Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 5929–5937
Marseille, 20-25 June 2022

Towards Speaker Verification for Crowdsourced Speech Collections

John Mendonça[1,2,*], **Rui Correia**[3], **Mariana Lourenço**[3], **João Freitas**[3], **Isabel Trancoso**[1,2]
[1]INESC-ID, Lisbon, Portugal
[2]Instituto Superior Técnico (IST), University of Lisbon, Portugal
[3]Defined.AI, Lisbon
{john.mendonca, isabel.trancoso}@tecnico.ulisboa.pt, {correia, mariana.lourenco, joao}@defined.ai

Abstract

Crowdsourcing the collection of speech provides a scalable setting to access a customisable demographic according to each dataset's needs. The correctness of speaker metadata is especially relevant for speaker-centred collections - ones that require the collection of a fixed amount of data per speaker. This paper identifies two different types of misalignment present in these collections: Multiple Accounts misalignment (different contributors map to the same speaker), and Multiple Speakers misalignment (multiple speakers map to the same contributor). Based on state-of-the-art approaches to Speaker Verification, this paper proposes an unsupervised method for measuring speaker metadata plausibility of a collection, i.e., evaluating the match (or lack thereof) between contributors and speakers. The solution presented is composed of an embedding extractor and a clustering module. Results indicate high precision in automatically classifying contributor alignment (> 0.94).

Keywords: Crowdsourcing, Speaker Verification, Datasets

1. Introduction

The success of Deep Neural Network-based solutions, which increasingly achieve (and surpass) human-level performance in several tasks, have allowed for Artificial Intelligence (AI) systems to incorporate our quotidian life in a multitude of areas, ranging from shopping, to banking, social media, or security, to name a few (Grace et al., 2018). The ubiquitous adoption of AI, paired with said systems' intrinsic characteristics, have put a high pressure on the sourcing of training data by Machine Learning practitioners (Halevy et al., 2009).

Crowdsourcing has been establishing itself as an alternative data sourcing paradigm, producing data with quality standards that are comparable to that of experts (Behrend et al., 2011). However, it does impose a new set of challenges. To get the most out of the "wisdom of the crowds" (Surowiecki, 2005) several methods to detect low quality work have been developed, including follow-up validation tasks, gold standard comparison (Snow et al., 2008), agreement between contributors (Aroyo and Welty, 2015), and behavioural capturing techniques (Rzeszotarski and Kittur, 2011).

But while these strategies address the quality of the data itself, they do not tackle the correctness of the contributors' self-reported metadata, typically provided during sign-up, and that, for a variety of reasons (including mistakes) may not correspond to the actual profile of the individual. Contributor metadata, as discussed above, is of importance to mitigate biases in the data, and can include dimensions such as age, gender, country of origin, languages spoken and corresponding proficiency levels.

For the particular case of speech data collections, which is the main subject of the present study, contributors' metadata is a critical component, as it is not only used to assure *fair* distributions over participants, but is part of the dataset labels itself. Certain aspects of the profile metadata, such as gender or language proficiency, can be validated with follow-up classification tasks, without what would be considered a prohibitive increase of the overall cost of the collection. For instance, one can pick a single recording per contributor and ask the remaining pool whether the voice matches the self-reported information. Furthermore, this process can be optimised by using high-performing ML solutions, like gender (Doukhan et al., 2018; Ghahremani et al., 2018) or language nativeness level classifiers (Abad et al., 2016; Botelheiro et al., 2021).

Another metadata dimension that is of particular interest for speech collection relates to speaker uniqueness, i.e., guaranteeing perfect contributor-speaker pairs. A common setting where this is imperative is when executing the so-called speaker-centred collections. These collections aim at recording a given amount of hours of speech per speaker, for a predefined number of speakers. Datasets with such characteristics are used, for instance, in the field of Speaker Recognition to train systems that are able to distinguish or pinpoint who is talking.

Misalignment between contributors and speakers can be classified into two distinct cases:

- *Multiple Speakers Misalignment* – contributors who shared accounts with other individuals;
- *Multiple Accounts Misalignment* – contributors who signed-up for the platform more than once.

With no quality control mechanism in place, these scenarios generate erroneous data: in the first case, it causes the same contributor identifier in a collection to

* J.M. completed most of this work at Defined.AI.

contain speech from more than one speaker and, in the second case, it causes different contributor identifiers to contain audio recorded by the same speaker.

In contrast with other profile dimensions, speaker uniqueness presents additional challenges for validation. On the one hand, relying on manual verification by other contributors would generate an unpractical (expensive) number of comparison combinations. If one imagines a collection targeting 100 speakers, each contributing with 15 utterances, validating *Multiple Speakers Misalignment* (MS) would require 10.5K pairwise comparisons between utterances[1], while validating *Multiple Accounts Misalignment* (MA) would add an extra 5K tasks approximately[2].

On the other hand, while automated solutions to speaker verification exist (Brummer et al., 2014; Snyder et al., 2018), the specific crowdsourcing setting poses yet another series of challenges that require further tailoring. In a mature crowdsourcing platform, a vast number of contributors (in the order of hundreds of thousands) can contribute to different collections, in different languages, at different proficiency levels, environment conditions, devices, and under particular instructions (which can even request for individuals to intentionally alter their natural speech production, for instance, to shout, whisper or to hyper-articulate words). These idiosyncrasies render the traditional two-stage process of enrolment followed by verification difficult to apply with success.

Under the circumstances described above, and assuming the relevance of speaker uniqueness information correctness, this paper tackles the issue by proposing an unsupervised speaker validation automated process, centred on the dataset that is to be verified (and no other data)[3].

The remainder of this paper will be organised as follows: Section 2 describes the current state-of-the-art of the Speaker Recognition area in general; Section 3 presents the architecture of the proposed solution, including experimental setup to evaluate its performance, testing datasets and results; Section 4 carries out an experiment where misalignment cases are simulated in a controlled manner in the data, and performance is reported in terms of precision and recall of the interest groups; Section 5 discusses results, elaborating on the suitability of the proposed solution to be used in a crowdsourcing setting; and finally, Section 6 presents the conclusions and points to future work directions.

[1]Assuming comparison between every combination of recordings of the same contributor, given by ${}^{15}C_2 \times 100 = 10,500$

[2]Assuming taking one recording per contributor, and comparing those for every contributor pair, given by ${}^{100}C_2 = 4,950$

[3]The Codebase for this paper can be found at www.github.com/johndmendonca/CrowdCluster

2. Background

In the context of the present paper, Speaker Verification (deciding whether a given utterance belongs to one of a closed set of known speakers) aligns the most with the problem statement described in the introductory section: make several binary decisions on the likelihood of a pair of utterances belonging to the same speaker (to validate *Multiple Speakers Misalignment*) or to different speakers (to validate the *Multiple Accounts* counterpart).

The representation of a speaker through embeddings is the current state-of-the-art in the Speaker Recognition area. Speaker embeddings provide a compact representation of speaker identity, as one single fixed-dimension vector. The training process of speaker embeddings, which occurs in large amounts of data, allows for inferring a relevant set of speaker characteristics (equal to the number of dimensions desired). This approach contrasts to the need of manually engineering a set of features (for instance related to prosody) that can distinguish between speakers.

For representing information in a unit-length hyperspace, speaker embeddings also come with the advantage of simplifying the implementation of the scoring module of the Speaker Verification system. Measuring the similarity between embeddings can be done with common algebraic operations, such as the cosine distance, or other dedicated operations between vectors, such as Gaussian-PLDA (Ioffe, 2006).

Two different embedding approaches are worth further analysis: x-vectors and ECAPA-TDNN embeddings. X-vectors, first presented by Snyder et al. (2018), were developed as an improvement of the i-vector system (Dehak et al., 2010), replacing Joint Factor Analysis of Gaussian Mixture Model supervectors by embeddings extracted from a feedforward DNN. The network of the x-vector system is divided into two different levels: the *frame level* uses a time delay architecture (TDNN) that functions on speech frames, offering temporal context (Peddinti et al., 2015); the *segment level* is connected to the frame level using a statistics pooling layer. This pooling layer calculates the mean and standard deviation from the aggregate output of the final frame level. This pooling procedure compiles information from the entire segment to subsequent layers. The training of the DNN is conducted using multi-class cross-entropy objective.

Concerning ECAPA-TDNN embeddings, introduced by Desplanques et al. (2020), while their architecture is largely based off the original x-vector, several enhancements are incorporated. Firstly, the initial frame layers are restructured into one-dimensional Res2Net modules with impactful skip connections. Similarly to SE-ResNet (Hu et al., 2018), Squeeze-and-Excitation blocks (SE) are introduced in these modules to explicitly model channel inter-dependencies. The SE block expands the temporal context of the frame layer by rescaling the channels according to global properties of

the recording. Secondly, information is aggregated and features are propagated to different hierarchical levels. Finally, the statistics pooling module is improved with channel-dependent frame attention. This enables the network to focus on different subsets of frames during each of the channel's statistics estimation. The network is trained by optimising the AAM-softmax (Deng et al., 2019) loss on the speaker identities in the training corpus. This loss directly optimises the cosine distance between the speaker embeddings.

3. Architecture of the Solution

As described in the introductory section, the purpose of the present work is to provide an unsupervised method for measuring speaker metadata plausibility of a speech collection, more specifically, to validate speaker uniqueness.

The formulation of the solution takes the approach found in Speaker Verification systems, in the sense that multiple binary decisions need to be made over a closed set of speakers (equivalent to the number of contributors in the dataset). Guaranteeing intra-speaker correctness (or the absence of *Multiple Speakers Misalignment*) involves a positive answer while matching all the possible pairwise combinations of every contributor's recordings. In contrast, guaranteeing inter-speaker correctness (or the absence of *Multiple Accounts Misalignment*) involves a negative answer while matching recordings of different contributors.

The differences between the proposed solution and a classical SV architecture have to do with the fact that neither enrolment nor the decision (and pre-defined threshold) components are present. Instead, a clustering approach is taken, similar to what is done in Speaker Diarization. In a sense, the goal is to allow for the data itself to define what is a consistent range of variability for a speaker, taking into consideration the dimensions that were varied in the collection.

In more detail, the proposed solution is composed by:

- **Embeddings Extractor** – generates a speaker embedding for every recording in the collection;
- **Clustering Module** – all embeddings are submitted for clustering, with the number of clusters equal to the number of contributors in the collection. With that information, the clustering algorithm is responsible for finding groups of recordings (clusters) that are the most similar amongst them.

In a full speaker correctness scenario, all recordings of a given contributor are allocated to one single cluster. Furthermore, that cluster has no other recordings other than those produced by the corresponding contributor.

Taking into consideration the literature on Speaker Recognition (Section 2), the process described above was tested under different settings by varying the embeddings extractor component (further detailed in Section 3.1) and the clustering module (Section 3.2). The resulting systems were evaluated on three distinct datasets, described in detail in Section 3.3. Results of these experiments are reported in Section 3.4.

3.1. Embeddings Extractor

The two embedding methods explored in this experiment are the ones described in Section 2. The x-vector embedding extraction followed the Kaldi Speech Recognition Toolkit (Povey et al., 2011) recipe for VoxCeleb (Nagrani et al., 2020). Training was conducted on the dev portion of VoxCeleb1, plus all of VoxCeleb2, augmented with reverberation and music, babble and noise from the MUSAN corpus (Snyder et al., 2015). The features were 30-dimensional MFCCs obtained every 10ms with a frame length of 25ms, mean-normalised over a sliding window of up to 3 seconds. An energy-based Voice Activity Detection module filtered out non-speech frames. X-vectors were extracted from the last layers of the pre-trained DNN model (before the softmax layer), outputting 512-dimensional embeddings.

Regarding the ECAPA-TDNN embedding extraction, the SpeechBrain toolkit was used (Ravanelli et al., 2021), which offers a model pre-trained on VoxCeleb1+2. Similar to the x-vector training, RIRs2 and MUSAN are used for data-augmentation purposes. The input features are 80-dimensional MFCCs from a 25 ms window with a 10 ms frame shift.

3.2. Clustering Module

The clustering algorithm chosen for the experiments was Agglomerative Hierarchical Clustering, a well-established method for performing unsupervised learning. In sum, this technique repeatedly aggregates the two closest nodes at every iteration, starting from a point where every data point (recording embedding) is an individual node, and concluding once reaching the desired number of clusters.

For this component, two parameters were varied: distance measure and linkage. With respect to distance measures, for x-vectors, cosine distance and PLDA Scoring were used; for ECAPA-TDNN embeddings, only cosine distance was considered.

Regarding linkage methods, results are reported according to complete-linkage (which at any given point chooses to merge the pair of clusters that will form the cluster with the smallest diameter) and average-linkage (which minimises the average distances between all pairs of objects in the resulting cluster).

3.3. Data

Experiments were carried out in three different datasets: DC–EN, DC–HE, and VC. Table 1 summarizes the datasets size in number of utterances and speakers, for convenience. We also present the detected misaligned after manual validation.

The first two datasets (with the DC– prefix) correspond to speech data collections executed on a real crowdsourcing platform. These collections consist of prompt

Dataset	# Utt	# Spk	MS %	MA %
DC–EN	2,745	277	0.0	0.0
DC–HE	2,144	147	3.4	0.0
VC	4,878	40	0.0	0.0

Table 1: Dataset summary. **MS** denotes Multiple Speaker Misalignment, **MA** denotes Multiple Account Misalignment.

reading tasks recorded in an application environment using a mobile phone, manually validated for speaker correctness. In these sets, wave files are downsampled to 16kHz, with a bit depth of 16, in a single channel. An energy-based VAD filters silence, leaving a leading silence of 300 ms and a trailing silence of 300 ms.

In more detail, the DC-EN dataset is a collection of American English in a silent environment. The prompts include wake-up calls of personal virtual assistants, followed by a request. A total of 277 adult contributors participated in the collection, resulting in a total of 2,745 executions (approximately 10 recordings per contributor on average). Average utterance duration for DC–EN is 4.98 seconds, with a minimum duration of 0.09s and a maximum of 10.71s.

The DC–HE dataset is a scripted speech data collection of Hebrew. The number of enrolled contributors for this job was 147 adults producing 2,157 executions (approximately 15 recordings per contributor on average). The average utterance duration is 8.19s, with a minimum duration of 4.12s and a maximum duration of 18.40s. Multiple Speaker misalignment was detected on 5 of the contributors, with the corresponding 13 utterances being excluded from the final dataset.

Finally, and for reproducibility reasons, the proposed solution was also evaluated on Voxceleb1 Test (VC). This subset of VoxCeleb1 contains 40 speakers, balanced with respect to gender, amounting to 4,874 utterances.

3.4. Results

This section presents the results in terms of V-measure (Rosenberg and Hirschberg, 2007) of the different system configurations discussed above, i.e., by varying embedding extractor (x-vector *vs* ECAPA-TDNN), distance metric (cosine *vs* PLDA), clustering linkage method (complete *vs* average), and dataset (DC-EN *vs* DC–HE *vs* VC). It is important to highlight that herein data has no misalignment, and all contributors in the datasets map to unique speakers. Therefore the goal is to understand the performance of the proposed solution in a perfect scenario.

Overall outcomes are summarised in Table 2. For x-vectors, PLDA consistently outperforms the cosine distance counterpart, as also observed by Snyder et al. (2018). In sync with the results reported by Desplanques et al. (2020) and Dawalatabad et al. (2021), ECAPA-TDNN embeddings surpass the x-vector approach for all experiments. With respect to Linkage, the V-measure for the best performer remains the same, with the exception of VC dataset, where complete-linkage achieves a relative improvement of 2%.

Dataset	Embedding	Distance	Linkage	
			Complete	Avg.
DC–EN	x-vector	Cosine	0.944	0.919
		PLDA	0.977	0.976
	ECAPA-TDNN	Cosine	**0.995**	**0.995**
DC–HE	x-vector	Cosine	0.971	0.969
		PLDA	0.996	0.998
	ECAPA-TDNN	Cosine	**1.000**	**1.000**
VC	x-vector	Cosine	0.835	0.725
		PLDA	0.976	0.950
	ECAPA-TDNN	Cosine	**0.998**	0.975

Table 2: Results for the multiple clustering settings.

In sum, the combination with ECAPA-TDNN embeddings, with cosine distance metric and complete linkage is chosen as the most suitable for assessing speaker correctness, consistently achieving V-measure > 0.995 across all datasets. The highest difference in performance was observed for the VC dataset, with a relative improvement of aproximately 16% between the x-vector and ECAPA-TDNN setting. This further corroborates the fact that ECAPA-TDNN's are a more robust speaker representation in face of variation in terms of contributors and recording conditions.

4. Speaker Misalignment

While collecting speech data in a crowdsourcing platform it is expected for some misalignment to be introduced in the data, either by contributors sharing accounts, or by holding several accounts and repeating participation under a different contributor identifier.

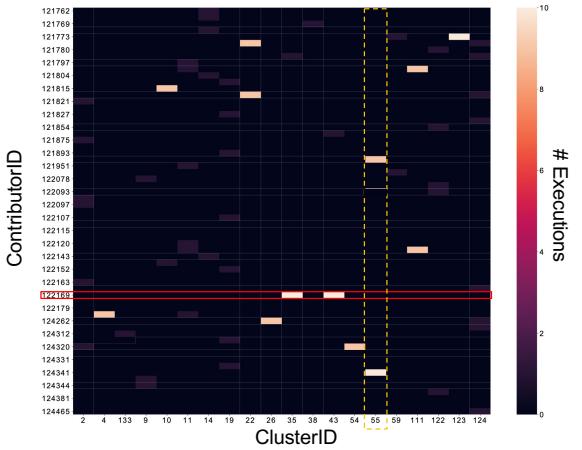


Figure 1: Example cluster assignment.

Figure 1 shows a heatmap with the results of the proposed method for a dataset that was not validated for speaker correctness. On the Y-axis, the figure displays different contributor identifiers (each associated

with one account), and on the X-axis it displays the cluster identifiers (which can be interpreted as identified speakers). The colour scheme at each contributor-cluster pair reflects the number of recordings that were assigned to that cluster (the lighter the colour, the more utterances, for a maximum of 10). For visualisation purposes, the figure was zoomed in on the portion of the data with conflicts. Observing the figure in detail, two interesting cases can be highlighted:

- Horizontal solid red box – contributor #122169 has a significant amount of recordings located in two distinct clusters, identified as clusters #35 and #43. This means that these two groups of utterances were considered distant enough to be separated. For the purposes of the problem statement, this can be interpreted as a sign of Multiple Speaker misalignment;
- Vertical dashed orange box – contributors #121893 and #124341 both have a significant amount of recordings located in the same cluster (#55). This means that the utterances recorded by these two contributors were considered similar enough to be clustered together, which can be a sign of Multiple Accounts misalignment.

Taking advantage of these observations of the clustering result, this section addresses the problem of classifying contributors according to three different classes: No Misalignment (NoM), Multiple Speakers misalignment (MS) and Multiple Accounts misalignment (MA).

4.1. Misalignment Detection

Having a clustering methodology defined, and given the particular configurations that can be observed after clustering, it is possible to define an algorithm capable of identifying and classifying misalignment in the resulting dataset of a speech collection.

Algorithm 1 details the sequence of steps proposed for the classification. Briefly put, given a dataset D (where each entry is a recording and the ID of the corresponding contributor), and the corresponding embeddings for each recording, E, the algorithm iteratively removes suspicious cases of both MS and MA, until no more can be detected.

In more detail, a contributor is signalled as having multiple account misalignment, C_{MA}, when all of their utterances are enclosed in a cluster that is shared with other contributor(s): this filtering is denoted as $get\_multiple\_accounts()$ in Algorithm 1. On the other hand, a contributor is identified as an instance of multiple speaker misalignment, C_{MS}, when their recordings are distributed over more than one cluster, and such clusters contain only occurrences of that contributor: this is denoted as $get\_multiple\_speakers()$ in Algorithm 1.

Upon removal of C_{MA} and C_{MS} from the data, D, the contributors without any misalignment are extracted, denoted C_{NoM}. The detection of these contributors is conducted by filtering $C \in D$ such that all of the executions pertaining the contributor are enclosed in a single cluster, with said cluster only having executions from a single contributor: in Algorithm 1 denoted as $get\_no\_misalignment()$. Contributors who end up not mapping to any of these three well-defined configurations are labelled as inconclusive (C_{inc}). Theoretically, these contributors may have misalignment of both types (MA and MS). In practice, the category C_{inc} is also expected to hold noisy results while clustering.

Central to the proposed process is the recurrent re-clustering of the dataset D between every operation of contributor classification (removal of misaligned contributors – C_{MA} and C_{MS} – from D). This strategy allows for the refinement of the clustering module, re-calibrating the number of clusters with the number of contributors as the dataset is processed.

Algorithm 1: Misalignment detection system

Input: D, E
Output: $pred$
$clst = cluster(D, E)$
$has\_MS, has\_MA = True$
$i = 0$
while has_MS *or* has_MA **do**
  $C_{MAi} = get\_multiple\_accounts(D, clst)$
  $has\_MA = (len(C_{MAi}) > 0)$
  if has_MA **then**
    $D = D - C_{MAi}$
    $clst = cluster(D, E)$
  end
  $C_{MSi} = get\_multiple\_speakers(D, clst)$
  $has\_MS = (len(C_{MAi}) > 0)$
  if has_MS **then**
    $D = D - C_{MSi}$
    $clst = cluster(D, E)$
  end
  $i+ = 1$
end
$C_{NoM} = get\_no\_misalignment(D, clst)$
$C_{MA} = set(C_{MA})$
$C_{MS} = set(C_{MS})$
$C_{inc} = D - C_{NoM}$
$pred = (C_{NoM}, C_{MA}, C_{MS}, C_{inc})$
return $pred$

4.2. Misalignment Generation

The final piece to understand the performance of the clustering strategy to validate speaker uniqueness correctness is to obtain correctly labelled misaligned datasets. As the process of manually validating a significant amount of data that would be sufficient to produce statistically sound results is unfeasible, this section addresses the task by introducing misalignment in face of validated datasets (such as the ones described in Section 3.3).

Given a validated dataset (with no misalignment be-

tween contributors and actual speakers), composed of N_C number of contributors, each with N_U utterances, misalignment is generated using the following heuristic:

- **Multiple Speakers** N_{MS} contributors are randomly selected from the dataset, forming a set of contributors denoted C_{MS}. For each contributor $C_i, \{i \in [0, .., N_{MS}/2 - 1]\}$ in C_{MS}, a random number of utterances are selected to be transferred to $C_j, \{j = i + N_{MS}/2\}$. The remaining utterances from C_i are removed from the dataset, together with the contributor identifier. As a result, C_j will contain utterances from two different speakers.
- **Multiple Accounts** N_{MA} contributors are randomly selected from the dataset, forming a set of contributors denoted C_{MA}. For each contributor $C_i, \{i \in [0, .., N_{MA}]$ in $C_{MA}\}$, a random number of utterances are selected to be transferred to a new contributor $C_j, \{j = i + N_{MA}\}$. As a result, C_i and C_j will contain utterances of the same speaker.

In the experiments carried out, C_{MA} is disjoint from C_{MS}[4], meaning that no contributor has more than one class of misalignment. Generating fraud with misalignment cases involving more than two contributors and in which each contributor can display both kinds of misalignment is a relevant topic for future work.
The heuristic defined above allows one to generate any desired scenario by defining the number of contributors (or percentage) for each type of misalignment, by tweaking the N_{MS} and N_{MA} parameters. For the purpose of the current paper, three types of scenarios are explored:

- No Misalignment – this scenario corresponds to running the process of contributor classification on the datasets without introducing any misalignment;
- Unbalanced Misalignment – scenarios that aim at testing the impact on performance when having an unmatched number of contributors with respect to actual speakers. If $C_{MA} > C_{MS}$, there will be a surplus of contributors in the dataset. The opposing setup, where $C_{MS} > C_{MA}$ will cause the data to be clustered according to less speakers than the ones that actually exists in the data.
- Balanced Misalignment: set of scenarios aiming at testing robustness of the system with increasing levels of misalignment.

4.3. Misalignment Detection Results

In this section, results when combining the misalignment generation process introduced in Section 4.2 with the detection system detailed in 4.1 are presented to gauge the performance of the proposed solution in the different proposed scenarios of misalignment. Due to the inherit randomness in the generation process, 100 runs were performed for each scenario. This process was executed for all datasets individually, however, given the similar conditions of DC–EN and DC–HE (and corresponding results), results for these are aggregated under **DC**. Results for VC are shown in separate for reproducibility reasons.
Precision and Recall and corresponding standard deviations for each class (**NoM**-No Misalignment; **MS**-Multiple Speaker; **MA**-Multiple Account) are presented in Table 3. The Confusion Matrices presented henceforth identify the average occurrence rate across all runs, and include the **Inc**-Inconclusive class, denoting the reject option.
In the absence of misalignment, the results are closely related to the ones presented in Section 3.4. More specifically, the reduction of Recall can be explained by the limited number of contributors with non-complete cluster assignments, which is exclusively a result of initial clustering errors, and not from the detection system in itself. As such, a small number of contributors are erroneously identified.
The unbalanced introduction of misalignment into the dataset (different amounts of MA and MS) leads to a change in the clustering assignments. As explained in Section 4.2, MA is simulated by splitting existing contributors into new ones, which in turn increases the actual number of contributors (and as a result the number of clusters). As such, clusters which would otherwise be pure are forced to split. Conversely, MS is simulated by combining 2 existing contributors into a single one, which reduces the actual number of contributors (and forcing the merger of portions of clusters). Consequently, the performance of the detector is affected.

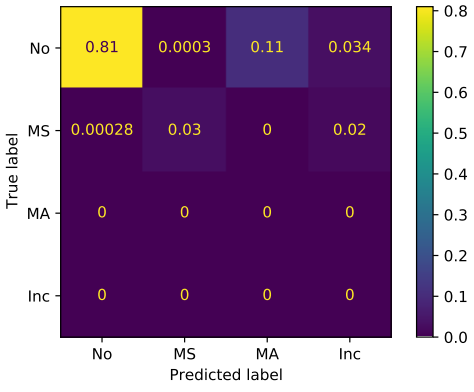


Figure 2: Confusion Matrix of **DC** with 5% MS.

When adding only MS to the dataset, Recall decreases significantly (.92 *vs*.86 NoM-Recall on DC; .93 *vs* .77 on VC), showing the misalignment detection method heavily relies on finding pure clusters in its heuristic. Furthermore, the Standard Deviation of the Recall

[4]In practice this means $N_{MS} + N_{MA} \leq N_C$.

Dataset	MS/MA (%)	NoM		MS		MA	
		Precision	Recall	Precision	Recall	Precision	Recall
DC	0/0	1.00 ± 0.00	0.92 ± 0.00	-	-	-	-
	0/5	1.00 ± 0.00	0.92 ± 0.01	-	-	0.74 ± 0.05	0.97 ± 0.06
	5/10	1.00 ± 0.00	0.90 ± 0.04	0.93 ± 0.05	0.75 ± 0.20	0.81 ± 0.09	0.98 ± 0.3
	5/0	1.00 ± 0.00	0.86 ± 0.05	0.98 ± 0.04	0.59 ± 0.21	-	-
	10/5	1.00 ± 0.00	0.79 ± 0.07	0.99 ± 0.07	0.52 ± 0.17	0.49 ± 0.11	0.99 ± 0.2
	5/5	1.00 ± 0.00	0.89 ± 0.07	0.94 ± 0.05	0.73 ± 0.19	0.65 ± 0.12	0.99 ± 0.03
	10/10	1.00 ± 0.00	0.82 ± 0.08	0.99 ± 0.02	0.61 ± 0.19	0.72 ± 0.11	0.99 ± 0.02
	25/25	0.99 ± 0.02	0.47 ± 0.09	0.99 ± 0.07	0.15 ± 0.07	0.91 ± 0.04	0.99 ± 0.01
VC	0/0	1.00 ± 0.00	0.93 ± 0.00	-	-	-	-
	0/5	1.00 ± 0.00	0.93 ± 0.02	-	-	1.00 ± 0.00	0.97 ± 0.12
	5/10	1.00 ± 0.01	0.82 ± 0.17	0.78 ± 0.40	0.57 ± 0.36	0.86 ± 0.22	0.99 ± 0.05
	5/0	1.00 ± 0.01	0.77 ± 0.12	0.62 ± 0.49	0.38 ± 0.34	-	-
	10/5	1.00 ± 0.01	0.66 ± 0.17	0.76 ± 0.43	0.31 ± 0.25	0.51 ± 0.24	0.98 ± 0.11
	5/5	1.00 ± 0.00	0.82 ± 0.16	0.78 ± 0.41	0,52 ± 0.34	0.77 ± 0.29	0.96 ± 0.14
	10/10	1.00 ± 0.01	0.68 ± 0.17	0.82 ± 0.38	0.36 ± 0.27	0.71 ± 0.23	0.98 ± 0.06
	25/25	0.94 ± 0.15	0.41 ± 0.18	0.57 ± 0.50	0.09 ± 0.09	0.92 ± 0.08	0.98 ± 0.04

Table 3: Misalignment Detection Results. **DC** denotes the combination of the crowdsourced datasets DC–EN and DC–HE.

shows larger deviations across different runs, indicating inconsistency of results, depending on the clusters affected. Figure 2 identifies some of the limitations of the system. Namely, the system erroneously identifies 10% of the contributor pool as being MA, when in reality they were not-misaligned and a combined 5% of contributors were identified as Inconclusive.

The introduction of only MA doesn't seem to affect the performance of the system as heavily as MS, with VC reaching near optimum Precision and Recall on the MA class, whereas DC reports a similar Recall but a Precision of 0.74.

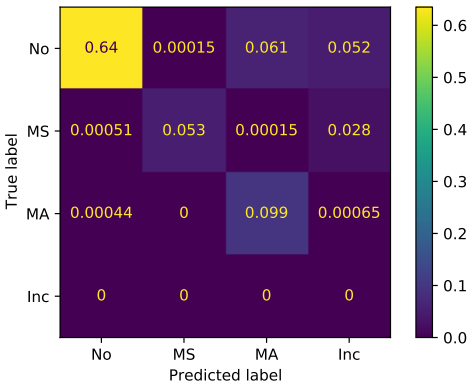
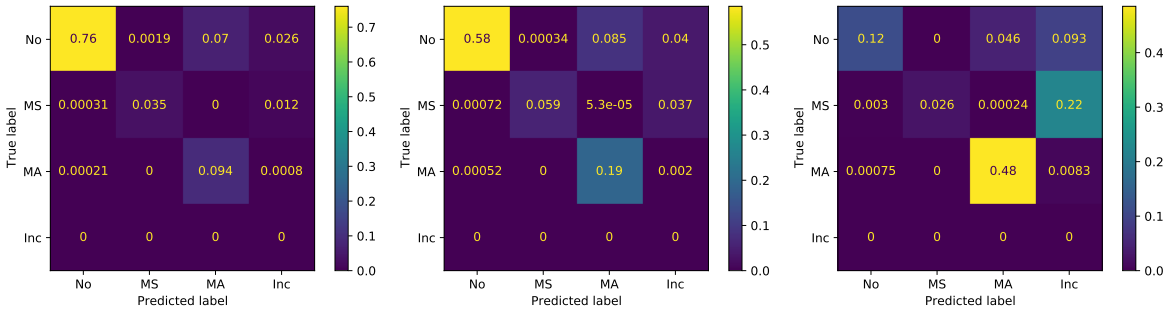


Figure 3: Confusion Matrix of **DC** with 10/5% MS/MA.

The experiments when combining both types of misalignment with different amounts are also in line with the results obtained individually. In both VC and DC, Recall on MS exhibits poor results when MS is larger than MA (.52 vs .75 MS-Recall on DC; .31 vs .57 on VC). Furthermore, a decrease in MA Precision was detected (.81 *vs* .49 MA-Precision on DC; .86 *vs* .51 on VC). If one compare these results with the ones obtained when inserting the same proportion of misalignment, the same hypothesis is also confirmed: When MA is larger than MS, MS precision remains about the same (.75 *vs* .73 on DC; .57 *vs* .52 on VC). The Confusion Matrix of C for the 10/5% scenario is presented in Figure 3.

When introducing the same amount of misalignment of both types, the number of clusters remains the same of the initial clustering with clean data. As such, the clustering performance does not affect the performance of the misalignment detector. A trade-off between Precision and Recall on MS and MA is detected when introducing both types at the same time. In DC, MS presents itself as having high Precision and low Recall, whereas MA presents high Recall, but lower Precision. The higher the amount of misalignment, the lower MS and NoM-Recall becomes. Precision on both MA and MS increases in DC, with the trade-off being a larger amount of Contributors being classed as Inconclusive. This is further evidenced looking at the Confusion Matrices in Figure 4.

It is important to note there are some differences in performance between DC and VC: VC exhibits a larger variability of results across runs (larger SD) and overall worse results for MS. This can be due to 2 different reasons: First and foremost, the dimensions of VC are quite disjoint from DC (double the amount of executions and less than 1/4 of the speakers). Secondly, unlike the crowdsourced datasets, which present themselves as having low variability (similar channel conditions, noise levels) the data mining process of Voxceleb includes a wide range of conditions, which leads to instability when forcing misalignment.



(a) Confusion Matrix 5% MS/MA. (b) Confusion Matrix 10% MS/MA. (c) Confusion Matrix 25% MS/MA.

Figure 4: Confusion Matrices of **DC** with increasing percentage of balanced MS/MA misalignment.

5. Discussion

The results achieved in the present study are promising with respect to the suitability of introducing automatic speaker verification into the crowdsourcing pipeline. While it cannot replace human validation altogether, the levels of performance achieved suggest that the validation process can not only be optimised (in terms of time and cost), but also improved (in terms of quality). A straightforward application of the methodology developed is a tool to guide subsequent speaker validation tasks. The high values of precision observed for cases with no misalignment state that such contributors do not need further validation, with up to precision $= 99\%$ for the crowdsourcing datasets. As seen in the previous section, it can make up to 76% (for 5% MS/MA) or 58% (10% MS/MA) of the contributors.

For the remaining cases, in face of lower precision results, the proposed systematic approach can be of help in identifying which utterances should be compared to corroborate or otherwise invalidate the system's categorisation. In fact, given that embeddings are extracted for all utterances, one can measure how distant every pair of utterances is from each other. To validate a contributor flagged with MS misalignment, it will be interesting to request other contributors to compare the two most distant utterances from that individual. On the other hand, for MA, one can compare the two closest recordings, i.e., the most confusable ones.

Taking as an example the volume of data in the DC–HE dataset (˜2K executions from 147 contributors), a full MS validation would require 13.4K pairwise comparisons, while validating MA would add an extra 10.7K tasks, for a total of approximately 24.1K tasks[5]. On the other hand, in a 5% MS/MA misalignment setting, the number of comparisons needed could amount to around 40[6]. Such a large difference stems from the fact the validation shifted from the need for comparing all executions individually, to picking the most promising examples for human comparison.

Altogether, the full automation on classifying no misaligned data and the significant reduction of human tasks required to ensure dataset consistency allows for faster cycles of data delivery in a crowdsourcing context and a considerable improvement in both efficiency and efficacy of the collection of speech data.

6. Conclusions

This work presents a speaker verification task in the context of misalignment detection for crowdsourced speech data collections. Noting the various combinations of different languages and conditions that occur during data collection, our proposed system leverages pre-trained embedding extractors to performs clustering of all submitted executions. Results show this method achieves near-optimal clustering without the need for cumbersome enrolment and threshold selection procedures. Additional experiments pertaining the automatic detection of misalignment were conducted. Namely, a generation heuristic is introduced and then clustering is performed to appoint contributors to their respective classes. Results of these experiments show high precision in the no misaligned class and a trade-off between precision and recall on the misaligned classes, which adding those deemed inconclusive, lays the groundwork for final human validation.

Future work directions on this work include experimentation with more data sources, privileging variability in terms of language and channel conditions. A simulation of detection of speaker misalignment while the crowdsourcing job is ongoing is worthwhile and, if successful, can prove useful in detecting misaligned contributors earlier. Additionally, a Machine Learning model that leverages a more extensive feature set that includes embedding distances and other cluster information could be implemented whilst maintaining a certain dimension of explainability.

7. Acknowledgements

This work has been supported by national funds through Fundação para a Ciência e a Tecnologia (FCT), under project UIDB/50021/2021 and grant PRT/BD/152198/2021.

[5] following the same approach used in the introductory section

[6] assuming a simplified scenario where one comparison is needed per case of MS, one comparison per case of MA, and two comparisons for each inconclusive contributor

Abad, A., Ribeiro, E., Kepler, F., Astudillo, R., and Trancoso, I. (2016). Exploiting phone log-likelihood ratio features for the detection of the native language of non-native english speakers. In *Interspeech 2016*, pages 2413–2417.

Aroyo, L. and Welty, C. (2015). Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Behrend, T. S., Sharek, D. J., Meade, A. W., and Wiebe, E. N. (2011). The viability of crowdsourcing for survey research. *Behavior research methods*, 43(3):800.

Botelheiro, D., Abad, A., Freitas, J., and Correia, R. (2021). Nativeness Assessment for Crowdsourced Speech Collections. In *Proc. IberSPEECH 2021*, pages 21–25.

Brummer, N., Mccree, A., Shum, S., Garcia-Romero, D., and Vaquero, C. (2014). Unsupervised domain adaptation for i-vector speaker recognition. In *Odyssey 2014*, pages 260–264.

Dawalatabad, N., Ravanelli, M., Grondin, F., Thienpondt, J., Desplanques, B., and Na, H. (2021). Ecapa-tdnn embeddings for speaker diarization. *arXiv preprint arXiv:2104.01466*.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699.

Desplanques, B., Thienpondt, J., and Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Proc. Interspeech 2020*, pages 3830–3834.

Doukhan, D., Carrive, J., Vallet, F., Larcher, A., and Meignier, S. (2018). An open-source speaker gender detection framework for monitoring gender equality. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5214–5218.

Ghahremani, P., Nidadavolu, P. S., Chen, N., Villalba, J., Povey, D., Khudanpur, S., and Dehak, N. (2018). End-to-end deep neural network age estimation. In *Proc. Interspeech 2018*, pages 277–281.

Grace, K., Salvatier, J., Dafoe, A., Zhang, B., and Evans, O. (2018). When will ai exceed human performance? evidence from ai experts. *Journal of Artificial Intelligence Research*, 62:729–754.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.

Ioffe, S. (2006). Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pages 531–542. Springer.

Nagrani, A., Chung, J. S., Xie, W., and Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December. IEEE Catalog No.: CFP11SRW-USB.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. (2021). Speechbrain: A general-purpose speech toolkit.

Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Rzeszotarski, J. M. and Kittur, A. (2011). Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *UIST*.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October. Association for Computational Linguistics.

Snyder, D., Chen, G., and Povey, D. (2015). Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE.

Surowiecki, J. (2005). *The wisdom of crowds*. Anchor.