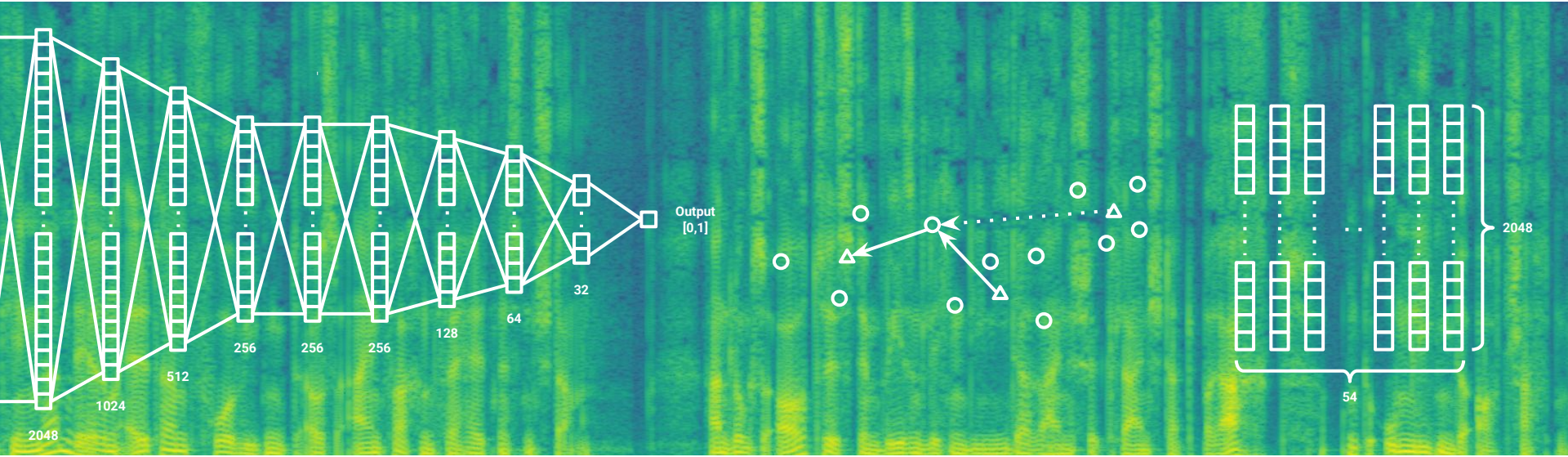
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Language Technology



January 5th, 2021 – Bachelor's Thesis Defence

Predicting Speaker Quality Using Embeddings

Author
Korbinian Koch

Supervisor
Dr. Timo Baumann

Co-Supervisor
Prof. Dr. Chris Biemann

In this presentation I will …

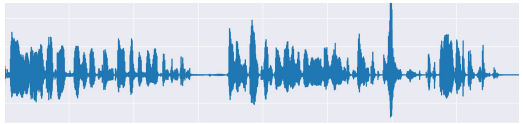
… explain the process and results of my thesis research

… show how you can be fooled by wrong assumptions

… summarize my learnings on the way

Table of Contents

Goal



“blah … blah … blah”

87% likeable

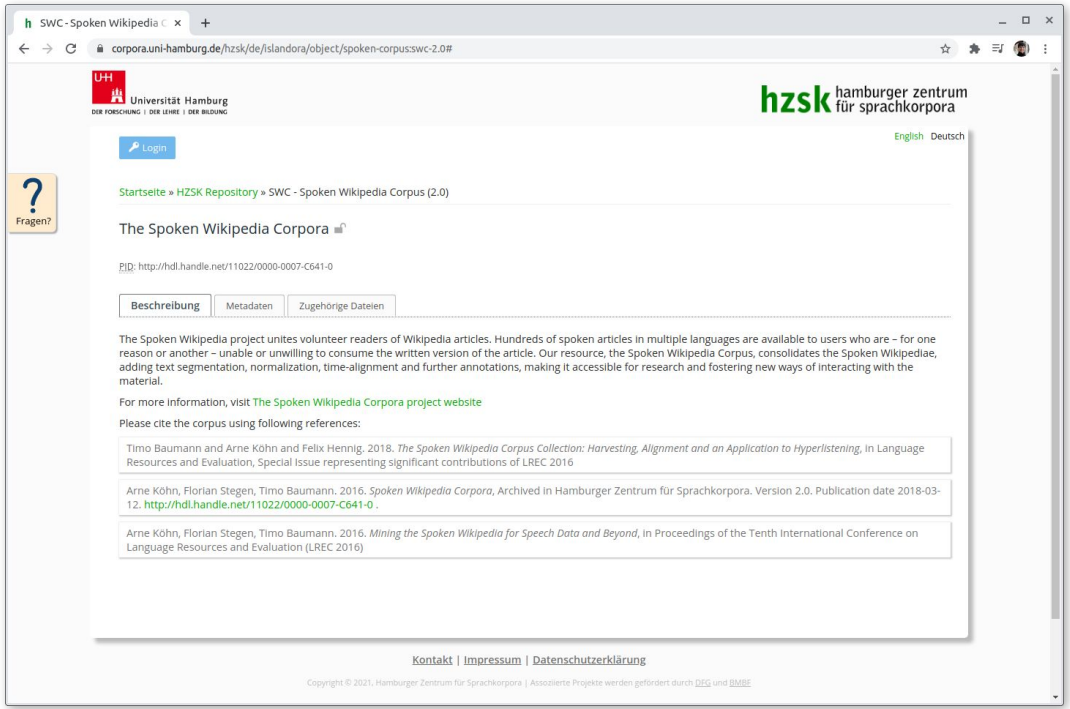
quality = likability

Research question

“Can pre-trained speech embeddings replace manual features for speech likability prediction without significant change of prediction accuracy and — more importantly — make them text-independent?”

To predict the likability of speakers we need ...

- Audio recordings of speakers
- Likability scores for these recordings
- A way to learn those scores



Spoken Wikipedia Corpus 2.0
(German part)

1014 articles

by

333 speakers

with a total duration of over

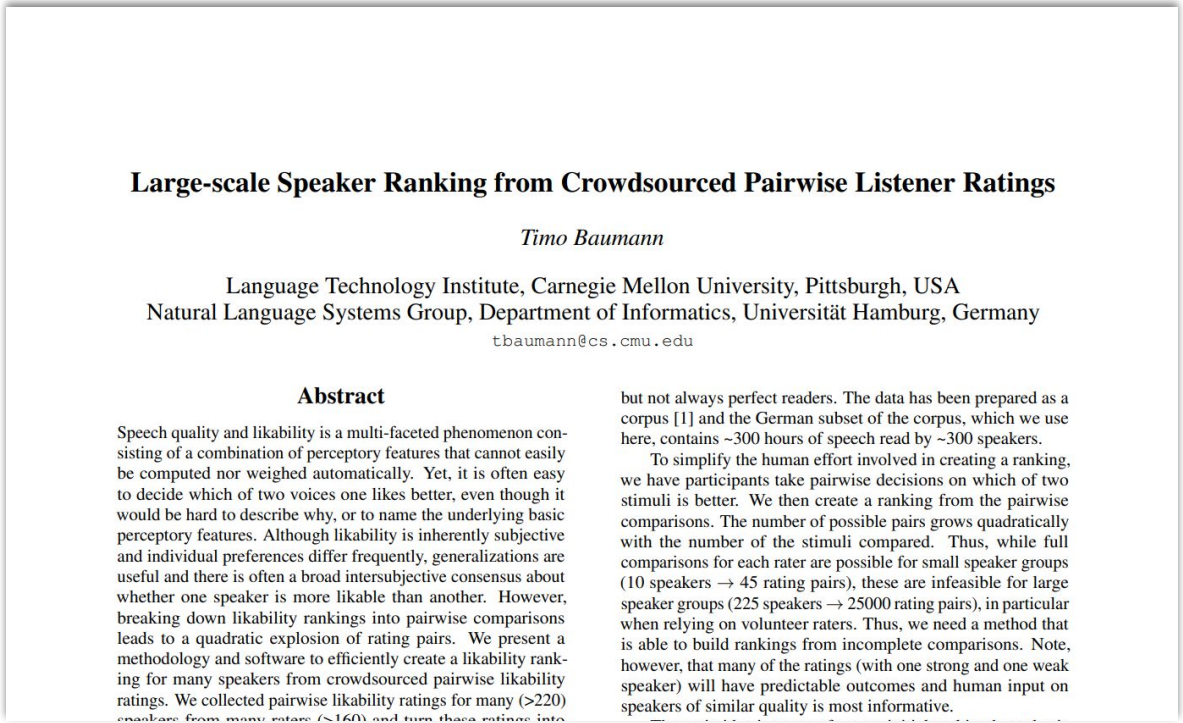
300 hours

Creating random 10 second samples sorted by speaker

```
korbinian@korbinian:~/Documents/Bachelorarbeit Timo/Speech-Quality-Estimation$ bash split.sh -p -m 10 -d 10
Target audio file length is 10 seconds.
Maximum amount of generated .wav files per article is 10.
Clipping warnings may occur.

Processing dir 1/1015 (german/3D/)
   ✅ Already generated 3d
Processing dir 2/1015 (german/3DO/)
   ✅ Already generated 3do
Processing dir 3/1015 (german/42_(Antwort)/)
...
   ✅ Already generated zwiebelturm
Processing dir 1014/1015 (german/Zwinger_(Dresden)/)
   ✅ Already generated zwingerdresden

❌ Found the following empty directories:
   german/Bissendorf/
   german/Osnabr%c3%bcck/

🗣️Found the following articles without readers:
   german/Erste_Marokkokrise/
   german/Hansken_(Elefant)/
   german/Hasenartige/
   german/Klaus_Kinski/
   german/Kutteln/
   german/Opossums/
   german/Plinius_der_J%c3%bcngere/
   german/Snookerweltmeisterschaft/
   german/Transkription_(Linguistik)/
   german/Ubuntu/
   german/Universalien_der_Musikwahrnehmung/
   german/Victor_Klemperer/
   german/Willi_Ostermann/

🦄 Done :)
```

To predict the likability of speakers we need ...

- Audio recordings of speakers ✓
- Likability scores for these recordings
- A way to learn those scores



Large-scale Speaker Ranking from Crowdsourced Pairwise Listener Ratings

Timo Baumann

Language Technology Institute, Carnegie Mellon University, Pittsburgh, USA
Natural Language Systems Group, Department of Informatics, Universität Hamburg, Germany
tbaumann@cs.cmu.edu

Abstract

Speech quality and likability is a multi-faceted phenomenon consisting of a combination of perceptory features that cannot easily be computed nor weighed automatically. Yet, it is often easy to decide which of two voices one likes better, even though it would be hard to describe why, or to name the underlying basic perceptory features. Although likability is inherently subjective and individual preferences differ frequently, generalizations are useful and there is often a broad intersubjective consensus about whether one speaker is more likable than another. However, breaking down likability rankings into pairwise comparisons leads to a quadratic explosion of rating pairs. We present a methodology and software to efficiently create a likability ranking for many speakers from crowdsourced pairwise likability ratings. We collected pairwise likability ratings for many (>220) speakers from many raters (>160) and turn these ratings into

but not always perfect readers. The data has been prepared as a corpus [1] and the German subset of the corpus, which we use here, contains ~300 hours of speech read by ~300 speakers.

To simplify the human effort involved in creating a ranking, we have participants take pairwise decisions on which of two stimuli is better. We then create a ranking from the pairwise comparisons. The number of possible pairs grows quadratically with the number of the stimuli compared. Thus, while full comparisons for each rater are possible for small speaker groups (10 speakers → 45 rating pairs), these are infeasible for large speaker groups (225 speakers → 25000 rating pairs), in particular when relying on volunteer raters. Thus, we need a method that is able to build rankings from incomplete comparisons. Note, however, that many of the ratings (with one strong and one weak speaker) will have predictable outcomes and human input on speakers of similar quality is most informative.

(Baumann, 2017)

Every article starts with ...

“Sie hören den Artikel […] aus Wikipedia, der freien Enzyklopädie.”

"Which one do you like more?"

"Sie hören den Artikel […] aus Wikipedia, der freien Enzyklopädie."

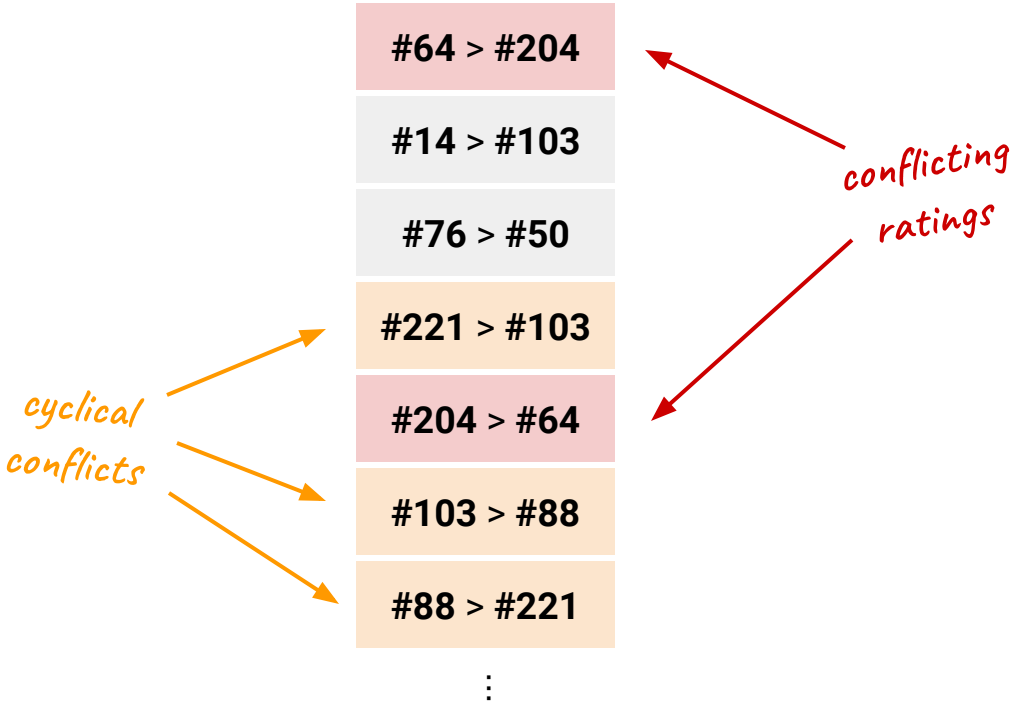
anonymized speaker **#64**

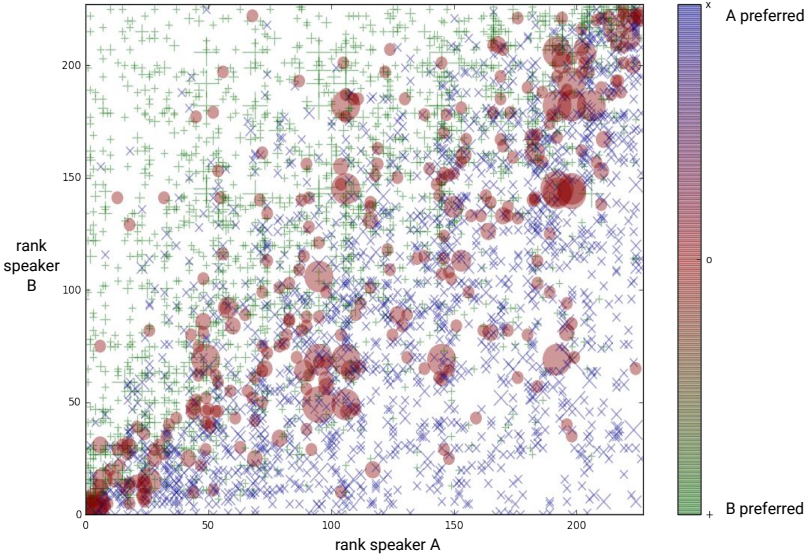
"Sie hören den Artikel […] aus Wikipedia, der freien Enzyklopädie."

anonymized speaker **#204**

"I like **#64** more than **#204**" ← rating

=> **4550 ratings** for **227 speakers**


#64 > #204
#14 > #103
#76 > #50
#221 > #103
#204 > #64
#103 > #88
#88 > #221
conflicting ratings
cyclical conflicts



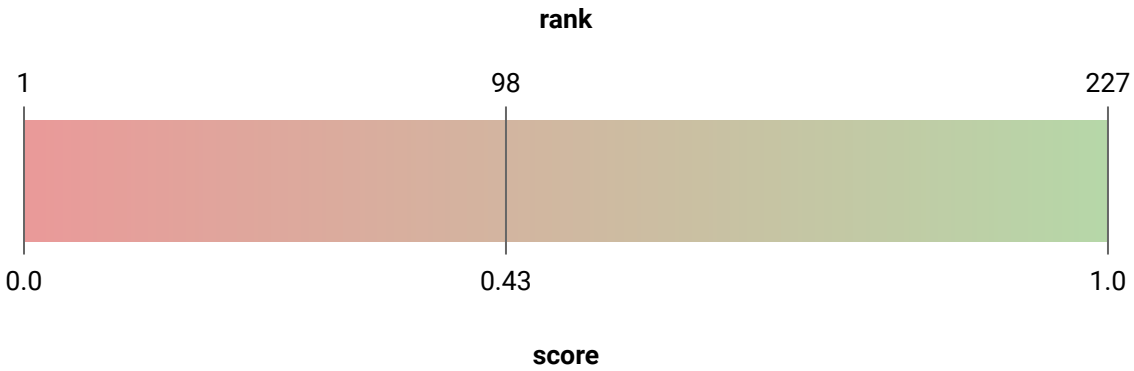
adapted from Baumann (2017)

- frequent disagreement indicates similar quality
- the TrueSkill Algorithm calculates median ranks for all speakers (x- and y-axis)
- speakers with similar quality are compared more often to determine a winner

Selected ranked speakers

	id	rank out of 227	speaker	article	anonymized	random
best speaker 😀	#111	227	ulrichbessler	Die Vorstadtkrokodile		
medium speaker 😐	#207	114	konsti	Deutsches Kaiserreich		
worst speaker 😣	#189	1	s1	Chruschtschowka		

From ranking to scores



To predict the likability of speakers we need ...

- Audio recordings of speakers ✓
- Likability scores for these recordings ✓
- A way to learn those ~~scores~~ *pairwise ratings*

What we already have:

#64 > #204

Learning to Determine Who is the Better Speaker

Timo Baumann

Language Technology Institute, Carnegie Mellon University, Pittsburgh, USA
tbaumann@cs.cmu.edu

Abstract

Speech can be more or less likable in various ways and comparing speakers by likability has important applications such as speaker selection or matching. Determining the likability of a speaker is a difficult task which can be simplified by breaking it down into pairwise preference decisions. Using a corpus of 5440 pairwise preference ratings collected previously through crowd-sourcing, we train classifiers to determine which of two speakers is "better". We find that modeling the speech feature sequences using LSTMs outperforms conventional methods that pre-aggregate feature averages by a large margin, indicating that the prosodic structure should be taken into account when determining speech quality. Our classifier reaches an accuracy of 97 % for coarse-grained decisions, where differences between speech quality in both stimuli is relatively large.

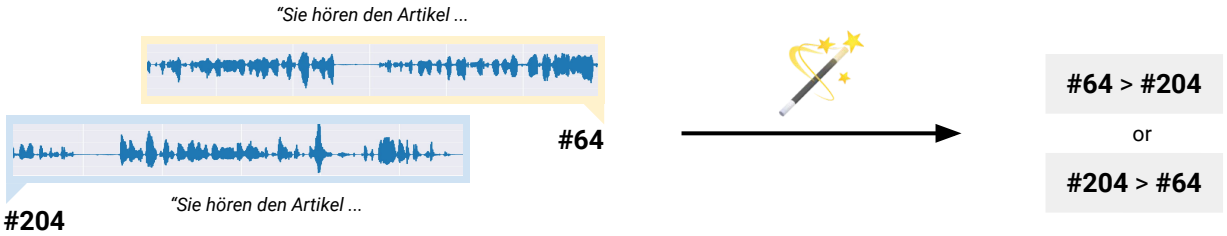
Index Terms: speech quality, likability ratings, sequence modeling

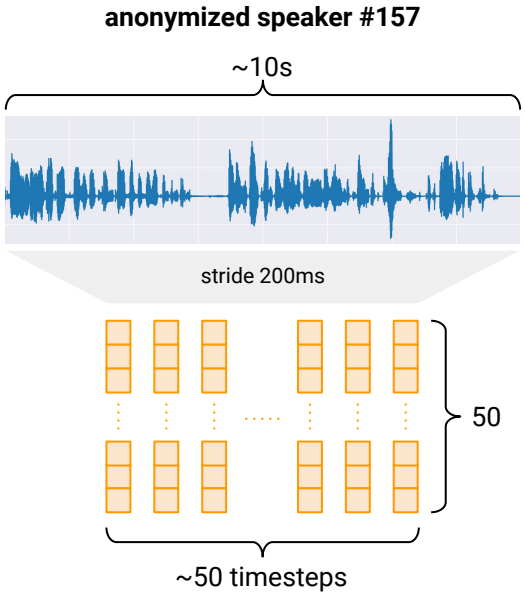
how to speak in a 'likable' way and improve one's ability to communicate more effectively.

In previous work [2], speaker likability has been modeled using OpenSmile [3] features based on linear and non-linear aggregation functions (such as means and medians) to aggregate over the duration of the stimulus. Features were used to train classifiers such as SVMs which resulted in moderately high (but better than chance) performance in classifying speakers as above or below median likability [2]. The abovementioned aggregation functions cannot take into account the context of feature characteristics in the stimulus, and are unlikely to accurately express more fine-grained details relevant for speech quality (such as where and how a pitch accent is realized, beyond mean pitch). In the present paper, we use neural sequence-learning methods (in particular: LSTMs [4]) to encode the complex speech quality into a latent feature space and use the difference in these features for pairs of speech stimuli to train our classifier. To the best of

(Baumann, 2018)

What we have:


"Sie hören den Artikel ...
#64
#204
"Sie hören den Artikel ...
#64 > #204
or
#204 > #64



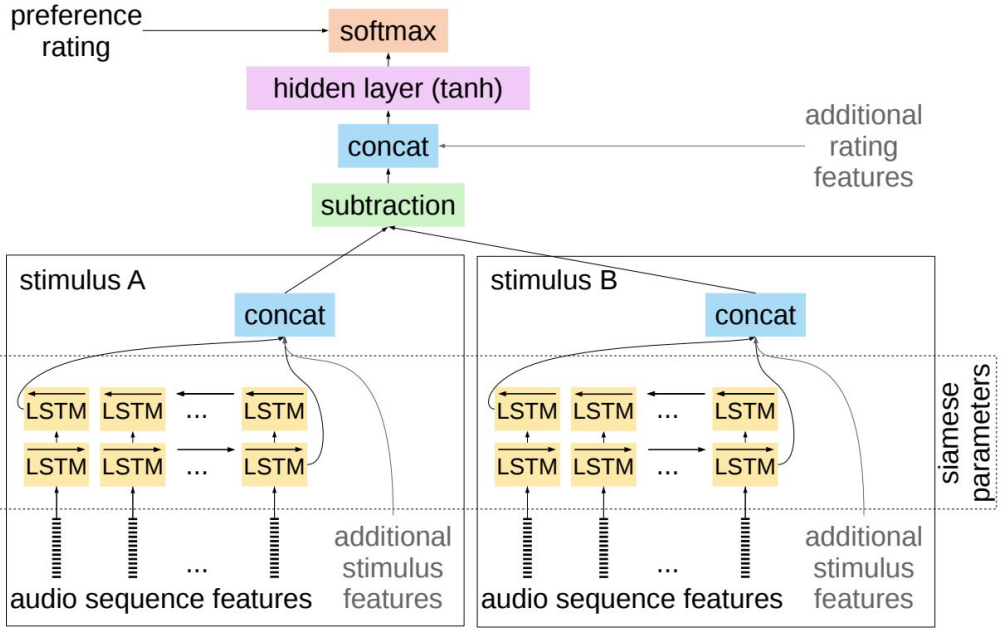
Mel-frequency cepstral coefficients (MFCC)

harmonics-to-noise ratio

jitter

shimmer

pitch



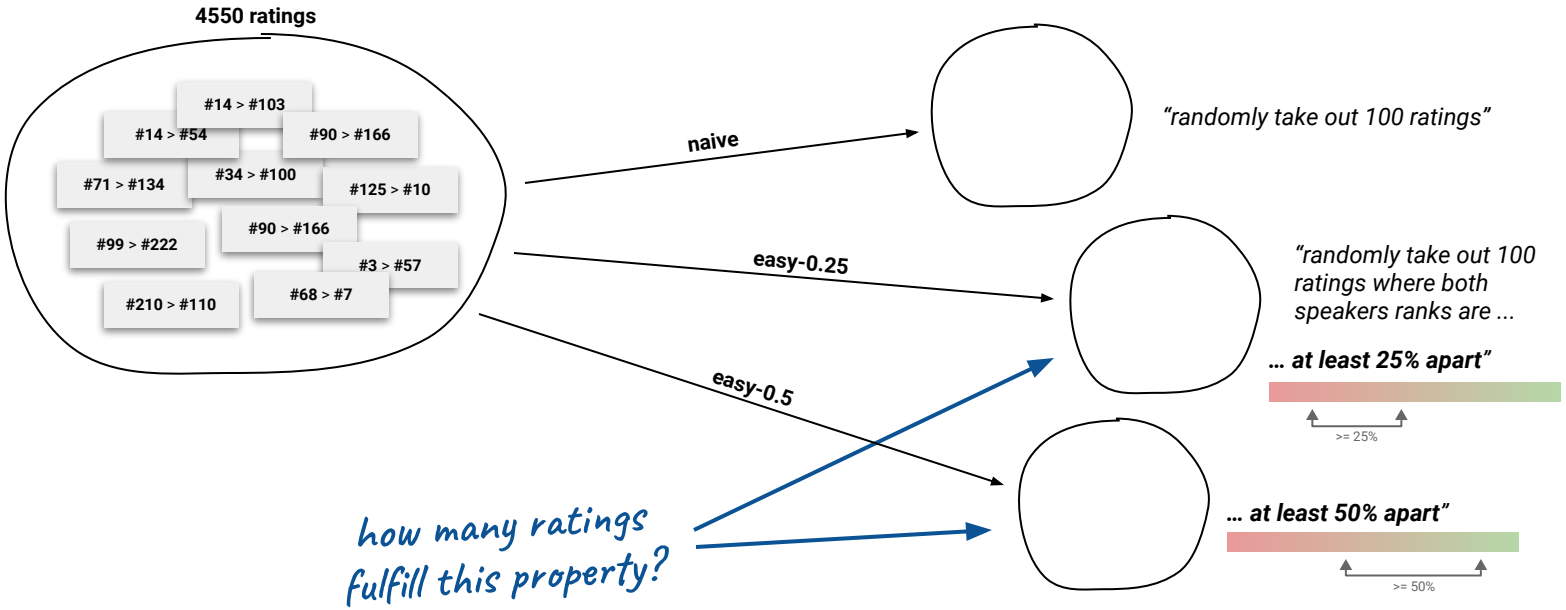
(Baumann, 2018)

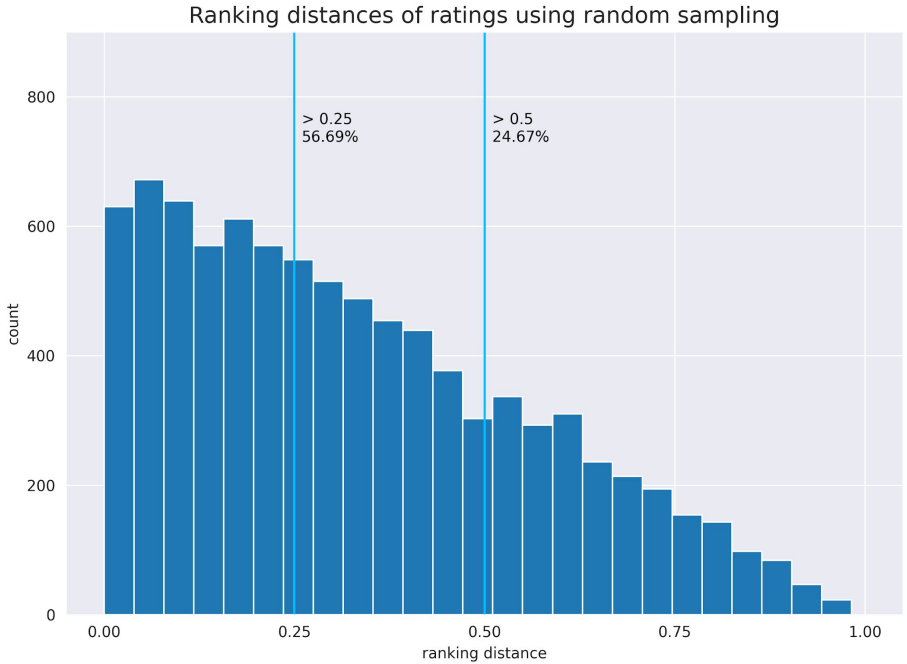
Table 2: *Accuracy (in percent) of full and reduced feature sets.*

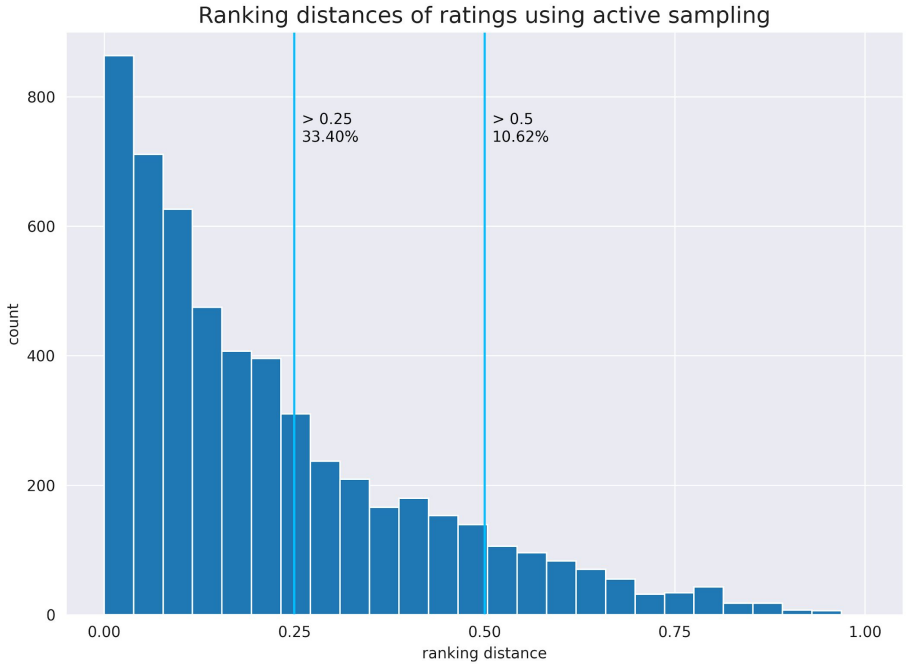
	setting	accuracy naïve	easy-0.25	easy-0.5
{d, ʒ, p, ŋ, …} + audio features	full	67.25	93	97
audio features	w/o phones	58.75	73	80

(Baumann, 2018)

		naive	easy-0.25	easy-0.50	extern	
audio features	siamese Bi-LSTM	58.75 %	73 %	80 %	*not reported*	as in paper
		64.65 %	86 %	89 %	60 %	reproduced

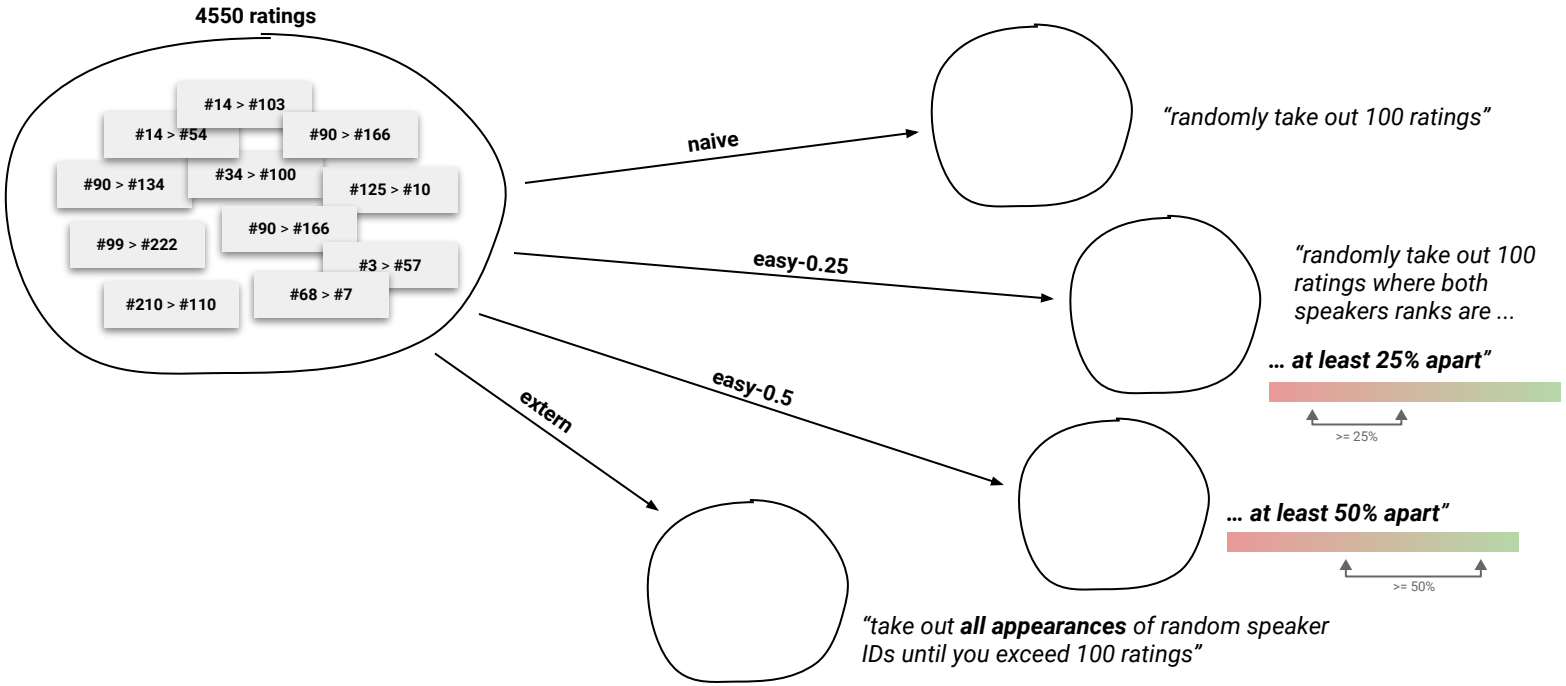

4550 ratings
#14 > #103
#14 > #54
#90 > #166
#71 > #134
#34 > #100
#125 > #10
#90 > #166
#99 > #222
#3 > #57
#210 > #110
#68 > #7
naive
easy-0.25
easy-0.5
"randomly take out 100 ratings"
"randomly take out 100 ratings where both speakers ranks are ...
... at least 25% apart"
>= 25%
... at least 50% apart"
>= 50%
how many ratings fulfill this property?

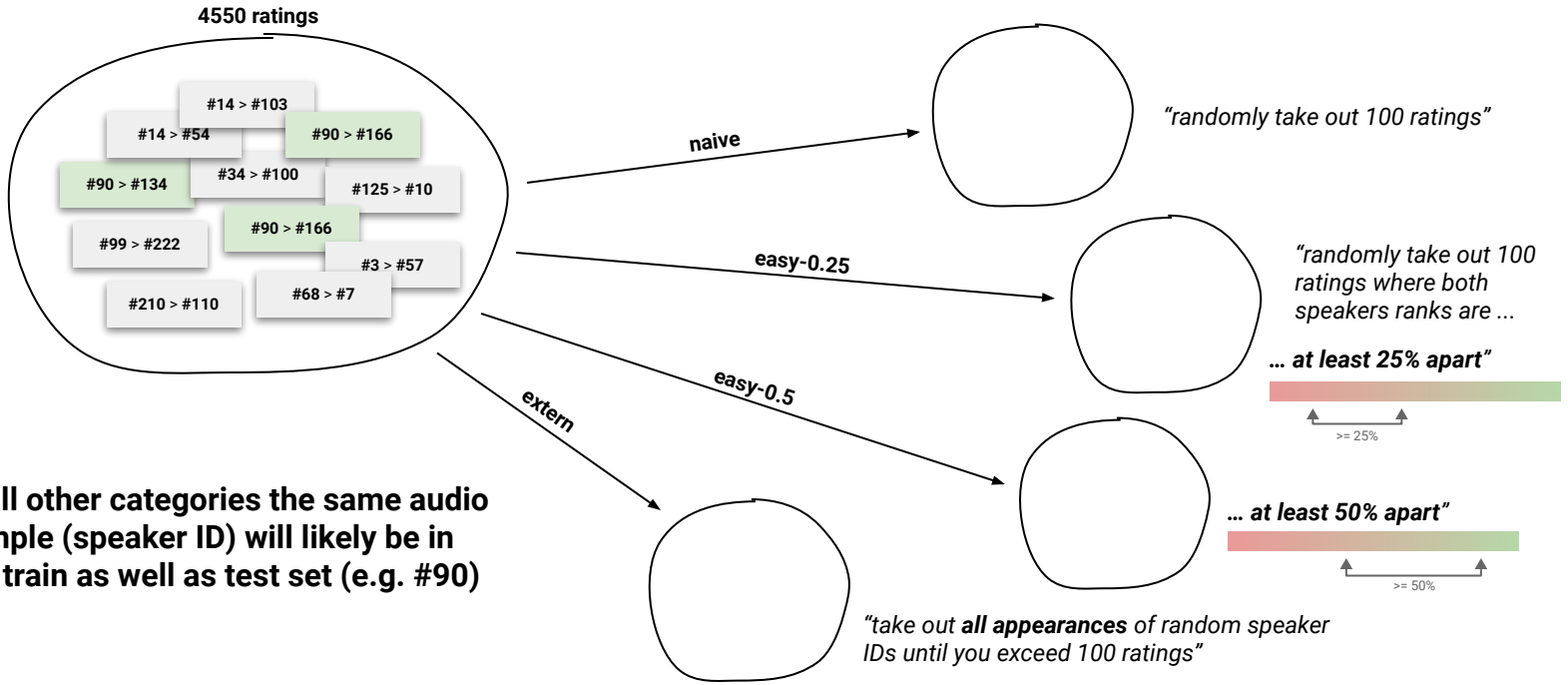

Ranking distances of ratings using random sampling
> 0.25
56.69%
> 0.5
24.67%
800
600
400
200
0
count
0.00
0.25
0.50
0.75
1.00
ranking distance


Ranking distances of ratings using active sampling
> 0.25
33.40%
> 0.5
10.62%
count
800
600
400
200
0
0.00
0.25
0.50
0.75
1.00
ranking distance

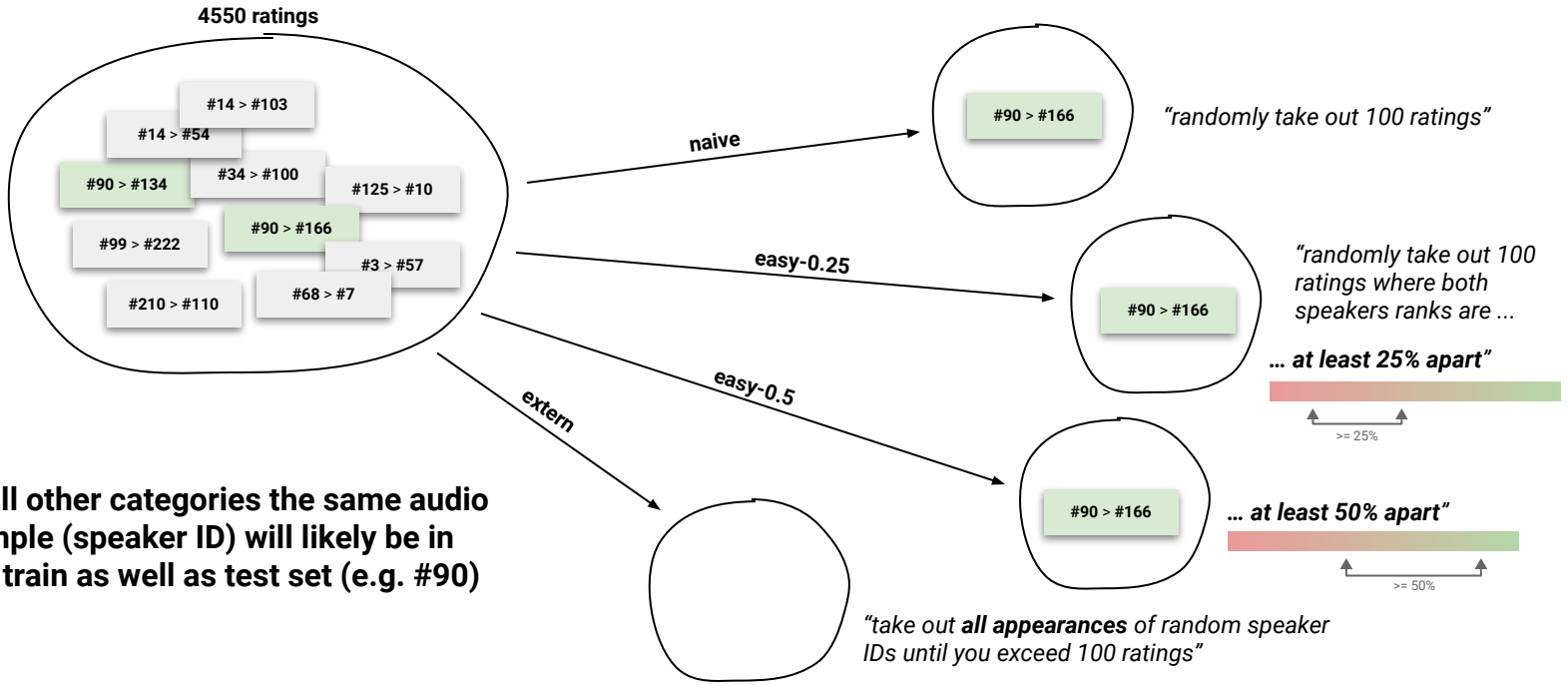
		naive	easy-0.25	easy-0.50	extern	
audio features	siamese Bi-LSTM	58.75 %	73 %	80 %	*not reported*	as in paper
		64.65 %	86 %	89 %	60 %	reproduced

```
test easy25 correct: 76.00, proportion: 0.8600
test easy50 correct: 87.67, proportion: 0.8867
test extern correct: 87.67, proportion: 0.6017
test naiveA correct: 64.00, proportion: 0.6600
test naiveB correct: 63.33, proportion: 0.6333
```

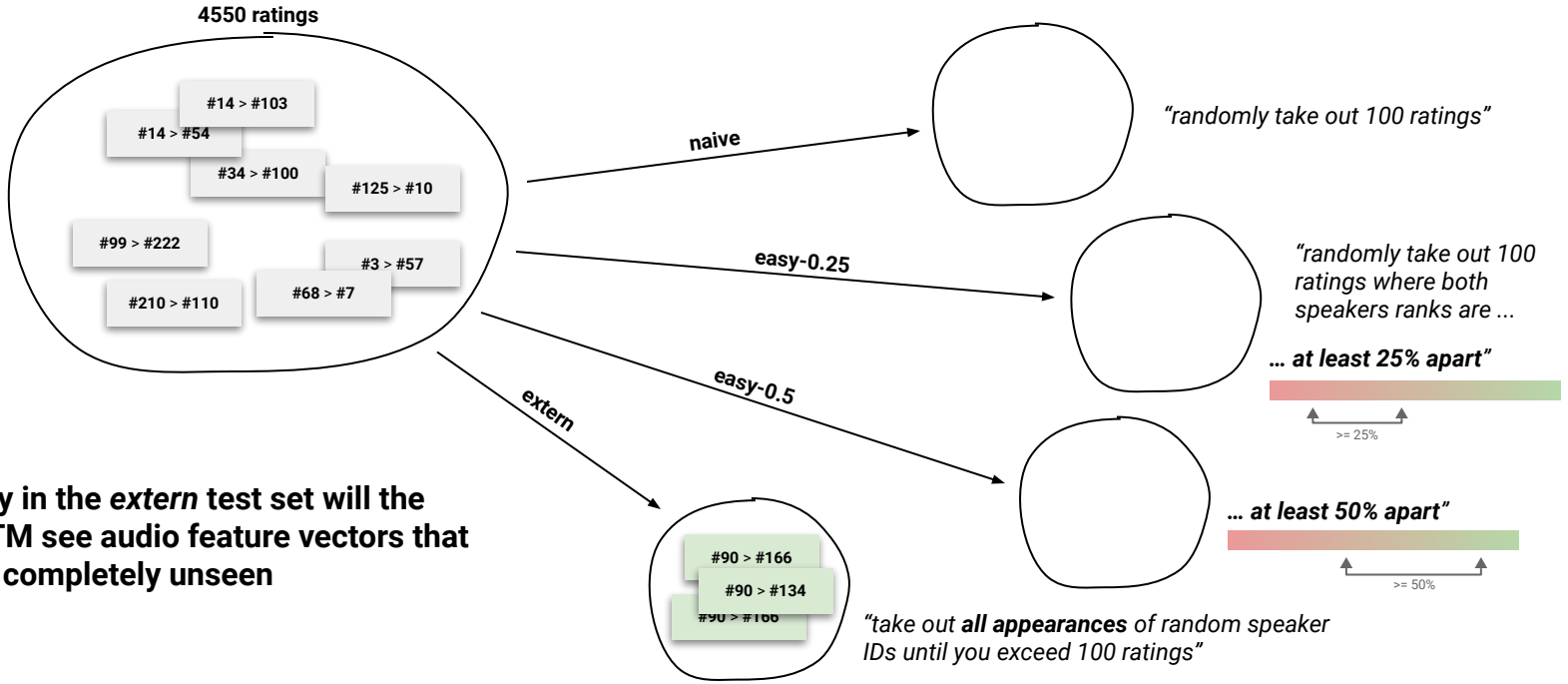

4550 ratings
#14 > #103
#14 > #54
#90 > #166
#90 > #134
#34 > #100
#125 > #10
#90 > #166
#99 > #222
#3 > #57
#210 > #110
#68 > #7
naive
"randomly take out 100 ratings"
easy-0.25
"randomly take out 100 ratings where both speakers ranks are ...
... at least 25% apart"
>= 25%
easy-0.5
... at least 50% apart"
>= 50%
extern
"take out all appearances of random speaker IDs until you exceed 100 ratings"



In all other categories the same audio sample (speaker ID) will likely be in the train as well as test set (e.g. #90)



In all other categories the same audio sample (speaker ID) will likely be in the train as well as test set (e.g. #90)



Only in the *extern* test set will the LSTM see audio feature vectors that are completely unseen

Let's create a simple baseline model

- with a few test sets of size ~100 we still have ~4000 ratings left to train on
- that's enough to calculate (relatively) accurate ranks using TrueSkill
- for ratings from ***naive***, ***easy-0.25*** and ***easy-0.5***, always let the higher rank win
- for the unseen speakers in ***extern***, simply assume they rank mediocre (rank 112/224 or score 0.5)

=> **rank-based classifier**

does not listen to the audio at all
only knows rank

		naive	easy-0.25	easy-0.50	extern	
audio features	siamese Bi-LSTM	58.75 %	73 %	80 %	*not reported*	as in paper
		64.65 %	**86 %**	89 %	60 %	reproduced
rank-based	baseline	**72 %**	82 %	**95 %**	**60.7 %**	

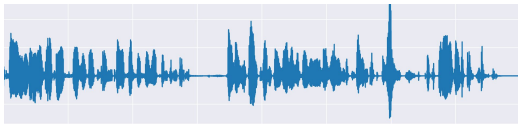
=> Can we really learn something from audio features alone?

Back to this approach ...

"How was your day? I ...

"Yesterday I was listening to ...

"Sie hören den Artikel ...



"blah … blah … blah"

87% likeable

we want predictions **for any spoken text**

(not only *"Sie hören den Artikel …"*)

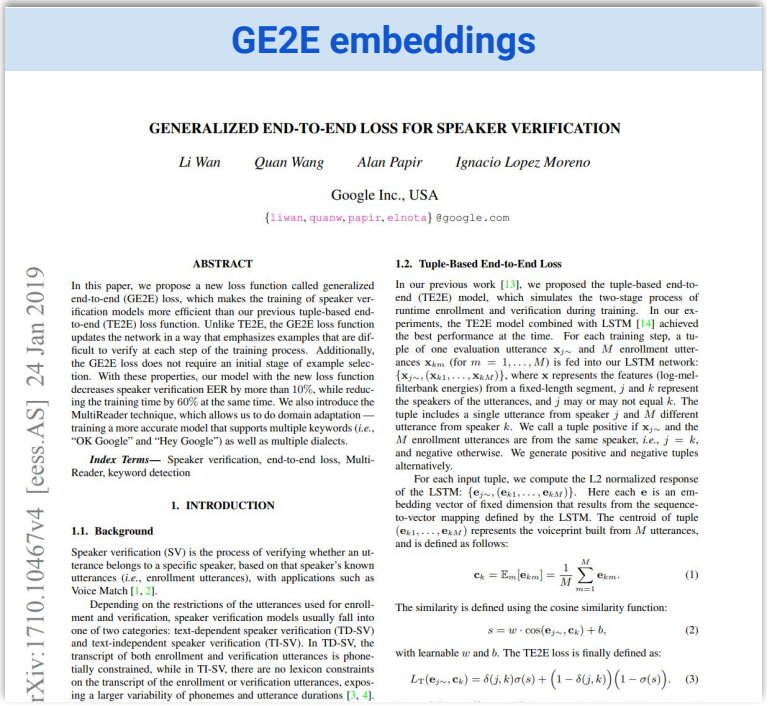
features should be similar **for the same speaker**
no matter what they say

What features should we use?

Speech embeddings for speaker verification

- **Pretrained:** great, because we only have 227 speakers to work with
- **Text-independent:** same speaker will always get similar embeddings
- **Meaningful:** similar voices are close to each other

GE2E embeddings



GENERALIZED END-TO-END LOSS FOR SPEAKER VERIFICATION

Li Wan *Quan Wang* *Alan Papir* *Ignacio Lopez Moreno*

Google Inc., USA

{liwan, quanw, papir, elnota} @google.com

ABSTRACT

In this paper, we propose a new loss function called generalized end-to-end (GE2E) loss, which makes the training of speaker verification models more efficient than our previous tuple-based end-to-end (TE2E) loss function. Unlike TE2E, the GE2E loss function updates the network in a way that emphasizes examples that are difficult to verify at each step of the training process. Additionally, the GE2E loss does not require an initial stage of example selection. With these properties, our model with the new loss function decreases speaker verification EER by more than 10%, while reducing the training time by 60% at the same time. We also introduce the MultiReader technique, which allows us to do domain adaptation — training a more accurate model that supports multiple keywords (*i.e.*, "OK Google" and "Hey Google") as well as multiple dialects.

***Index Terms*—** Speaker verification, end-to-end loss, MultiReader, keyword detection

1. INTRODUCTION

1.1. Background

Speaker verification (SV) is the process of verifying whether an utterance belongs to a specific speaker, based on that speaker's known utterances (*i.e.*, enrollment utterances), with applications such as Voice Match [1, 2].

Depending on the restrictions of the utterances used for enrollment and verification, speaker verification models usually fall into one of two categories: text-dependent speaker verification (TD-SV) and text-independent speaker verification (TI-SV). In TD-SV, the transcript of both enrollment and verification utterances is phonetically constrained, while in TI-SV, there are no lexicon constraints on the transcript of the enrollment or verification utterances, exposing a larger variability of phonemes and utterance durations [3, 4].

1.2. Tuple-Based End-to-End Loss

In our previous work [13], we proposed the tuple-based end-to-end (TE2E) model, which simulates the two-stage process of runtime enrollment and verification during training. In our experiments, the TE2E model combined with LSTM [14] achieved the best performance at the time. For each training step, a tuple of one evaluation utterance $\mathbf{x}_{j\sim}$ and M enrollment utterances $\mathbf{x}_{km}$ (for $m = 1, \dots, M$) is fed into our LSTM network: $\{\mathbf{x}_{j\sim}, (\mathbf{x}_{k1}, \dots, \mathbf{x}_{kM})\}$, where $\mathbf{x}$ represents the features (log-mel-filterbank energies) from a fixed-length segment, j and k represent the speakers of the utterances, and j may or may not equal k. The tuple includes a single utterance from speaker j and M different utterance from speaker k. We call a tuple positive if $\mathbf{x}_{j\sim}$ and the M enrollment utterances are from the same speaker, *i.e.*, $j = k$, and negative otherwise. We generate positive and negative tuples alternatively.

For each input tuple, we compute the L2 normalized response of the LSTM: $\{\mathbf{e}_{j\sim}, (\mathbf{e}_{k1}, \dots, \mathbf{e}_{kM})\}$. Here each $\mathbf{e}$ is an embedding vector of fixed dimension that results from the sequence-to-vector mapping defined by the LSTM. The centroid of tuple $(\mathbf{e}_{k1}, \dots, \mathbf{e}_{kM})$ represents the voiceprint built from M utterances, and is defined as follows:

$$\mathbf{c}_k = \mathbb{E}_m[\mathbf{e}_{km}] = \frac{1}{M}\sum_{m=1}^{M}\mathbf{e}_{km}. \quad (1)$$

The similarity is defined using the cosine similarity function:

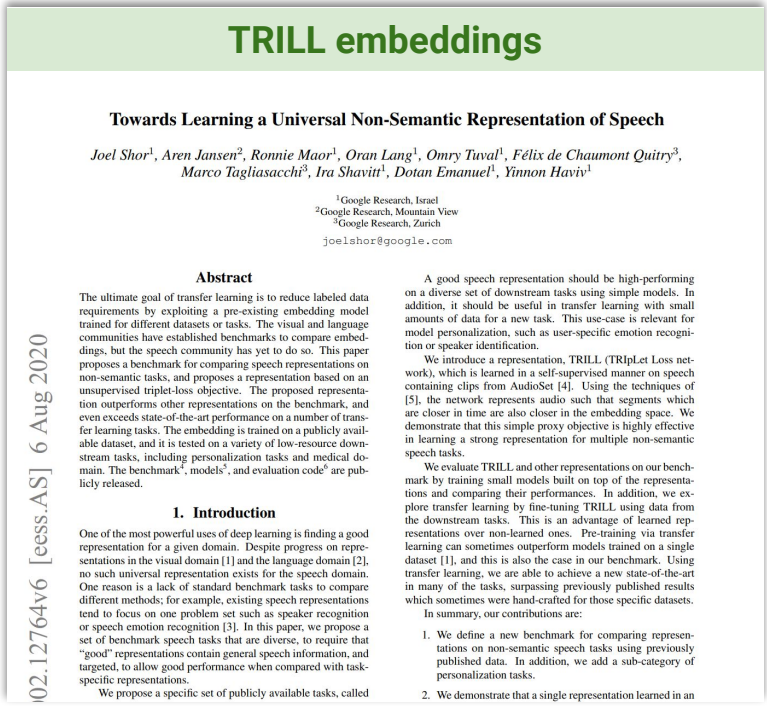
$$s = w \cdot \cos(\mathbf{e}_{j\sim}, \mathbf{c}_k) + b, \quad (2)$$

with learnable w and b. The TE2E loss is finally defined as:

$$L_T(\mathbf{e}_{j\sim}, \mathbf{c}_k) = \delta(j,k)\sigma(s) + \Big(1 - \delta(j,k)\Big)\Big(1 - \sigma(s)\Big). \quad (3)$$

(Wan et. al., 2019)

TRILL embeddings



Towards Learning a Universal Non-Semantic Representation of Speech

Joel Shor[1], *Aren Jansen*[2], *Ronnie Maor*[1], *Oran Lang*[1], *Omry Tuval*[1], *Félix de Chaumont Quitry*[3], *Marco Tagliasacchi*[3], *Ira Shavitt*[1], *Dotan Emanuel*[1], *Yinnon Haviv*[1]

[1]Google Research, Israel
[2]Google Research, Mountain View
[3]Google Research, Zurich

joelshor@google.com

Abstract

The ultimate goal of transfer learning is to reduce labeled data requirements by exploiting a pre-existing embedding model trained for different datasets or tasks. The visual and language communities have established benchmarks to compare embeddings, but the speech community has yet to do so. This paper proposes a benchmark for comparing speech representations on non-semantic tasks, and proposes a representation based on an unsupervised triplet-loss objective. The proposed representation outperforms other representations on the benchmark, and even exceeds state-of-the-art performance on a number of transfer learning tasks. The embedding is trained on a publicly available dataset, and it is tested on a variety of low-resource downstream tasks, including personalization tasks and medical domain. The benchmark[4], models[5], and evaluation code[6] are publicly released.

1. Introduction

One of the most powerful uses of deep learning is finding a good representation for a given domain. Despite progress on representations in the visual domain [1] and the language domain [2], no such universal representation exists for the speech domain. One reason is a lack of standard benchmark tasks to compare different methods; for example, existing speech representations tend to focus on one problem set such as speaker recognition or speech emotion recognition [3]. In this paper, we propose a set of benchmark speech tasks that are diverse, to require that "good" representations contain general speech information, and targeted, to allow good performance when compared with task-specific representations.

We propose a specific set of publicly available tasks, called

A good speech representation should be high-performing on a diverse set of downstream tasks using simple models. In addition, it should be useful in transfer learning with small amounts of data for a new task. This use-case is relevant for model personalization, such as user-specific emotion recognition or speaker identification.

We introduce a representation, TRILL (TRIpLet Loss network), which is learned in a self-supervised manner on speech containing clips from AudioSet [4]. Using the techniques of [5], the network represents audio such that segments which are closer in time are also closer in the embedding space. We demonstrate that this simple proxy objective is highly effective in learning a strong representation for multiple non-semantic speech tasks.

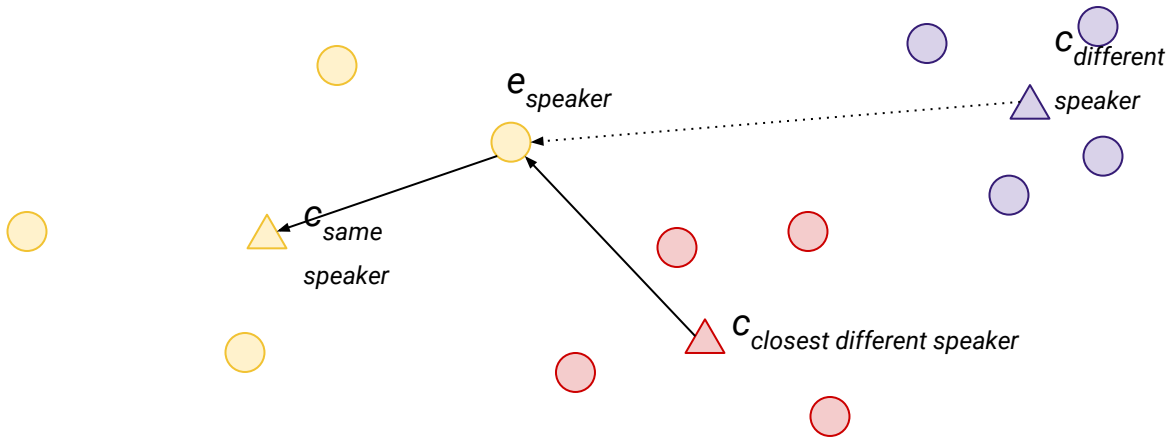
We evaluate TRILL and other representations on our benchmark by training small models built on top of the representations and comparing their performances. In addition, we explore transfer learning by fine-tuning TRILL using data from the downstream tasks. This is an advantage of learned representations over non-learned ones. Pre-training via transfer learning can sometimes outperform models trained on a single dataset [1], and this is also the case in our benchmark. Using transfer learning, we are able to achieve a new state-of-the-art in many of the tasks, surpassing previously published results which sometimes were hand-crafted for those specific datasets.

In summary, our contributions are:

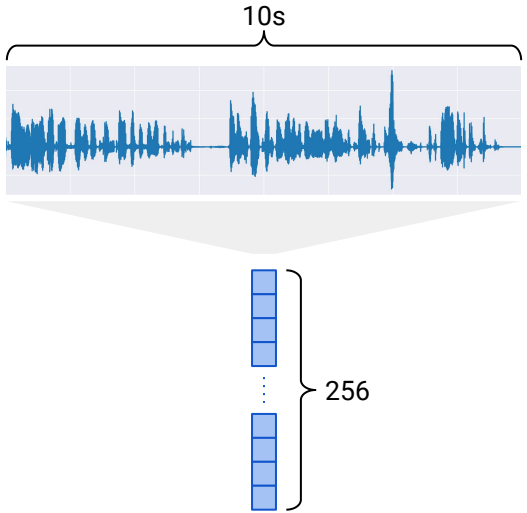
1. We define a new benchmark for comparing representations on non-semantic speech tasks using previously published data. In addition, we add a sub-category of personalization tasks.
2. We demonstrate that a single representation learned in an

(Shor et. al., 2020)

Generalized End-to-End loss (GE2E)



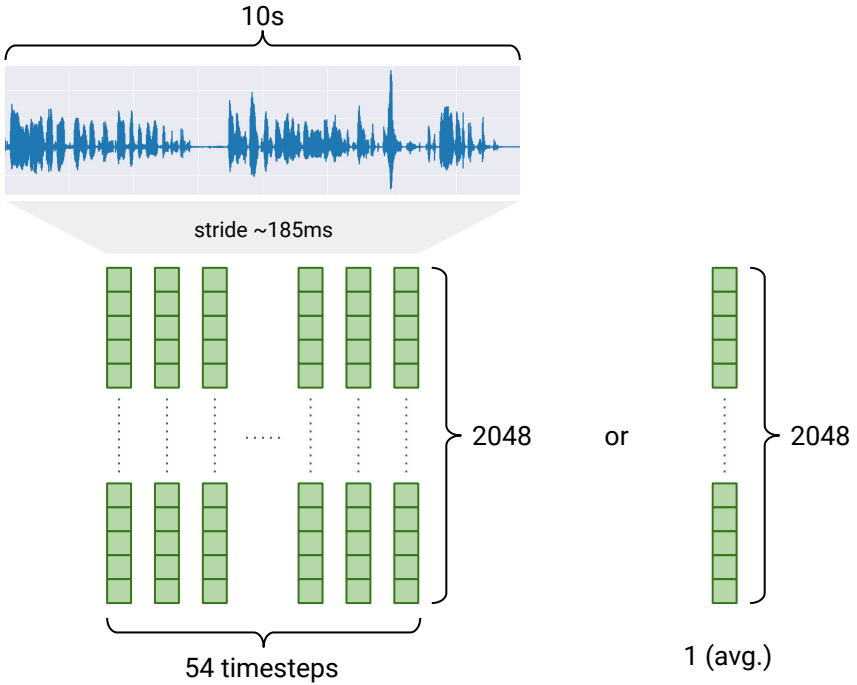
adapted from Wan et. al. (2019)



GE2E embeddings

TRIpLet Loss (TRILL)

- intendend for all kinds of paralinguistic tasks, not just speaker verification
- more or less the same as GE2E (calculated on triplets of anchor, positive, and negative)
- much higher dimensionality (2048)
- one embedding per ~180ms
- beats SOTA on some paralinguistic tasks

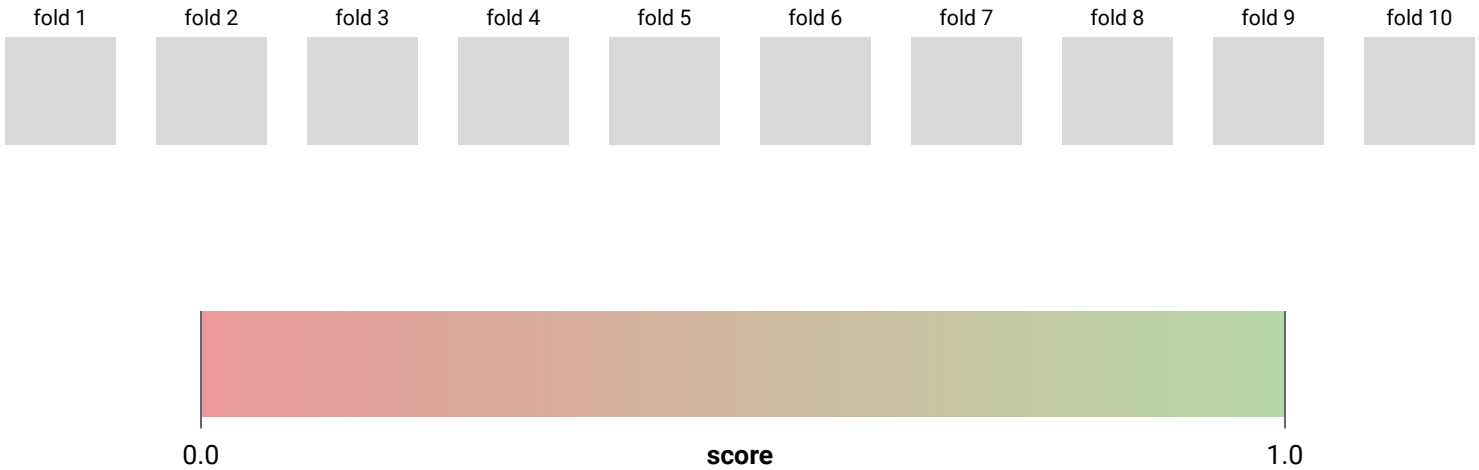


TRILL embeddings

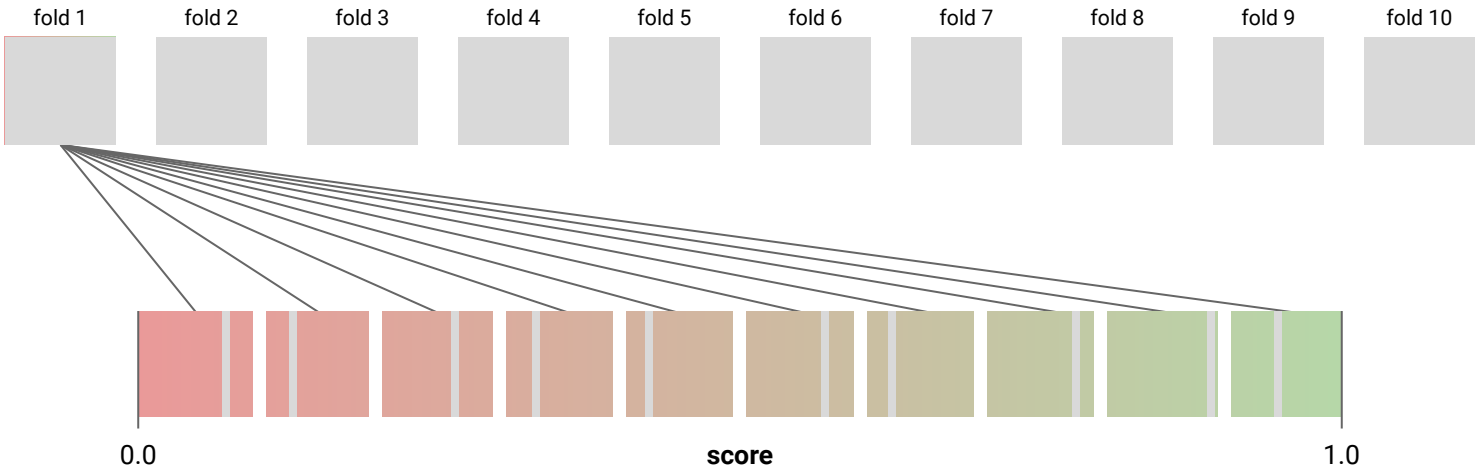
Creating embeddings for all audio samples

```
Updating trill type embeddings in directory embeddings-trill.
🔊 There are 9962 .wav files inside wavs.
Starting update ...

100%|████████████████████████████████████████| 9925/9962 [00:01<00:00, 9867.54it/s]

✔️Done
9922 embeddings not modified, 0 deleted, 3 created
Loaded encoder "pretrained.pt" trained to step 1564501

Updating ge2e type embeddings in directory embeddings-ge2e.
🔊 There are 9962 .wav files inside wavs.
Starting update ...

100%|████████████████████████████████████████| 9925/9962 [00:14<00:00, 678.26it/s]

✔️Done
9922 embeddings not modified, 0 deleted, 3 created
```

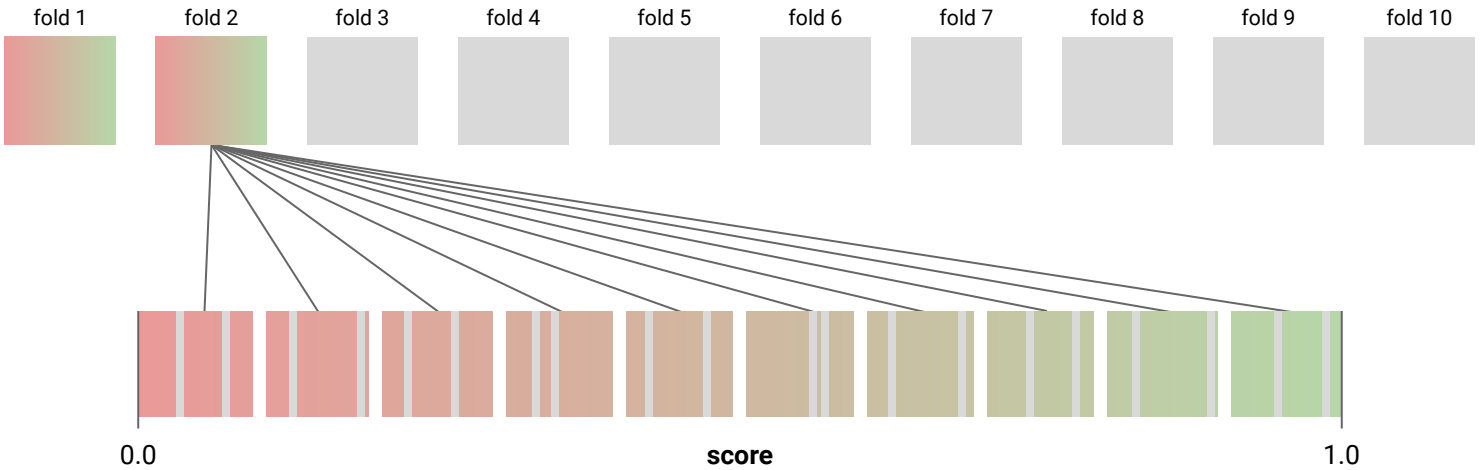
Stratified 10-fold cross-validation



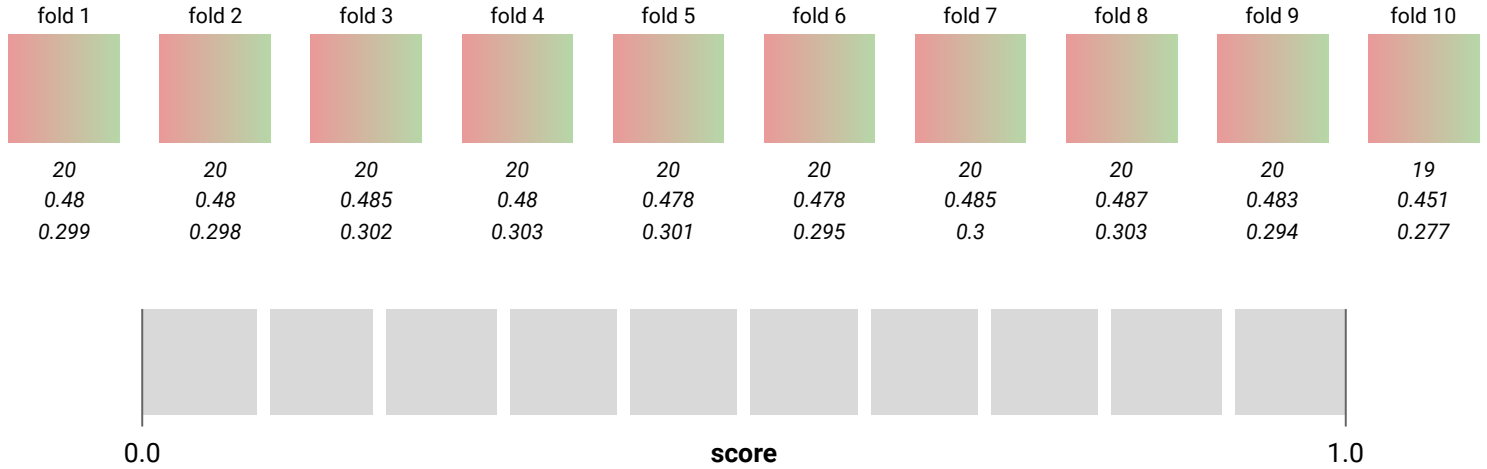
Stratified 10-fold cross-validation



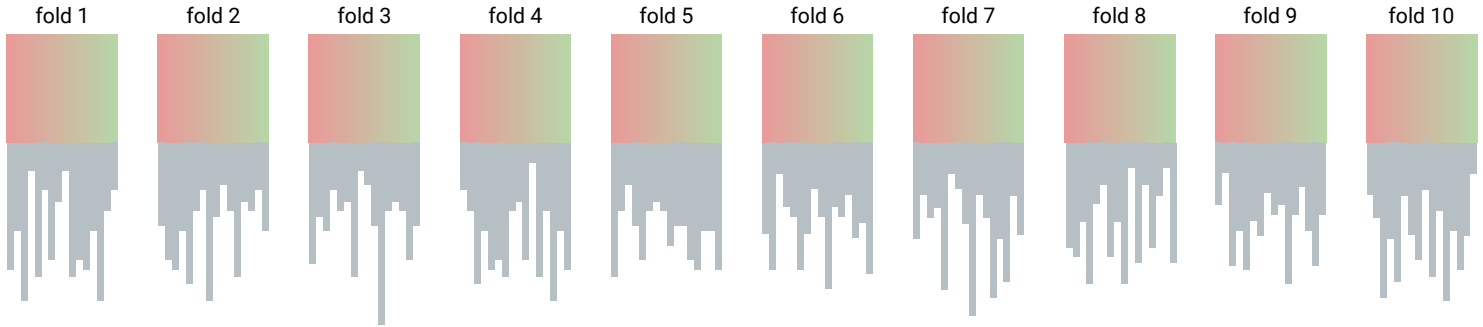
Stratified 10-fold cross-validation



Stratified 10-fold cross-validation



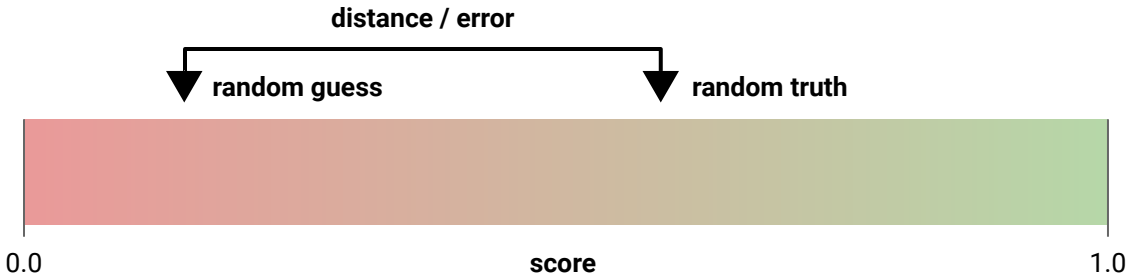
Stratified 10-fold cross-validation



For each speaker in each fold: draw between 10 and 100 random audio samples (depending on how much each person spoke)

neckbreaker 💀

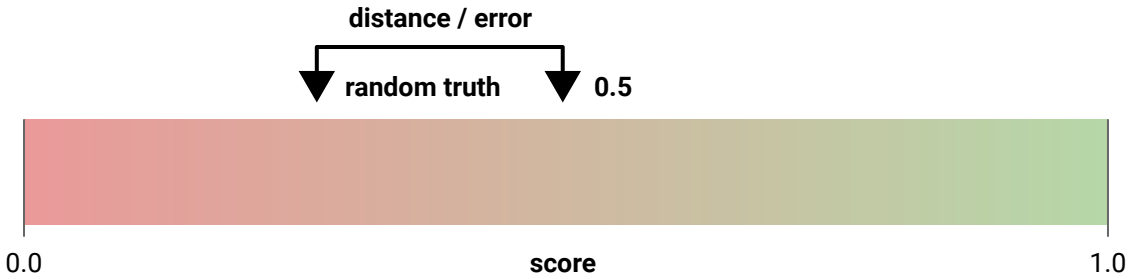
Baseline: random guessing



average distance: 0.3333…

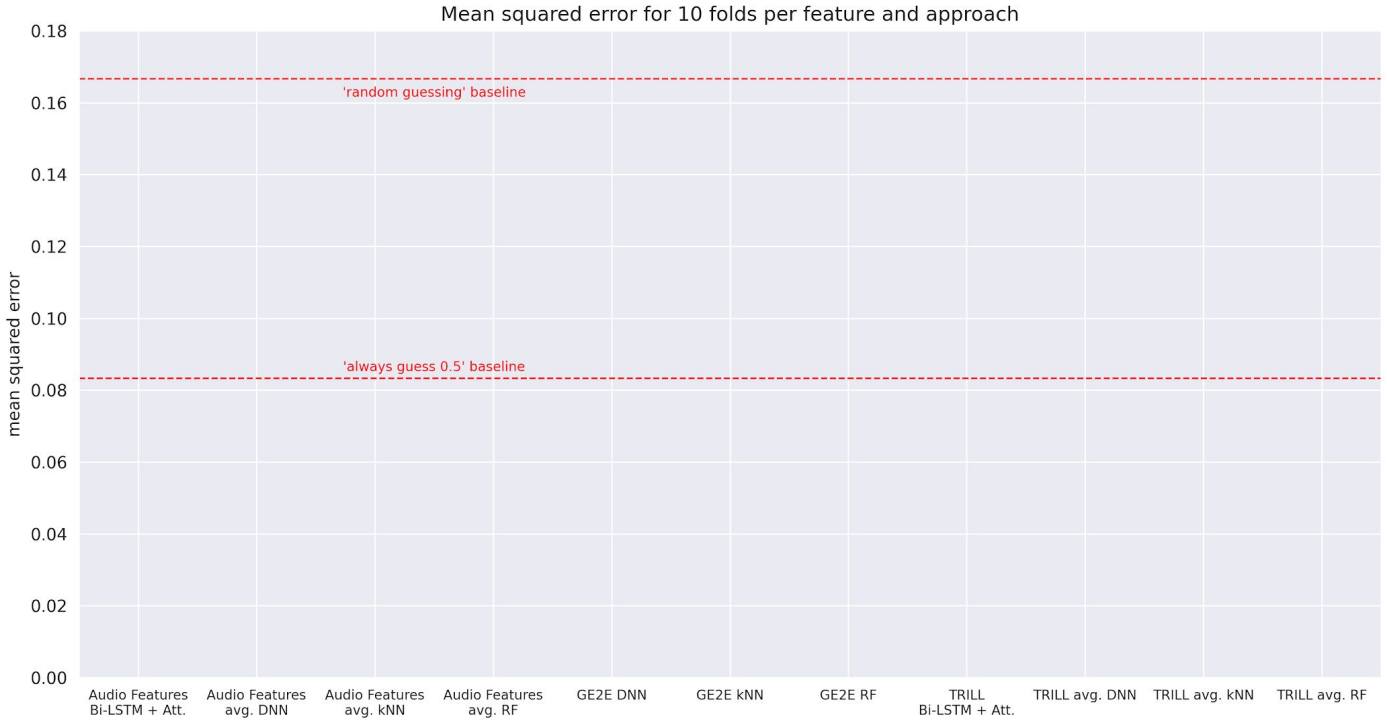
mean squared error: 0.1666…

Baseline: always guess 0.5



average distance: 0.25

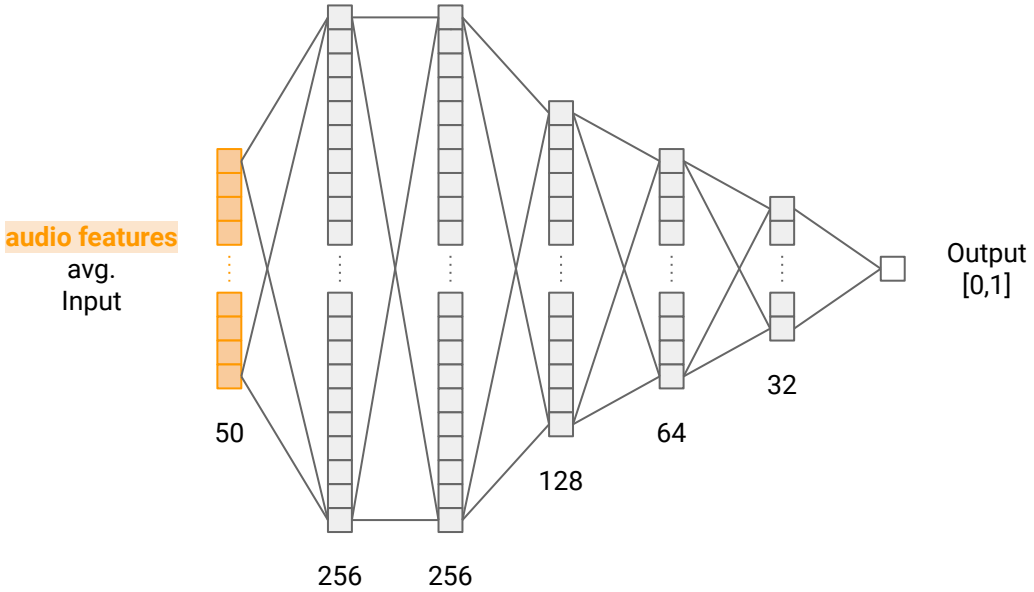
mean squared error: 0.0833…


Mean squared error for 10 folds per feature and approach
mean squared error
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
'random guessing' baseline
'always guess 0.5' baseline
Audio Features Bi-LSTM + Att.
Audio Features avg. DNN
Audio Features avg. kNN
Audio Features avg. RF
GE2E DNN
GE2E kNN
GE2E RF
TRILL Bi-LSTM + Att.
TRILL avg. DNN
TRILL avg. kNN
TRILL avg. RF

Evaluation grid

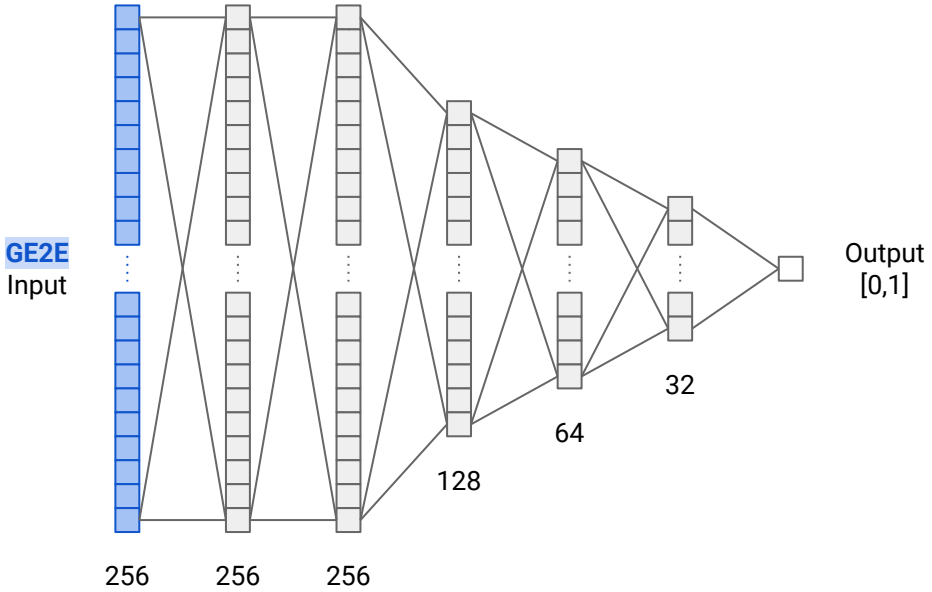
	timeseries	single input vector		
	Bi-LSTM + Att.	**DNN**	**kNN**	**RF**
audio features				
GE2E embeddings	–			
TRILL embeddings				

Audio features DNN



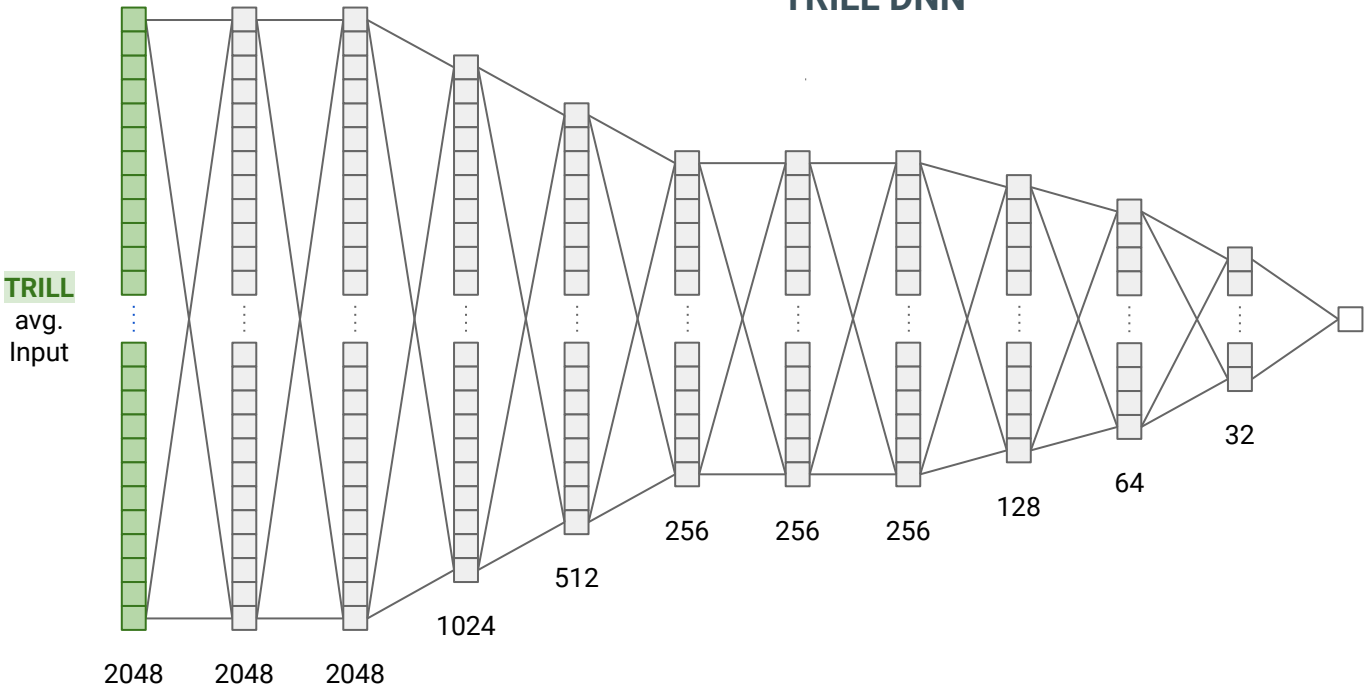
Dropout	0.5
Loss	mse
Act. func.	tanh
Optimizer	adamax
Batch size	20
Learning rate	5e-8

GE2E DNN



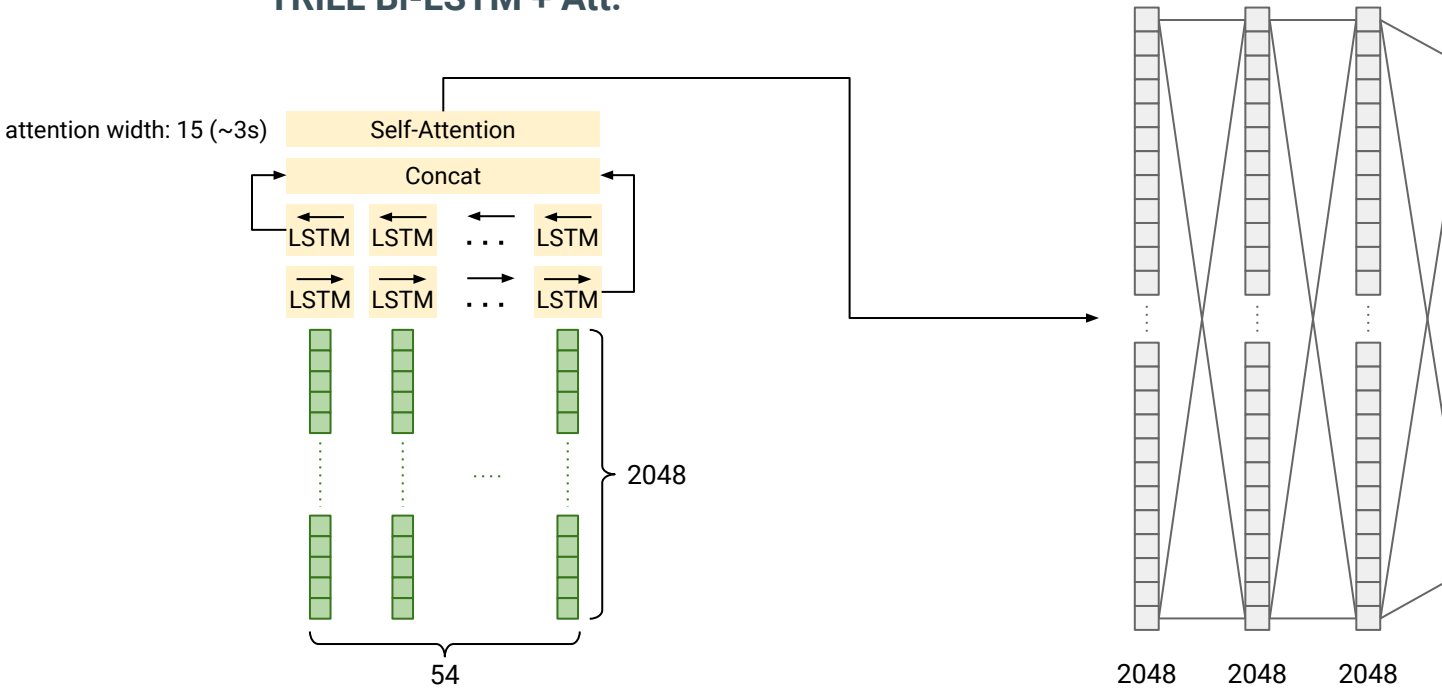
Dropout	0.5
Loss	mse
Act. func.	tanh
Optimizer	adamax
Batch size	20
Learning rate	5e-8

TRILL DNN

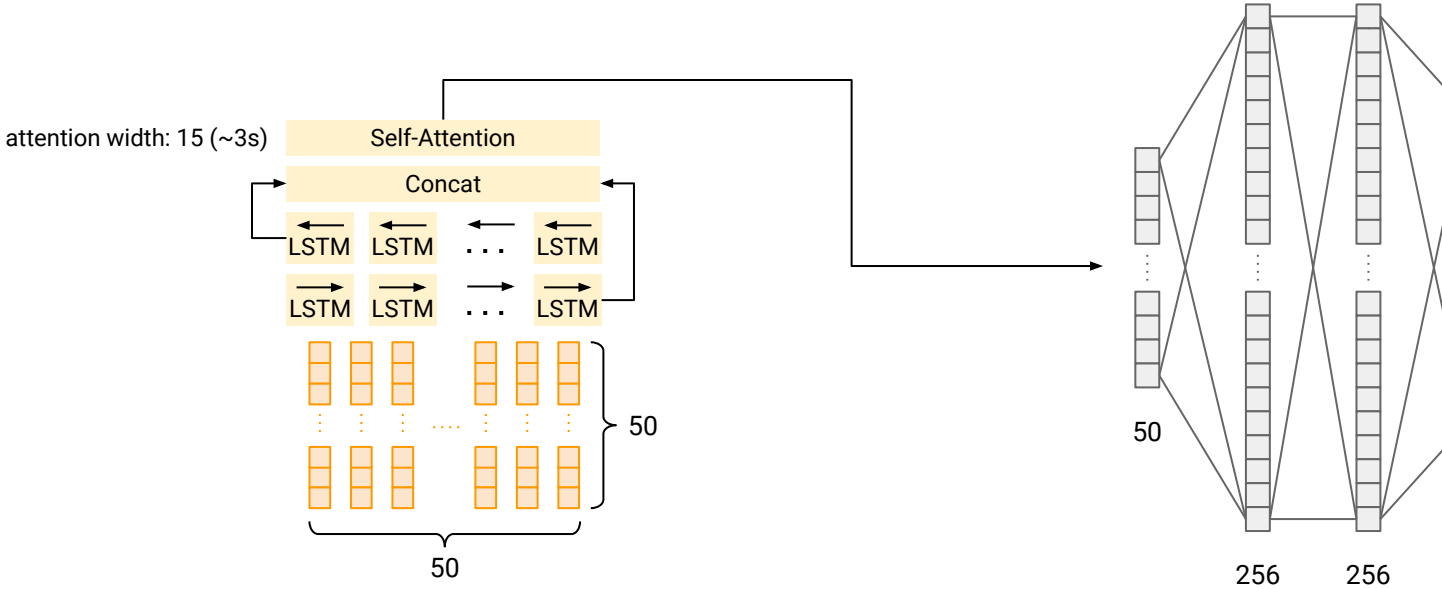


Dropout	0.5
Loss	mse
Act. func.	tanh
Optimizer	adamax
Batch size	20
Learning rate	5e-8

TRILL Bi-LSTM + Att.



Audio feature Bi-LSTM + Att.



xkcd, https://xkcd.com/1838/

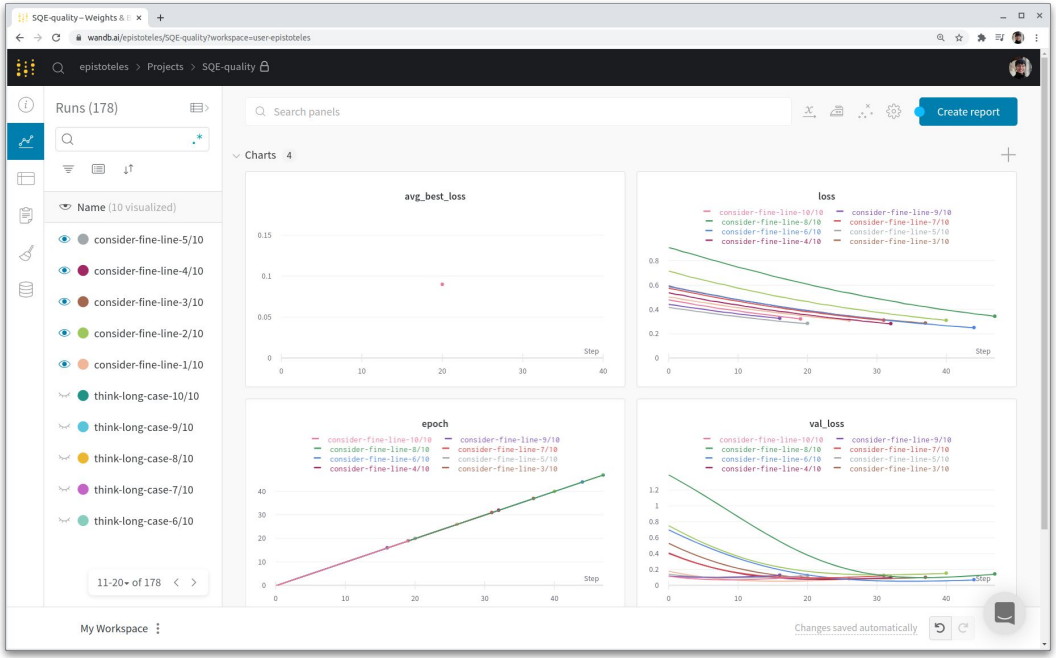
Method used for optimization

trying ...
and trying ...
and trying again ...

then take the best
performing parameters

proper grid search not feasible
(TRILL Bi-LSTM took ~34 h to train)

Model tracking using W&B (Weights&Biases)



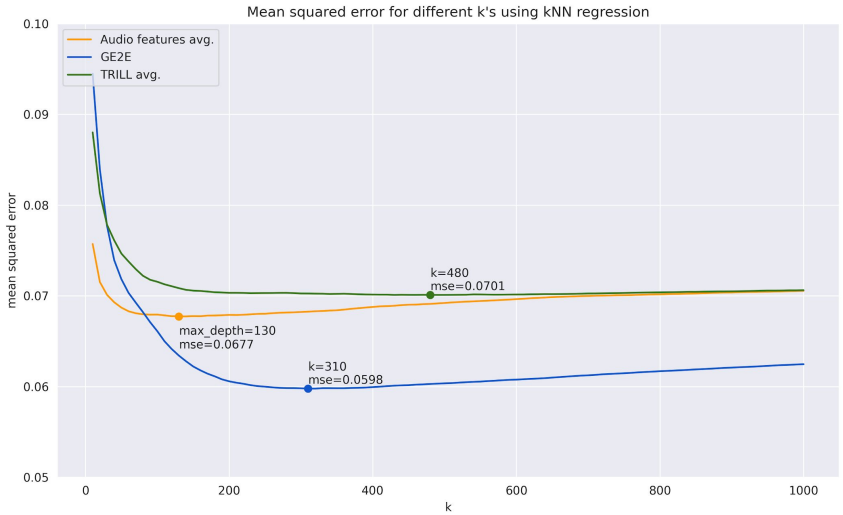
k-Nearest-Neighbors

Search over k:
10-1000 in steps of 10

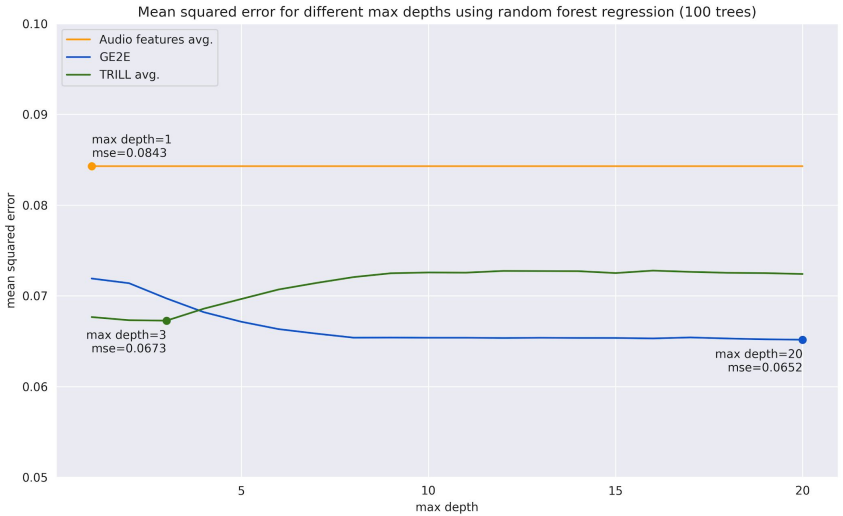
Random Forests (100 trees)

Search over maximum depth:
1-20 in steps of 1

k-Nearest-Neighbors

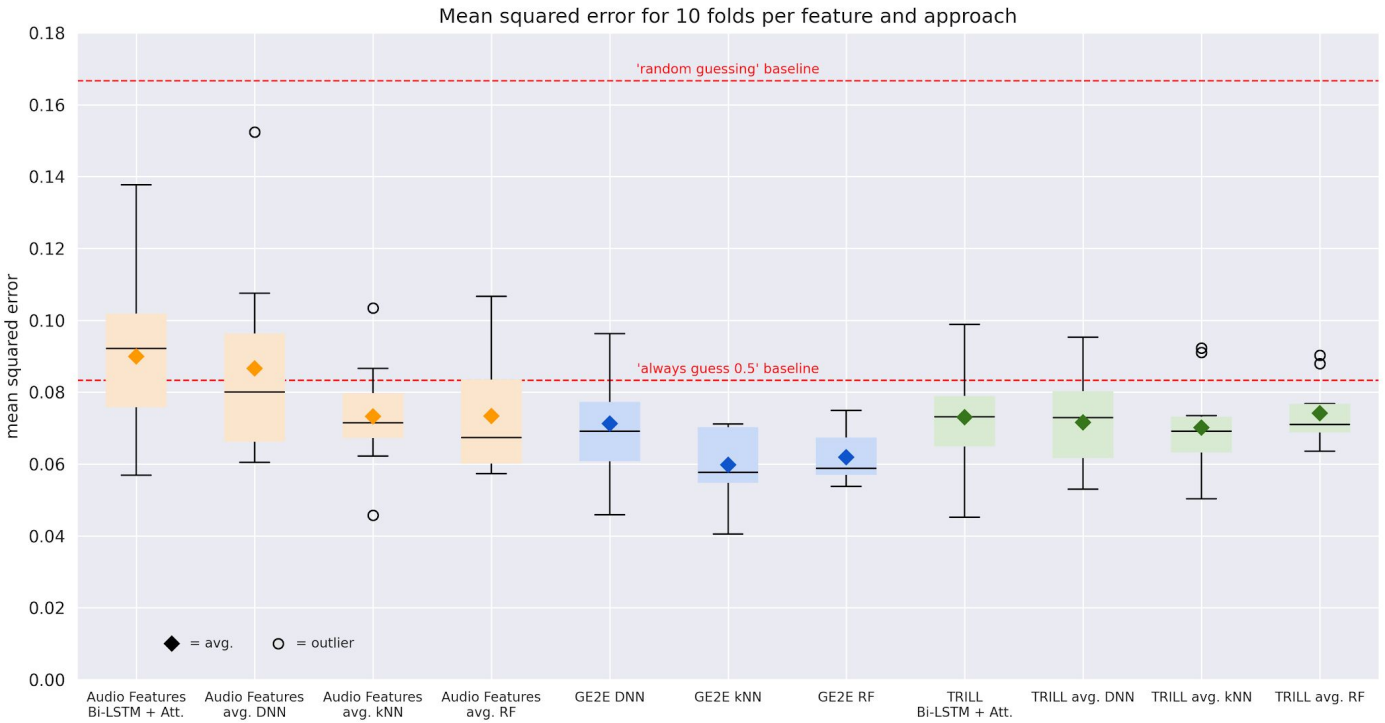


Random Forests



Best mean square error for stratified 10-fold cross-validation

	timeseries	single input vector		
	Bi-LSTM + Att.	**DNN**	**kNN**	**RF**
audio features	0.0899	0.0866	0.0732	0.0734
GE2E embeddings	–	0.0712	0.0598	0.0619
TRILL embeddings	0.0730	0.0716	0.0701	0.0742


Mean squared error for 10 folds per feature and approach
mean squared error
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
'random guessing' baseline
'always guess 0.5' baseline
= avg.
= outlier
Audio Features Bi-LSTM + Att.
Audio Features avg. DNN
Audio Features avg. kNN
Audio Features avg. RF
GE2E DNN
GE2E kNN
GE2E RF
TRILL Bi-LSTM + Att.
TRILL avg. DNN
TRILL avg. kNN
TRILL avg. RF

However, those baselines are wrong ...

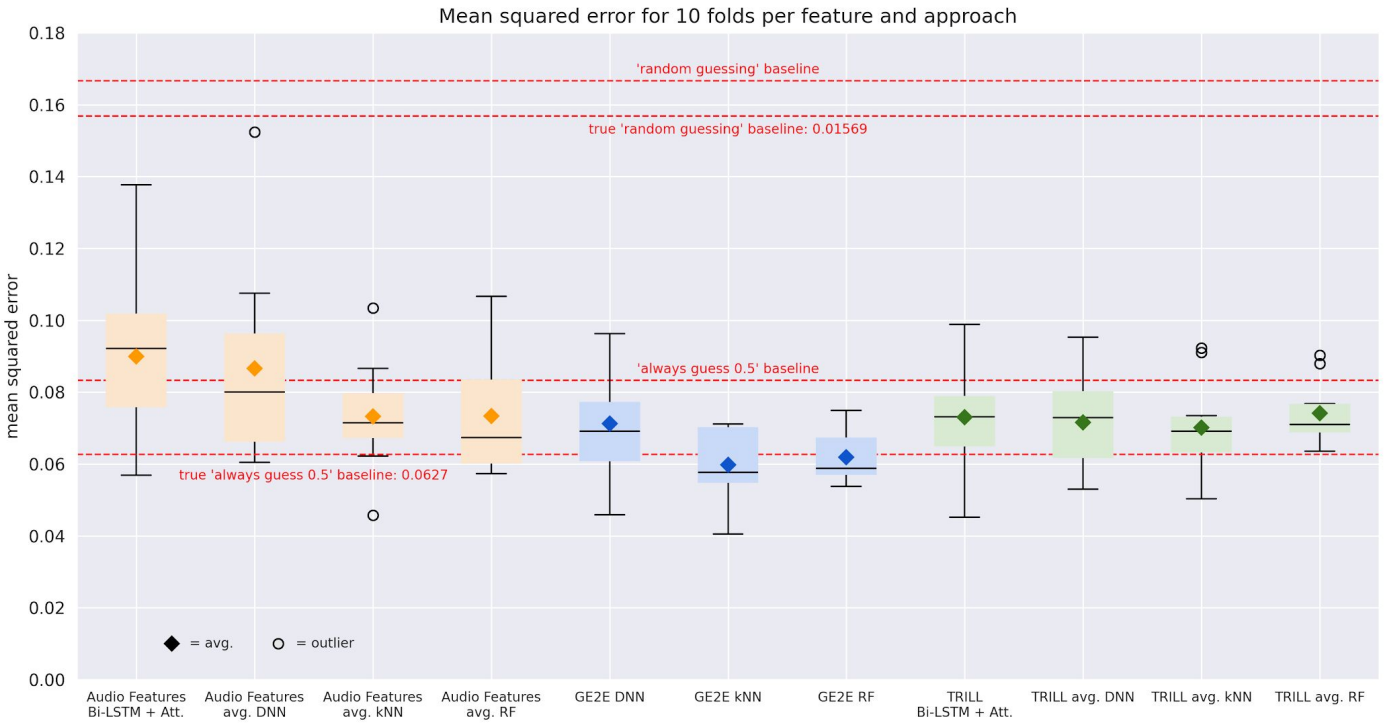
Why?

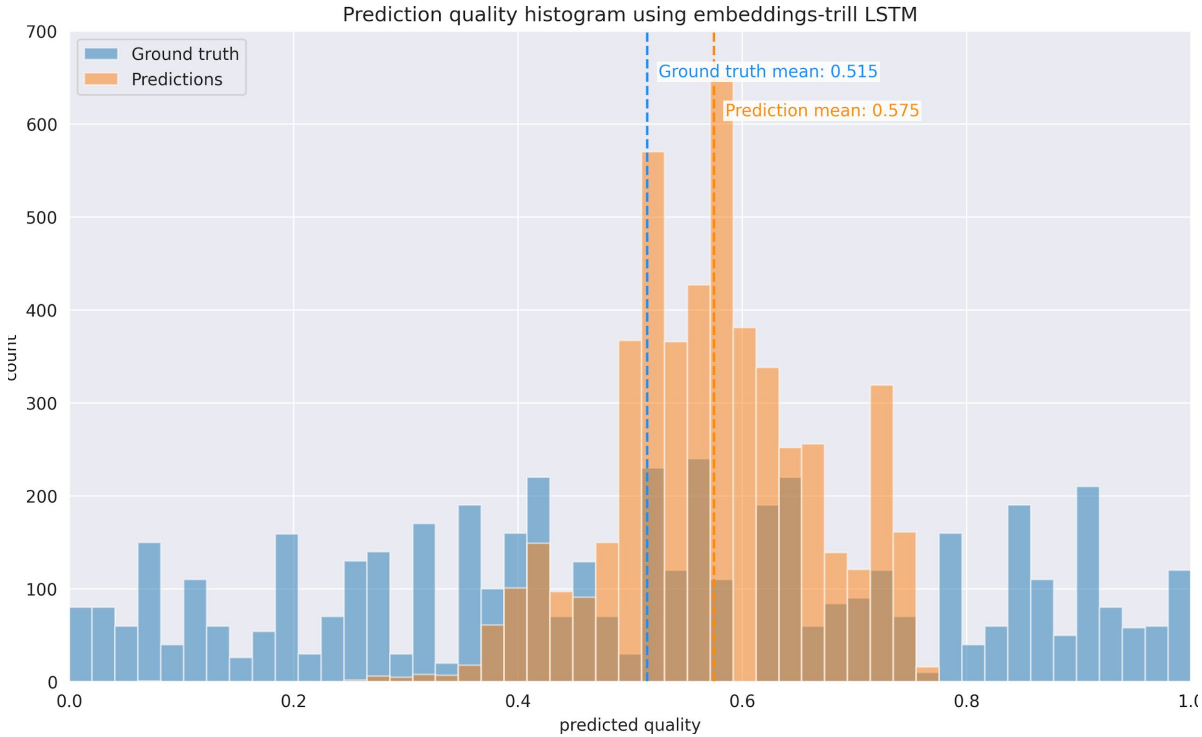
For each speaker in each fold: draw between 10 and 100 random audio samples (depending on how much each person spoke)

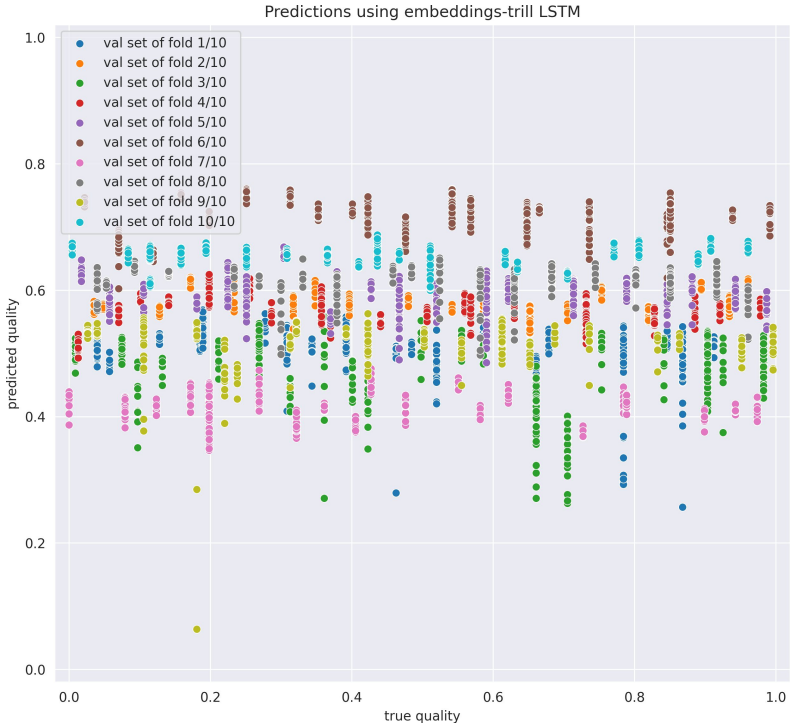
neckbreaker 💀

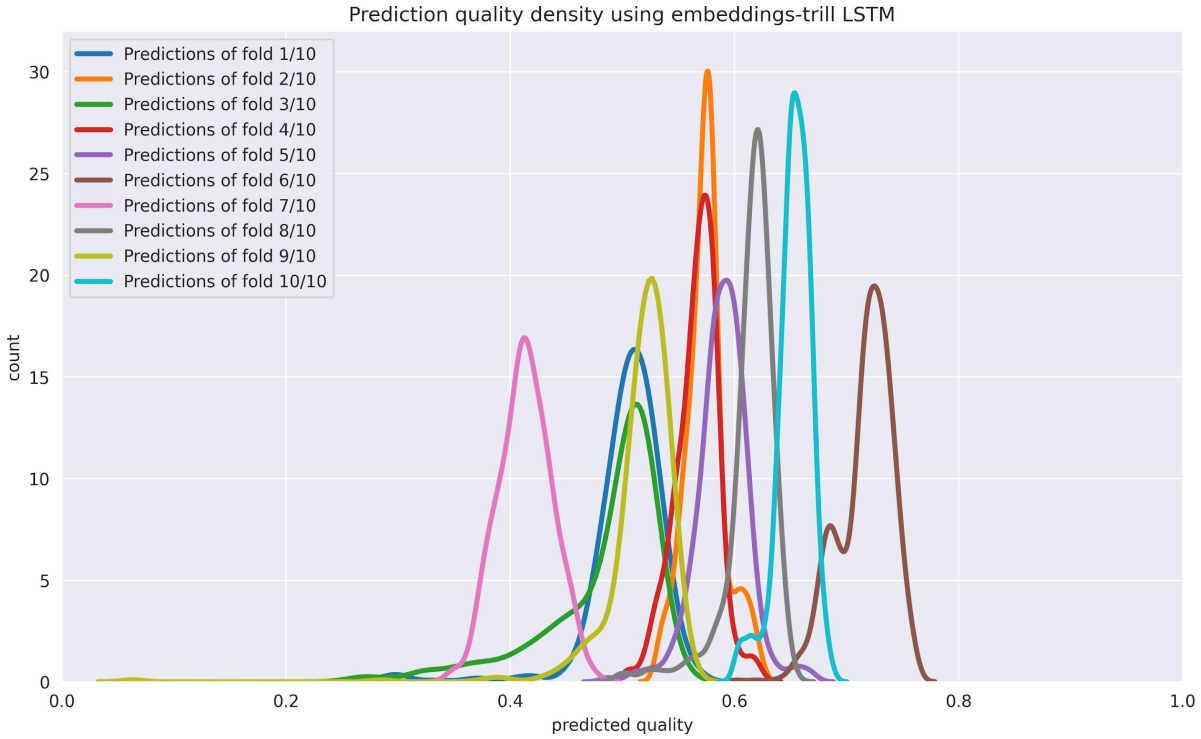
=> means are shifted slightly up or down in each test set

=> baselines on test sets are 0.1569 and 0.0627


Mean squared error for 10 folds per feature and approach
mean squared error
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
'random guessing' baseline
true 'random guessing' baseline: 0.01569
'always guess 0.5' baseline
true 'always guess 0.5' baseline: 0.0627
= avg.
= outlier
Audio Features Bi-LSTM + Att.
Audio Features avg. DNN
Audio Features avg. kNN
Audio Features avg. RF
GE2E DNN
GE2E kNN
GE2E RF
TRILL Bi-LSTM + Att.
TRILL avg. DNN
TRILL avg. kNN
TRILL avg. RF


Prediction quality histogram using embeddings-trill LSTM
Ground truth
Predictions
Ground truth mean: 0.515
Prediction mean: 0.575
count
predicted quality
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8


Predictions using embeddings-trill LSTM
val set of fold 1/10
val set of fold 2/10
val set of fold 3/10
val set of fold 4/10
val set of fold 5/10
val set of fold 6/10
val set of fold 7/10
val set of fold 8/10
val set of fold 9/10
val set of fold 10/10
predicted quality
true quality
0.0
0.2
0.4
0.6
0.8
1.0

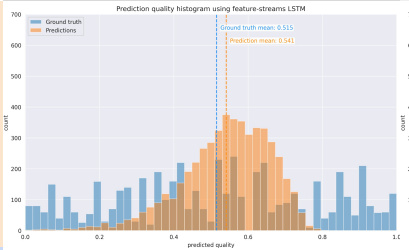
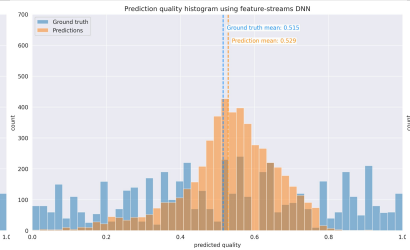
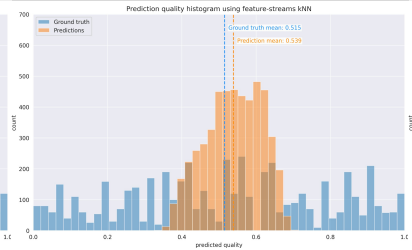
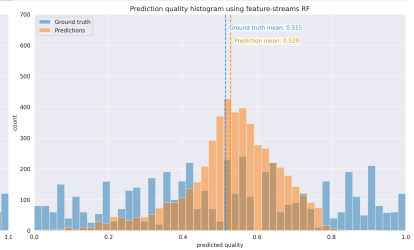
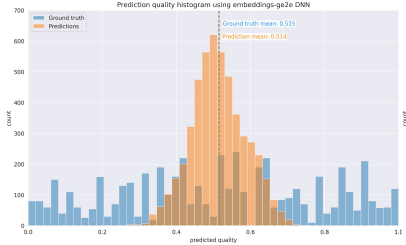
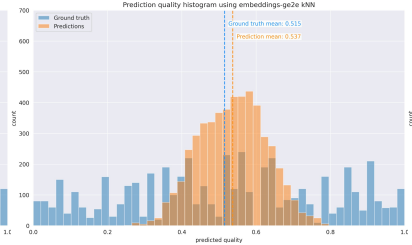
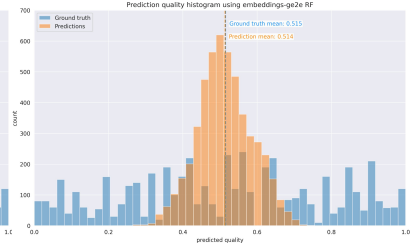
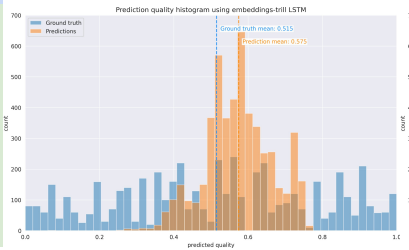
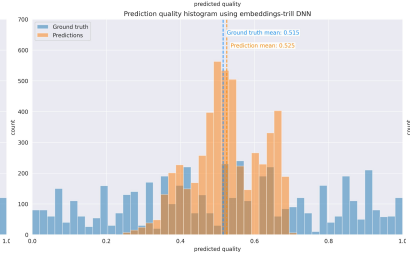
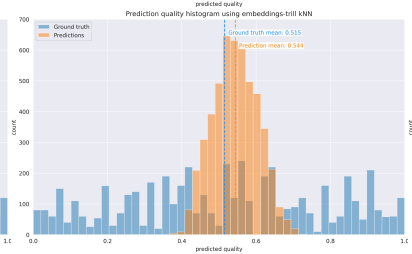

Prediction quality density using embeddings-trill LSTM
Predictions of fold 1/10
Predictions of fold 2/10
Predictions of fold 3/10
Predictions of fold 4/10
Predictions of fold 5/10
Predictions of fold 6/10
Predictions of fold 7/10
Predictions of fold 8/10
Predictions of fold 9/10
Predictions of fold 10/10
count
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0
predicted quality

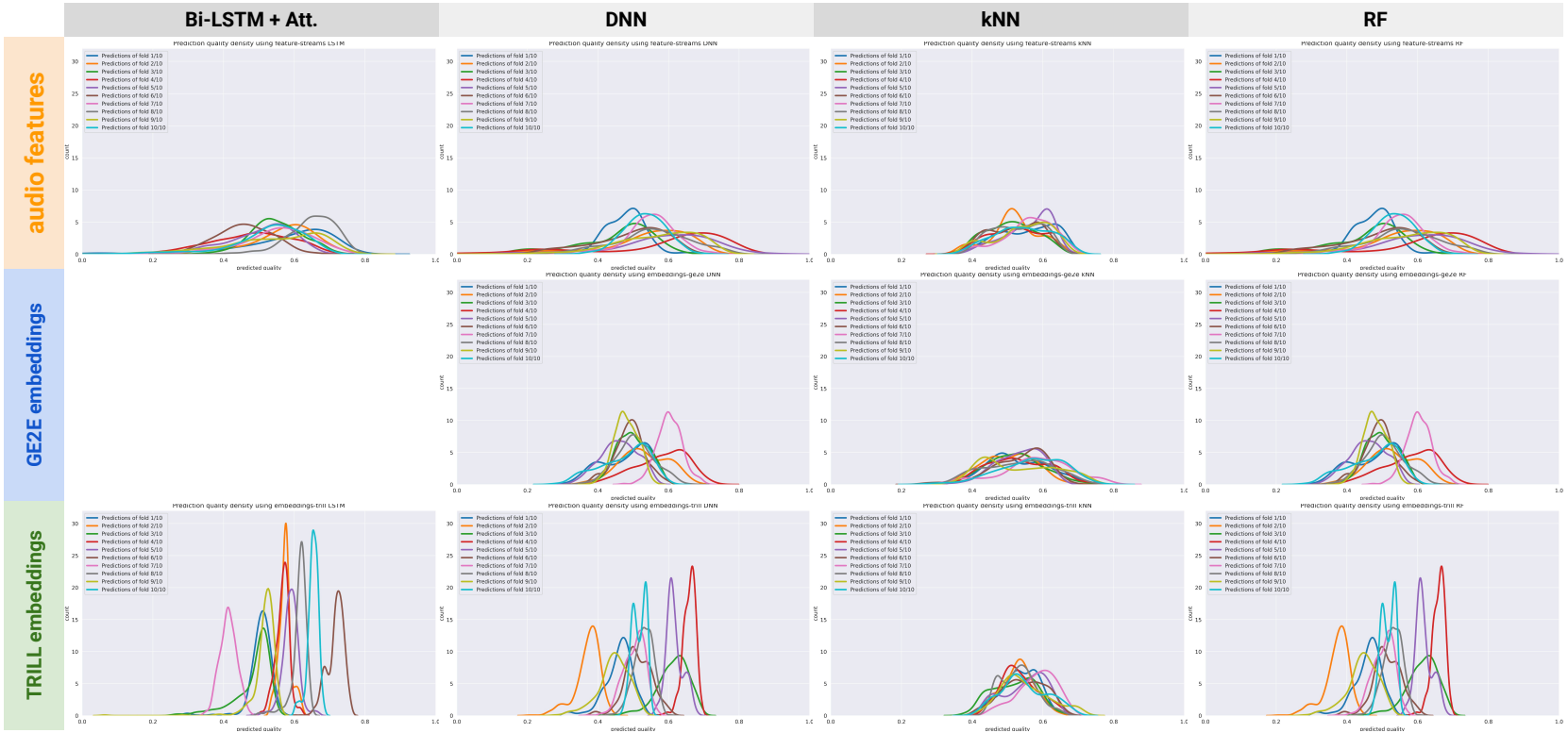
What should I have done?

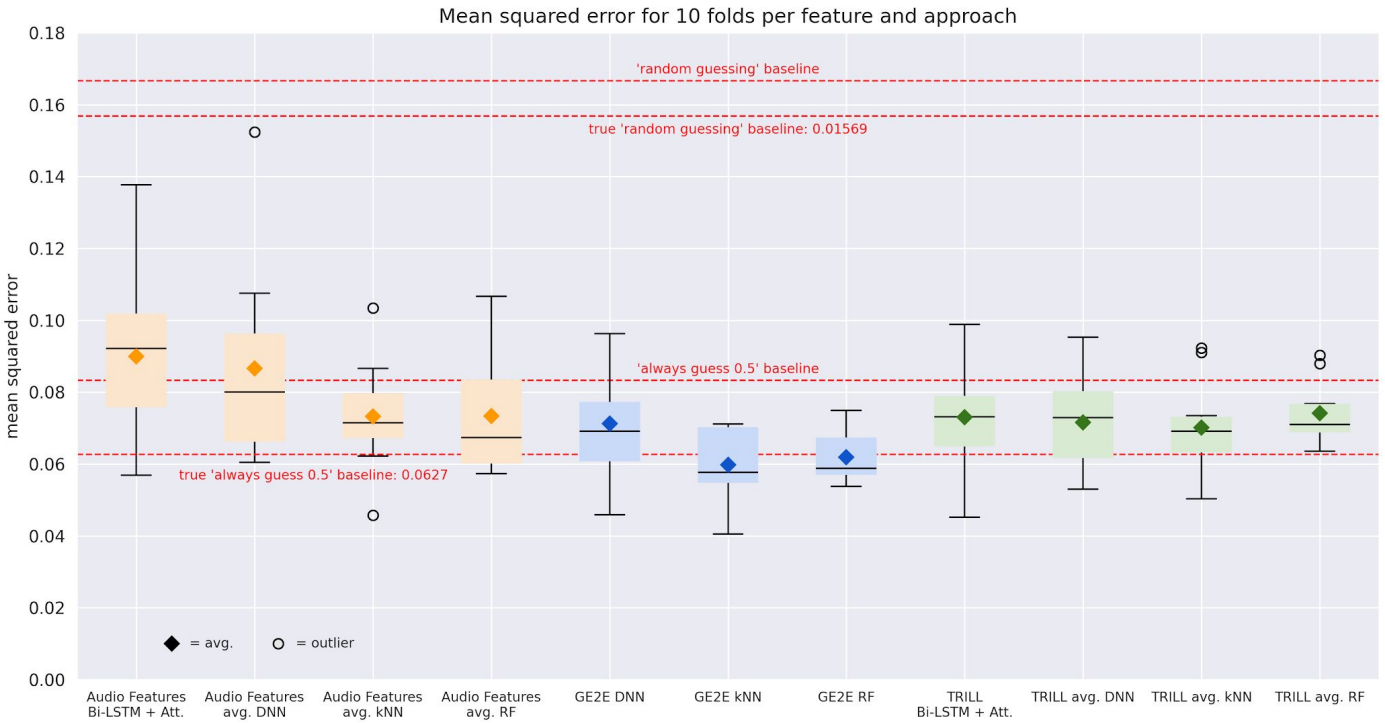
For each speaker in each fold: draw **exactly 10** random audio samples

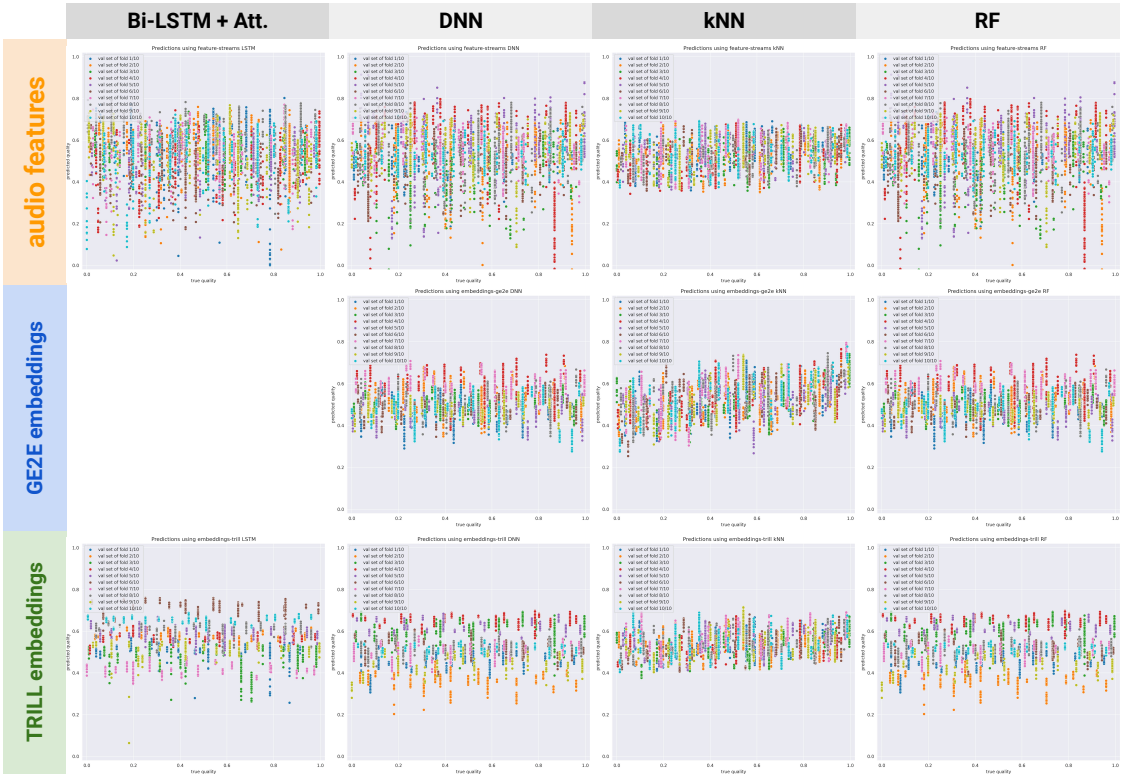
=> means stay where they are

=> baselines on test sets stay 0.1656 and 0.0843

	Bi-LSTM + Att.	DNN	kNN	RF
audio features				
GE2E embeddings				
TRILL embeddings				


Bi-LSTM + Att.
DNN
kNN
RF
audio features
GE2E embeddings
TRILL embeddings

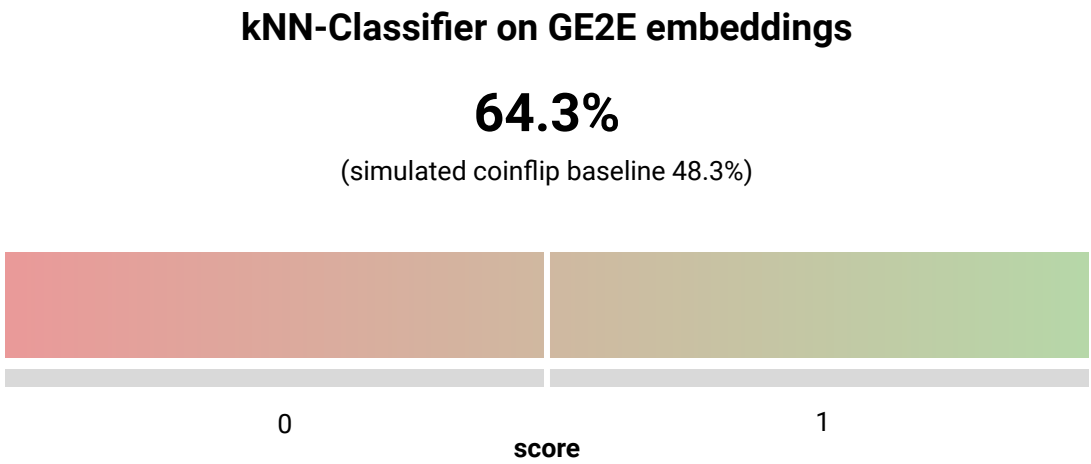

Mean squared error for 10 folds per feature and approach
mean squared error
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
'random guessing' baseline
true 'random guessing' baseline: 0.01569
'always guess 0.5' baseline
true 'always guess 0.5' baseline: 0.0627
= avg.
= outlier
Audio Features Bi-LSTM + Att.
Audio Features avg. DNN
Audio Features avg. kNN
Audio Features avg. RF
GE2E DNN
GE2E kNN
GE2E RF
TRILL Bi-LSTM + Att.
TRILL avg. DNN
TRILL avg. kNN
TRILL avg. RF


Bi-LSTM + Att.
DNN
kNN
RF
audio features
GE2E embeddings
TRILL embeddings

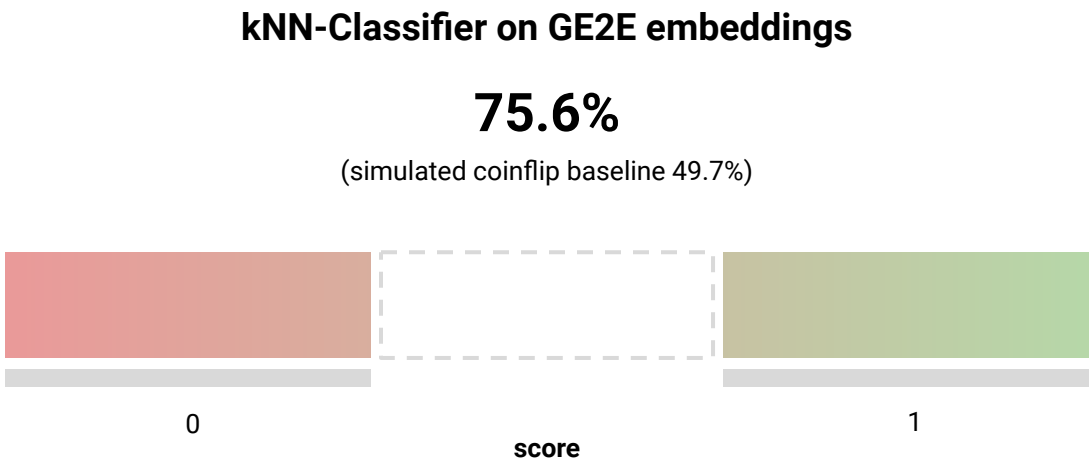

Predictions using embeddings-ge2e kNN
val set of fold 1/10
val set of fold 2/10
val set of fold 3/10
val set of fold 4/10
val set of fold 5/10
val set of fold 6/10
val set of fold 7/10
val set of fold 8/10
val set of fold 9/10
val set of fold 10/10
predicted quality
true quality
0.0
0.2
0.4
0.6
0.8
1.0

$r^2 = 0.451$

Further validation: binary prediction accuracy



Further validation: binary prediction accuracy



Answer to our research question

Yes, pre-trained speech embeddings can be used to predict the likability of speech, **independent of the spoken text**.

They are however only able to explain in to a certain degree.

Demo

	GE2E DNN	TRILL DNN	GE2E kNN	TRILL kNN	Class	Class w/o middle
DE_1234.wav	0.5024	0.5438	0.4308	0.5624	0	0
korbinian.wav	0.5936	0.5279	0.5345	0.5516	1	1
mosche.wav	0.4580	0.5166	0.5048	0.4957	0	0

Questions

Jorge Cham, http://phdcomics.com/comics/archive.php?comicid=588

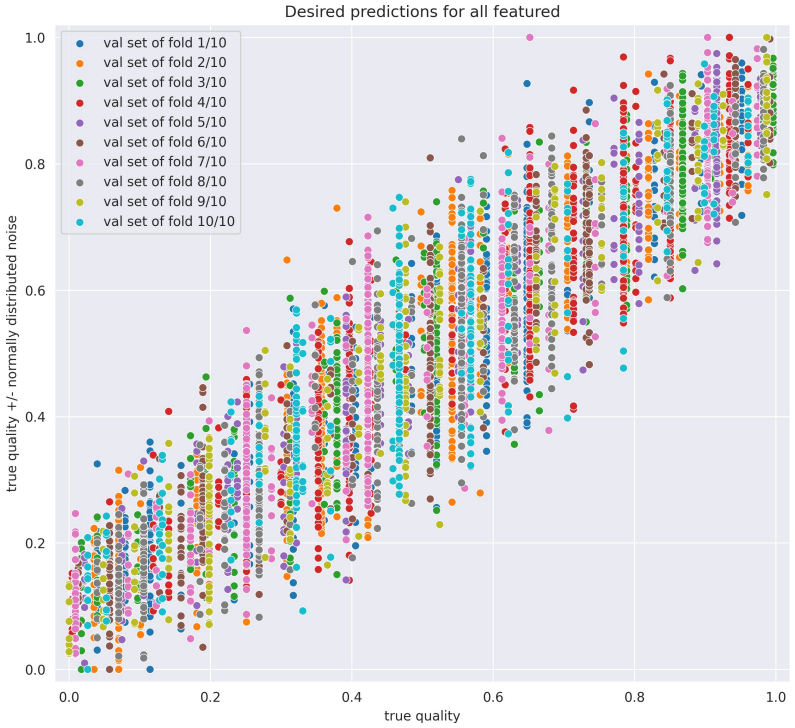
References

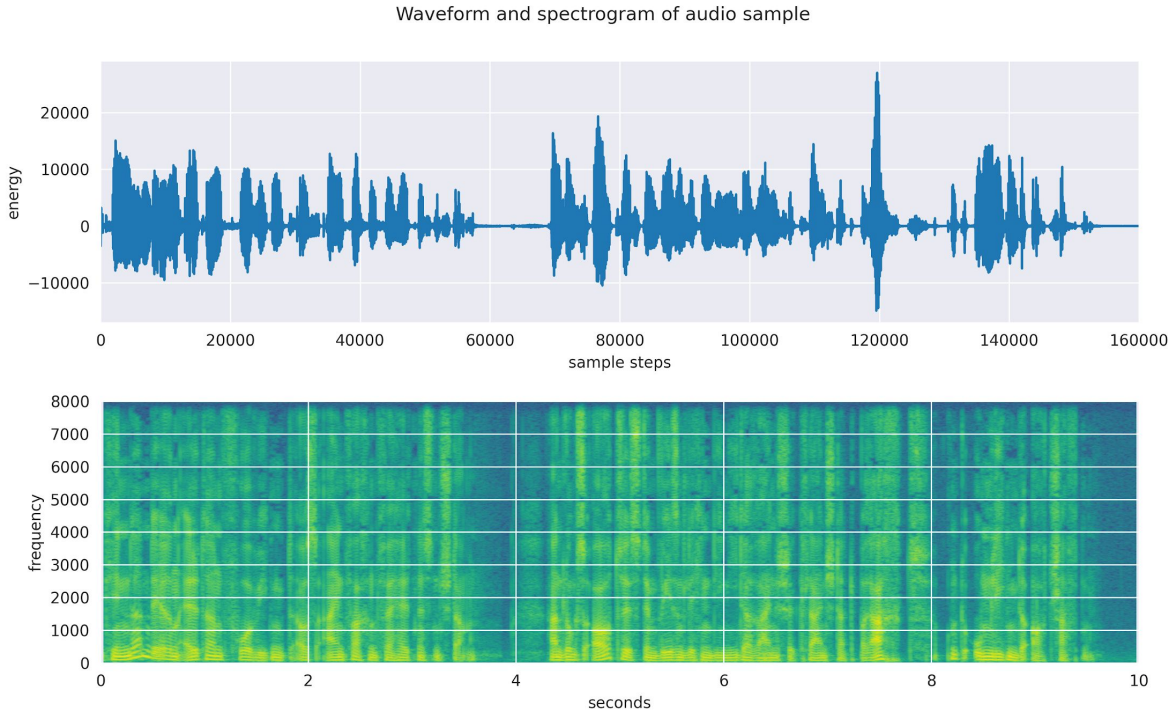
Shor et. al. (2020)
Shor, J., Jansen, A., Maor, R., Lang, O., Tuval, O., Quitry, F.D., Tagliasacchi, M., Shavitt, I., Emanuel, D., & Haviv, Y.A. (2020). Towards Learning a Universal Non-Semantic Representation of Speech. ArXiv, abs/2002.12764.

Baumann (2018)
Baumann, T. (2018) Learning to Determine Who is the Better Speaker. Proc. 9th International Conference on Speech Prosody 2018, 819-822, DOI: 10.21437/SpeechProsody.2018-165.

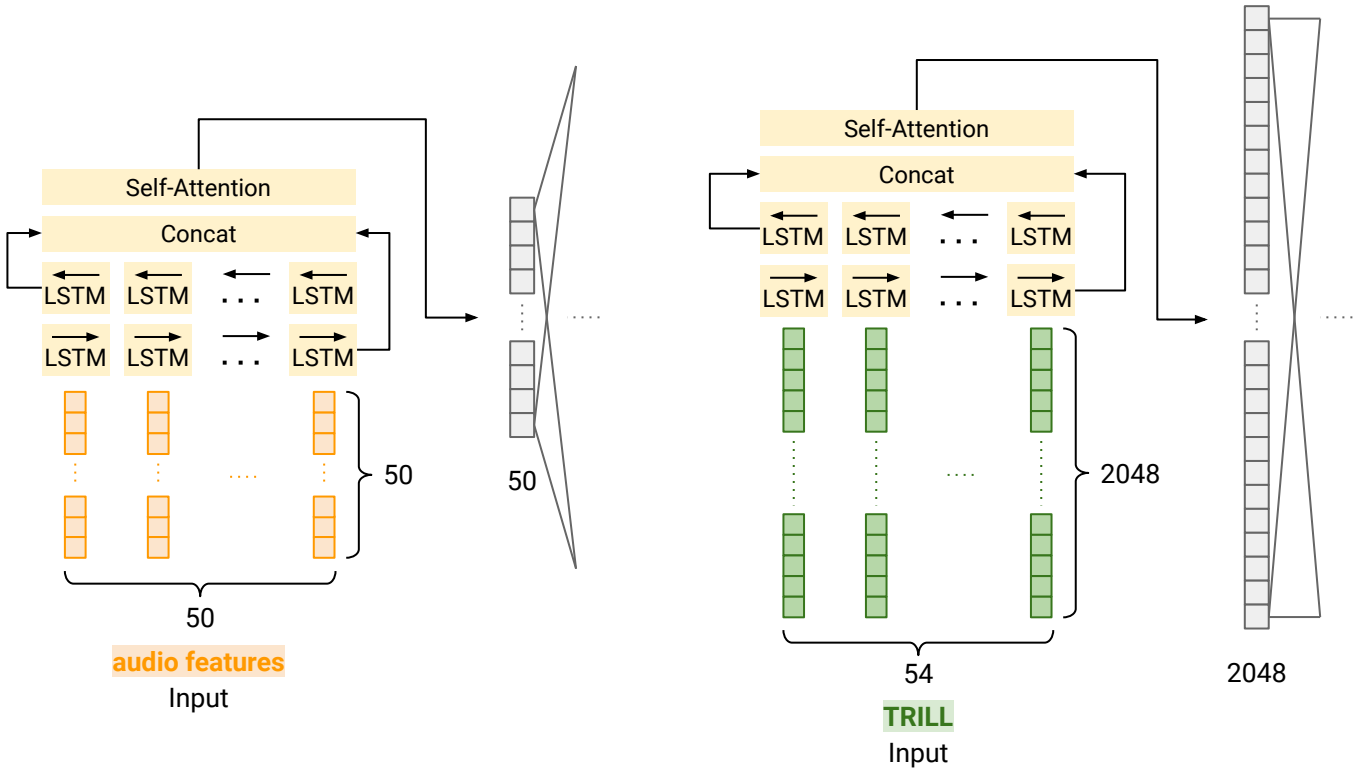
Wan et. al. (2018)
Wan, L., Wang, Q., Papir, A., & Lopez-Moreno, I. (2018). Generalized End-to-End Loss for Speaker Verification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4879-4883.

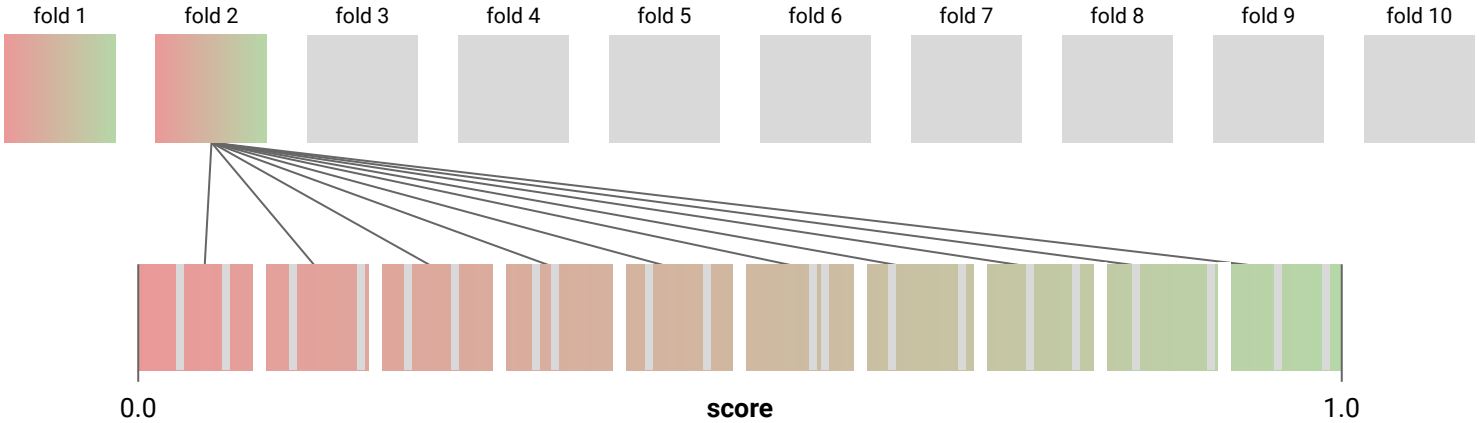
Baumann (2017)
Baumann, T. (2017) Large-Scale Speaker Ranking from Crowdsourced Pairwise Listener Ratings. Proc. Interspeech 2017, 2262-2266, DOI: 10.21437/Interspeech.2017-1697.


Desired predictions for all featured
val set of fold 1/10
val set of fold 2/10
val set of fold 3/10
val set of fold 4/10
val set of fold 5/10
val set of fold 6/10
val set of fold 7/10
val set of fold 8/10
val set of fold 9/10
val set of fold 10/10
true quality +/- normally distributed noise
true quality
0.0
0.2
0.4
0.6
0.8
1.0


Waveform and spectrogram of audio sample
energy
20000
10000
0
−10000
0
20000
40000
60000
80000
100000
120000
140000
160000
sample steps
frequency
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
seconds


audio features
GE2E embeddings
TRILL embeddings
Berlin
aligned.swc
audio1.ogg
audio2.ogg
audiometa.txt
info.json
wiki.html
wiki.txt
wiki.xml
pianistberlin
berlin
1.wav
2.wav
3.wav
4.wav
5.wav
6.wav
7.wav
8.wav
9.wav
10.wav
1.ffv
1.hnr
1.jitter
1.lfbank
1.mfc
1.pitch.esps
1.shimmer
2.ffv
2.hnr
2.jitter
1.pickle
2.pickle
3.pickle
4.pickle
5.pickle
6.pickle
7.pickle
8.pickle
9.pickle
10.pickle


Self-Attention
Concat
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
50
50
50
audio features
Input
Self-Attention
Concat
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
2048
54
TRILL
Input
2048



	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
# speakers	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*20*	*19*
quality mean	*0.48*	*0.48*	*0.485*	*0.48*	*0.478*	*0.478*	*0.485*	*0.487*	*0.483*	*0.451*
quality sd.	*0.299*	*0.298*	*0.302*	*0.303*	*0.301*	*0.295*	*0.3*	*0.303*	*0.294*	*0.277*

Stratified 10-fold cross-validation