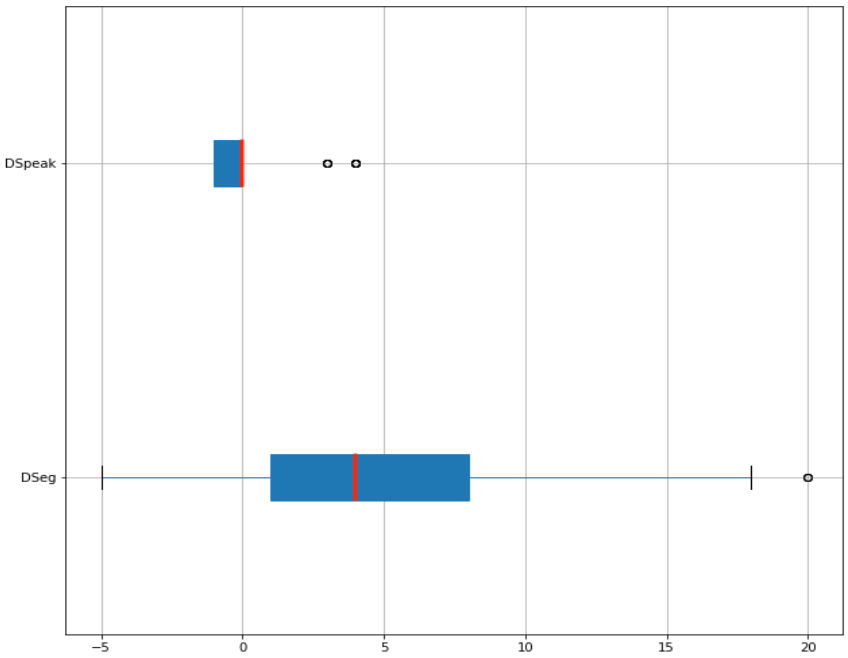
There are some drawbacks of this baseline. To start with, the clusterer is fed values related to the volume of the segment, respectively, if the segments have approximately the same volume levels, then the allocation of clusters becomes more difficult. Moreover, it is necessary to know how many speakers are on recording, because the K value should equal the number of speakers.

5.3. System evaluation

However, this solution proved to be excellent during spot check (Figure 3(b)), it also needed to be checked on a large dataset. Applying the baseline to a wide range of data required rather sophisticated data preprocessing, for instance, only annotated segments were extracted from audio files. This was made for the convenience of subsequent comparison of the diarization results with the original annotation, as well as to speed up the processing of the entire dataset.

I used fairly simple metrics to assess the work of the code: distance in the number of speakers (DSpeak)[6] and distance in the number of segments (DSeg)[7]. Figure 4 shows their distributions.

Figure 4. Distributions of DSpeak and DSeg



[6] DSpeak = Number on speakers annotated - Number of speakers recognised; (Perfect value is 0)
[7] DSeg = Number on segments annotated - Number of segments recognised; (Perfect value is 0)]

As one can see, the baseline copes very well with the problem of identifying speakers. The median is 0, the sample spread is small. However, there is a new problem. Unlike all the graphs presented above (where the selection of segments corresponds to the "sounding" intervals on the recording) it can be seen here that the number of recognized segments is very different from the number of segments in the annotation[8].

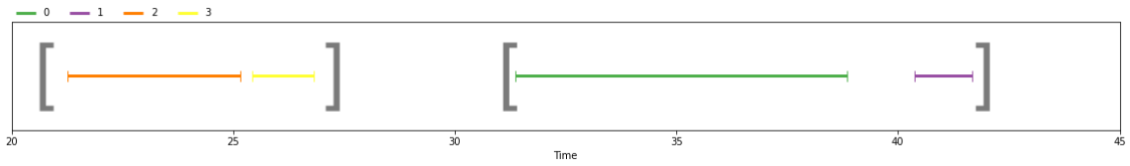
Figure 5 below shows an explanation to this phenomenon.

It can be noted that the ground truth plot has much more segments. Moreover, looking at the diarization result, it is obvious that some of the labels are assigned for the wrong speaker. For example, the last segment must have the same label as the first one, but it does not. The reason for this is that there are segments when speakers talk together. So, it is extremely difficult to understand who speaks when.

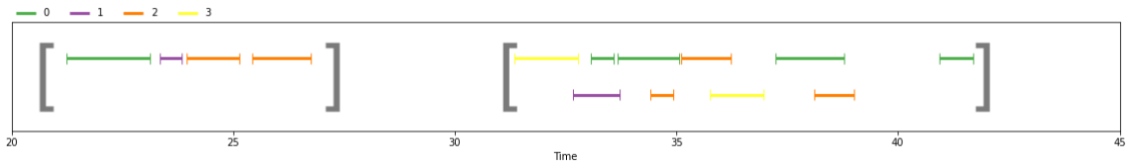
In spite of the fact that this almost ruins the performance of the baseline, one still can see that the clusters of several pieces[9] are located similarly.

Figure 5. Diarization results for a file (a) vs. ground truth (b)

a)



b)



5.4. Results

To start with, the task of determining the number of speakers was performed correctly in most cases. However, there is a problem with label assigning. Thus, to be

[8] Since the median equals 4, while the perfect value is 0, and whiskers are very long in both directions.

[9] Highlighted by gray brackets

considered a reliable method for improving speaker diarization tools, the usage of clustering algorithms should be modified and improved .

However, a new problem was encountered – the low quality of selection segments from audio files. Since many various values are common (from -5 to 20), it is extremely difficult to form any possible object patterns that affect the number of selected segments (selection seems to be quite random). Such a large spread of DSeg metric values was an absolutely unexpected situation, and the reason for this may be multiple speakers talking simultaneously. So, perhaps there is a need to abandon the method of diarization and a new approach for processing files of this kind[10].

6. ASR Baseline

There are a lot of speech recognition models. During my research I tried several models and libraries, such as PyAudio, Speech brain, etc. Despite the fact that they all have a low CER score, they are just **audio-to-text** conversion tools, which are common nowadays. I wanted to try a more specific and useful for linguistic purposes recognition method, which is **audio-to-IPA** conversion. That is why, I chose Wav2vec[11] model fine-tuned on multi-lingual Common Voice to transform field data to phonetic transcription.

This particular model can be applied only to WAV-files whose SplR = 16000 Hz. Furthermore, as input it should receive tensors that have been calculated by PyTorch.audio.

6.1. System description

Baseline building contains the following steps:

- using PyTorch.audio to create tensors
- applying Wav2vec model to these tensors

6.2. System evaluation

[10] see Further investigation section

[11] Full name is wav2vec2-xlsr-53-espeak-cv-ft

Firstly, I conducted an experiment with data in good quality, in the absence of strong background noise and other speakers. I recorded several recordings on my phone and applied the chosen model to them. I also made phonetic transcriptions (using Wiktionary and my own experience) to compare them with Wav2vec's output. To evaluate the performance, I used CER, and Table 2 shows the results I received.

Table 2. Speech recognition results for my audios

source	**recognised**	**cer**
krəkɐdiɬ	krakaɪdʒio	0.625000
bɪgɪmotsjeɬtorttɪperbɐlitɪvozʲivo	vigimosieltorttiperbali:tji:voʒyvot	0.424242
miʂkəmɐrtiʂkəkriʂkə	muʃkamartu:ʃkakriʃkɑ	0.578947
beɬajbɛrɛzɑpadmɐimɐknom	bilabiro:zapʌdmai:maknom	0.478261

At first glance, it seems that the model does not work well – in two out of four cases, the recognition quality is less than 50%, but after closer look it can be noted that both utterances are pretty similar. For example, [ʂ] is commonly recognised as [ʃ], [ɬ] as [l], etc. These units sound very much alike, which suggests that it is necessary to be "softer" about such substitutions when evaluating.

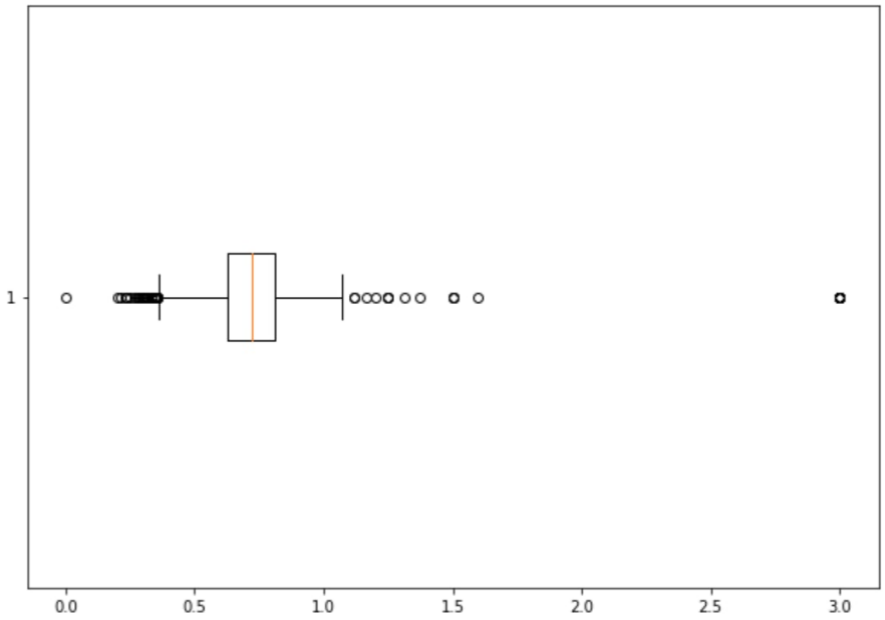
Secondly, I tested a couple of field files. Here is the automatic transcription result for one of them: *umaanmxmxamoaxanisa*. Since I did not have any transcription here and did not manage to do it myself, it is hard to say whether Wav2vec performs worse on field data or not. However, it is possible to catch by ear some clusters found in automatically created transcription.

Finally, I used the dataset described in section 3 to understand how well the baseline performs on field recordings. The task, which included the recognition of 3000 files[12], took two hours to execute, which is normal for this kind of processing. To assess the quality in this case, I also utilized the CER metric, and its distribution is demonstrated in Figure 6.

[12] which is approximately 151,739 minutes of material

According to the processing results, the median for the metric used is about 0.6, which is usually considered not a very good score. The size of a statistically significant sample is approximately 0.6, that is, the metric values normally range from 0.5 to 1.1.

Figure 6. Distribution of CER for field data



It is also interesting that there are two types of outliers on the graph:

- With CER >= 1.1
- With CER <= 0.5

Speaking about the first group, such high metric values probably occur because of the fact that the model recognized more characters than it was in the original transcription. To confirm or refute this hypothesis, I uploaded 3 outliers with the worst CER values. The results are presented in Table 3.Table3. 3 outliers with the worst CER score.

transcription	**recognised**	**cer**
-	nan	3.0
-	nan	3.0
-	nan	3.0

This sample does not confirm, but also does not refute the theory, it is simply uninformative. In the source and recognized data, these files were not annotated at all. “Nan” is just a way to note that there is nothing in this row of the dataset. However, it is recognized as a string completely different from the original transcription, and therefore when comparing two rows it gives such a high CER value. So, to make a reasonable conclusion, I discarded these cases and made a new sample shown in Table 4.

Table 4. 5 outliers with the worst CER score (without uninformative data)

transcription	**recognised**	**cer**
Tiiyudanyuudatikadmi	tʃimiːsentʃiːhiːjuːdeɲuːdʒdəka	1.25
guśo	uːsaː	1.25
Yedpi	eːaðəpiː	1.20
Nyanda	ɲamteei	1.17
gēbəjəgēhārgigūśo	geːbojogeːjxaːdiːkuːsa	1.12

This sample completely proves the hypothesis to be right. For all rows the number of recognized characters is bigger since “ː*”* is counted as a separate character.

As for the second group, it is unclear what affected their high recognition quality, so I filtered the sample and selected the first 5 outliers on the left. One could assume that it depends on phrase length, however, the data given in Table 5 prove the fallacy of this hypothesis.

Table 5. 5 outliers with the best CER score.

transcription	**recognised**	**cer**
da	da	0.0
ororinačir	ororinaːsir	0.20
aŋikākunkamendantkrimlabičōn	aŋikakonokamendantkrimlabitʃon	0.21
ďapkalīgirkum	japkalijirkum	0.23
potomhurumboγďapkalīn	potamurumbokjapkalin	0.24

6.3. Results

Considering that almost any code without additional improvements works worse on field recordings than on regular ones, I believe this high CER value to be quite normal. The reason for such CER-score may be both the problems specified in paragraph 5, and the problems of replacing very similar sounding units or different traditions of transcription. For example, in the dataset used, one can often notice sound [č] that is transcribed by the model as [t͡ʃ] according to the IPA standard, which also leads to an increase in error.

Particularly speaking about the latter two issues, some modified metric, which has a system of weights for various types of substitutions, could be used for a more accurate assessment.

Moreover, it was found that the main influence on the quality of transcription is not the length of the original phrase, but some technical and extralinguistic factors.

7. Further Investigation

As discussed above, the baselines do not perform perfectly, so further research is needed. To start with, it is possible to use other tools for diarization goals. For example, speaker separation models. These are tools designed to divide one audio track where several people are talking at the same time into two or more tracks. The speech of each participant of the conversation is recorded on his personal track. Thus, it would be possible to solve the problem of unsuccessful diarization in the presence of simultaneously speaking interlocutors.

Speaking of ASP, more complex indicators should be provided for evaluation. For example, the difference in transcription traditions should be taken into account. It is necessary to create a data set with character ratios showing that they are the same. Thus, when evaluating it, it will not be considered as a transcription error. Moreover, there are already customized metrics. For example, *abydos.phones* compares 2 strings, converting each unit in the string into a vector and calculating the difference between them. This module also has a system of weights, where a value can be specified by the user for each phonetic feature, so that the difference can be calculated in accordance with the objectives of the work.

8. Conclusion

After all, automatic speech processing has a large number of applications in linguistic science. However, the use of existing solutions does not always show the desired level of quality. In order to improve performance, it is necessary to use various additional methods. Moreover, it is necessary to use assessment tools competently, adjusting them to a specific task.

References

[1] – *Field matters. The first workshop on NLP applications to field linguistics* //field-matters.github.io – 2022

[2] – Salesky E. et al. *SIGTYP 2021 shared task: robust spoken language identification* //arXiv preprint arXiv:2106.03895. – 2021.

[3] – Park T. J. et al. *A review of speaker diarization: Recent advances with deep learning* //Computer Speech & Language. – 2022. – Т. 72. – С. 101317.

[4] – Anguera Miró X. *Robust speaker diarization for meetings*. – Universitat Politècnica de Catalunya, 2006.

[5] – Teknomo K. *K-means clustering tutorial* //Medicine. – 2006. – Т. 100. – №. 4. – С. 3

[6] – Ryant N. et al. *The third DIHARD diarization challenge* //arXiv preprint arXiv:2012.01477. – 2020.

[7] – Jimerson R., Prud'hommeaux E. *ASR for documenting acutely under-resourced indigenous languages* //Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). – 2018.

Data

[8] – Abaza language, fieldwork recordings

[9] – http://evenlang.ru

[10] – https://events.spbu.ru/eventsContent/files/corpling/corpora2013/Kazakevich.pdf

[11] – http://chuklang.ru

[12] – A. A. Petukhova, E. O. Sokur. 2021. *Yakut-Russian Corpus of Code-Switching*, Moscow: International Laboratory of Language Convergence, Higher School of Economics. (Available online at: http://lingconlab.ru/cs_yakut, accessed on 25.05.2022.)

[13] – Anna Volkova, Aigul Zakirova, Mikhail Voronov, Maria Dolgodvorova, Zinaida Klyucheva, Svetlana Kokoreva, Ilya Makarchuk, Irina Khomchenkova, Timofey Arkhangelskiy, Elena Sokur. Spoken corpus of Meadow Mari (as spoken in the village of Staryj Toryal, Novyj Torjal district, Mari El Republic, Russia). Moscow: Linguistic Convergence Laboratory, HSE University. (Available online at http://lingconlab.ru/spoken_meadow_mari/, accessed on 25.05.2022.)