IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

Nativeness Assessment for Crowdsourced Speech Collections

Diogo Botelheiro[1,2], *Alberto Abad*[1], *Rui Correia*[2], *João Freitas*[2]

[1]INESC-ID/Instituto Superior Técnico, Universidade de Lisboa, Portugal
[2]DefinedCrowd Corporation

{diogo.botelheiro, correia,joao}@definedcrowd.com, alberto.abad@inesc-id.pt

Abstract

Access to large amounts of annotated data is a challenge for companies developing high-quality AI-based services. Crowdsourcing presents itself as a solution to this growing need for training data, by gathering and distributing work across a large pool of human contributors. This, however, comes at a cost: the difficulty to source the right crowd and maintain data quality. Regarding speech, a critical aspect of data quality relates to the verification of crowd participants as native speakers of a specific language. This work investigates the use of automatic Nativeness Classification (NC) solutions to tackle this problem, integrating a variant-sensitive nativeness classifier component in the speech collection pipeline for Portuguese (European and Brazilian variants) and English (American, British and Indian). By rating individual recordings according to their nativeness, it is possible to both automatically discard substandard work and prevent certain contributors to continue to participate in the collection. Herein, three different speaker-embedding-based frameworks are tested: i-vector, x-vector, and h-vector. Results show that the proposed system based on h-vector outperforms the baseline system with a 8% relative improvement.
Index Terms: nativeness classification, crowdsourcing, deep neural networks, x-vector

1. Introduction

Human-machine collaboration is moving towards more natural means of interaction, in particular via speech. Several speech-based commercial applications exist nowadays, ranging from personal assistants (Siri[1]), dictation systems (Otter.ai[2]), or home automation (Google Assistant[3]). Such technologies are based on high-performance Automatic Speech Recognizers (ASR) [1], which depend on large amounts of training data to improve and to have good performance in new markets (domains/languages). While traditionally such data resources were collected on-site, with experts, the massive amounts of data needed to support these systems deems this collection strategy unfeasible.

To respond to these needs, crowdsourcing emerged as a more scalable (both faster and less costly) approach to speech data collection [2]. Briefly, the crowdsourcing paradigm consists of making available a set of Human-Intelligence Tasks (HITs) to a large pool of contributors, typically via an online-platform. Upon successful completion, a reward or a payment is given to the contributors in an amount proportional to their participation. Given these particularities, crowdsourcing presents a new set of challenges, pertaining to both the loss of control by data requesters and the exploitation attempts by some contributors. More particularly, in the case of speech data collections, requirements such as recording/noise conditions, demographic balancing (age/gender), or nativeness, need to be ensured.

In the crowdsourcing field, these issues are commonly addressed by submitting each generated recording to further human validation [3]. In other words, a common speech data collection pipeline is composed of two steps:

- **Generation Step** - contributors are requested to read and record a given prompt;
- **Validation Step** - contributors are requested to validate certain aspects of a given audio (previously recorded by other contributor), such as the absence of background noise, and/or the speaker's nativeness degree.

However, this validation step increases the price and time to complete the collection. In the circumstances of the amounts of data necessary to train a state-of-the-art ASR model, this is a non-negligible cost since you need to validate thousands of hours of speech. Furthermore, if all the necessary components that need to be validated are included in one single step, this adds to the contributors' cognitive load. As a consequence, it requires higher payment for each of the individual tasks and increases the chances of errors in the validation.

In an effort to reduce the human validation load for the collection of speech data, this work addresses one of the most common validation components: nativeness. The hypothesis is that by integrating an automatic nativeness classifier, it is possible to either remove all the human validation concerning nativeness from the pipeline or provide hints of fraudulent behavior, reducing the amount of data that goes through human validation.

The work presented in this paper was done using real data from DefinedCrowd's[4] prorietary crowdsourcing platform – Neevo[5]. Due to the importance of generalizing language in speech collections, we explore two distinct languages and several variants of those languages, up to a total of five language-locale pairs: Portuguese (European and Brazilian variants) and English (American, British, and Indian variants).

The remainder of this paper is structured as follows: Section 2 sets some terminology and presents background work in the areas of Crowdsourcing and Nativeness Classification; Section 3 describes the experimental setup; Section 4 presents the results; and Section 5 concludes with a discussion of the results and some remarks about future work.

2. Background and state-of the-art

This section further describes the background in which this work was developed on (Section 2.1) and the state-of-the-art of the Nativeness Classification task (Section 2.2).

[1]https://www.apple.com/siri/
[2]https://www.otter.ai
[3]https://assistant.google.com/
[4]https://www.definedcrowd.com/
[5]https://www.neevo.ai/

10.21437/IberSPEECH.2021-5

2.1. Crowdsourcing

The introduction of the term crowdsourcing appears in 2006 by Jeff Howe [4] referring to the increasing practice of outsourcing tasks to the internet as a distributed procedure over various users. Crowdsourcing allows leveraging the so-called *wisdom of the crowds* [5]: the combined knowledge of potentially large groups of individuals. Common tasks approached with crowdsourcing are labeling images, translating or transcribing text, or recording speech, to name a few.

As previously mentioned, the current work was based on Neevo's crowdsourcing platform. From the contributors' point of view, participation is divided into four phases:

- **Registration** - contributors sign up, providing demographic (age/gender) and language data (including reading, writing, and speaking proficiency per language);
- **Work Selection** - depending on their qualifications, contributors see the matching tasks, which are organized into *Jobs*. A Job is a set of tasks (HITs) with a common goal, for instance, "Record yourself reading sentences in European Portuguese", or "Validate the English US recordings in terms of nativeness and noise". When a contributors accepts a Job they are referred to as *Job Members*;
- **Execution** - Job Members read the instructions of the Job and perform the HITs that are still available, usually in the order of the hundreds or thousands. An instance of a submitted HIT is called a *HIT Execution*;
- **Payment** - upon successful completion of the work (which can be dependent on subsequent validation jobs), the contributor is paid accordingly.

In the case of a Speech Collection, as also already mentioned, there are typically two jobs involved: a generation job and a validation one. The generation job can ask for spontaneous speech (where the contributor should talk about a topic for a certain amount of time) or for scripted speech (where the contributor reads a sentence). The validation job typically encompasses all the aspects that need to be validated (which can include text-audio match, background noise, and nativeness). In other words, if *any* of the aspects is not verified, the HIT Execution is canceled and should be re-recorded (not necessarily by the same contributor). Given the sensitivity of this decision, each HIT in the validation job is usually assigned to several distinct Job Members (two or three), in order to analyze for consistency and agreement between their answers.

2.2. Nativeness Classification

Nativeness classification is a subject that has been investigated for the past twenty five years. It is well-known that the presence of non-native speakers in the training set pose problems for speech recognition models, typically degrading their performance [6]. Recent literature on binary classification of nativeness uses distinct techniques. In studies like Shriberg et al. [7], the authors adress NC by applying effective speaker recognition methods based on Maximum Likelihood Linear Regression (MLLR), prosodic information, phone N-gram, and word N-gram features. Combining the different systems allowed to achieve an reasonable Equal Error Rate (EER) for detecting American English non-native speakers. Lopes et al. [8] developed a nativeness classifier using TED talks. A combination of acoustic and prosodic cues led to a good performance. In Mehrabani et al. [9], another implementation based on prosodic features, the authors were able to exceed the baseline accuracy of a Gaussian Supervector by over 10.0%. Another approach from Ribeiro et al. [10] developed several feature sets, including i-vectors, phonotactic models and n-grams counts based features. The results were superior from the presented baseline, with 44% improvements compared to results obtained by Honig et al [11]. Also introduced by Rajpal et al. [12] the use of longer duration cepstral features, namely Mel Frequency Cepstral Coefficients (MFCC) and auditory filterbank features learned from the database using Convolutional Restricted Boltzmann Machine (ConvRBM), allowed for accuracy improvements in the order of 40%.

3. Experimental Setup

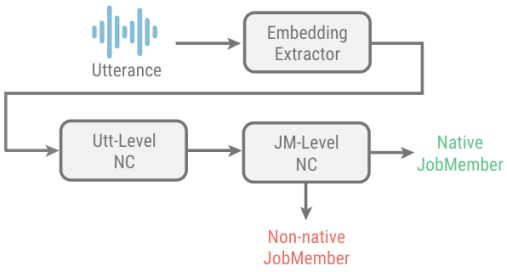

Figure 1: *Proposed Architecture.*

Figure 1 presents the architecture of the solution proposed to accomplish the goal of integrating nativeness assessment into a crowdsourcing speech data collection pipeline. The system is divided into three main components:

- **Embedding Extractor** - transforms the input speech data into a less-dimensional (vectorial) space;
- **Utterance-Level Nativeness Classification** - assigns a nativeness score to each HIT Execution;
- **Job Member-Level Nativeness Classification** - taking into account all the executions from any given Job Member, aggregates them providing a score on the likelihood of the Job Member being either a native or non-native speaker.

The following subsections will describe each component in detail, and introduce the experiments carried out for each of them. The work developed herein was based on the open-source toolkit Kaldi[6] for the feature extraction, data augmentation, and scoring, and TensorFlow[7], for neural network training.

3.1. Embedding Extractor

Advancements in the Automatic Language Identification field also impacted the Nativeness Classification field because of their similarities in intrinsic properties of each language [13, 14]. Nevertheless, with the emergence of Big Data [15], neural networks begun to generate interest. For today's Acoustic Models state-of-the-art, most systems use DNN for the embedding extractor [16], known as x-vector. Currently, to our knowledge, there is no published work with x-vector applications on NC that way, it would be interesting to apply the actual state-of-the-art of NLI [16, 17] to our research. The embedding extractor,

[6]https://github.com/kaldi-asr/kaldi
[7]https://github.com/sun-peach/x-vector-kaldi-tf

responsible for providing a vectorial representation of the input, is the variable component in this work experiments. Four different systems were developed:

- i-vector - previously referred to as state-of-the-art, we use this framework as a method of comparison [13];
- x-vector - TDNN implementation [16];
- h-vector - CNN implementation without attention mechanism [18];
- h-vector + attention - CNN implementation with attention mechanism [18].

The embedding structure follows the work of [16], using ReLU layers for the TDNN and Leaky-ReLU layers for the CNN. Given the specific context in which the solution will be used, and in face of available data, a dedicated corpus was built to train all four embedding systems (despite the various open-source corpus for tasks like spoken language recognition and native classification [19, 20, 21]).

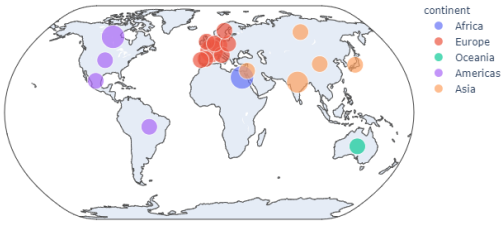

Figure 2: *Global representation for the embedding extractor corpus.*

To this end, validated data from different speech collections at DefinedCrowd was gathered, ultimately resulting in a dataset comprised of 25 language-locale pairs, with 132 hours of audio in total (113K utterances) represented in Figure 2. Data was balanced by trimming it to 5K utterances per language-locale pair (with the exception of ta_in, te_in, and zn_cn with only 1K utterances each). Data augmentation techniques were also used for robustness to noisy background, by applying additive noises using the Musan dataset [22]: 900 noises, 42 hours of music from various genres, and 60 hours of speech from twelve languages. This resulted in a final embedding training set of approximately 378K utterances.

In this work, we used 20-dimensional Mel-Frequency Cepstrum Coefficients (MFCC) with a frame-length of 25ms.

3.2. Utterance-Level Nativeness Classification

The second component that comprises the proposed solution is a nativeness classifier at the utterance level. In sum, it receives as input the vectorial representation of audio (embedding) and outputs a likelihood score with respect to its nativeness.

As mentioned in the introductory section, this work addresses five language-locale pairs: European Portuguese (pt-PT), Brazilian Portuguese (pt-BR), American English (en-US), British English (en-GB), and Indian English (en-IN). Consequently five corpora with nativeness information at the utterance level were built.

As in the case of embedding training, the data to train and test this component was extracted from real crowdsourcing data collections in the Neevo platform. Different strategies were applied for correctly labeling positive (native) and negative (non-native) labels. An utterance was considered positive when the job member producing them *a)* claimed (during sign-up) to be native of that language-locale pair, *b)* claimed (during sign-up) to live in the territory corresponding to the pair, and *c)* had **all** utterances approved in the validation job. On the other hand, utterances representing non-native speech were added through manual verification (by the author) based on work that was not accepted in the validation task. This manual process was necessary since, as already said, there are several reasons (other than nativeness) why a recording can be marked as invalid, including containing background noise, stuttering, hesitations, to name a few. Additionally, negative cases of each dataset were enriched with the positive cases of the other variants, i.e., for instance, utterances produced by pt-pt speakers were added to the non-native cases of the pt-BR set. Table 1 provides a description of the size of each dataset (after splitting into train, development and test).

Table 1: *Division between train, dev and test dataset*

	Train set		**Dev set**		**Test set**	
Model	***#utt***	***#hours***	***#utt***	***#hours***	***#utt***	***#hours***
pt-PT	17300	31	2219	4	1167	2
pt-BR	15838	30	2792	5	1205	2
en-US	57522	112	4019	7	5098	8
en-GB	64495	111	5720	9	5028	8
en-IN	60997	113	5122	8	5790	5

For each target language, we extracted and computed the native and the non-native average vectors. Therefore, the representations are centered and projected using the training set. We start by applying the LDA with a dimension tuned to 200. After dimensionality reduction, the representations are length-normalized and modeled by PLDA, where we get two scores for each utterance. The score for native and non-native are normalized using adaptive s-norm [23]. Next, we performed a ratio between the scores, subtracting the native scores from the non-native to get the final score in Figure 3.

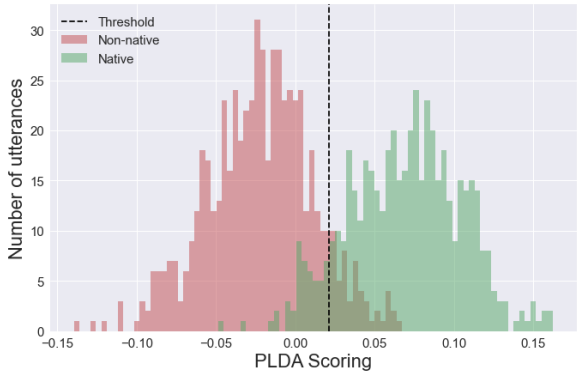

Figure 3: *Example of the result obtained for the h-vector framework for the European Portuguese language set.*

The optimal decision threshold corresponds to one which minimizes the Equal Error Rate (EER) on the dev set. The new utterances are processed by the scoring system and compared to the decision threshold. If the score is greater than the threshold, the utterance is classified as native and non-native otherwise.

Table 2: *Utterance-level classification results.*

	pt-pt		pt-br		en-us		en-gb		en-in	
System	**EER**	**F1-Score**	**EER**	**F1-Score**	**EER**	**F1-Score**	**EER**	**F1-Score**	**EER**	**F1-Score**
i-vector	0.15	0.76	0.32	0.69	0.16	0.83	0.18	0.81	0.16	0.84
x-vector (TDNN)	0.09	0.91	0.19	0.82	0.11	0.89	0.13	0.86	0.11	0.89
h-vector (CNN)	0.07	0.90	0.14	0.87	0.12	0.87	0.10	0.90	0.10	0.89
h-vector (CNN + attention)	0.02	0.96	0.11	0.90	0.10	0.89	0.09	0.91	0.09	0.90

3.3. JobMember-Level Nativeness Classification

Having a nativeness decision per utterance, the next step is to aggregate such information at Job Member level. A Job Member was considered to be a native speaker if the following two conditions were verified:

- Proportion of the Job Member's utterances considered to be native is above 50%;
- Average vector of all the Job Member's utterances is classified as native when using the designated language threshold.

If both conditions failed to be verified, the Job Member was classified as a non-native speaker. For the cases where only one condition was met, the job member was classified as 'ambiguous', potentially needing human verification.

4. Results

To better understand the performance of each component, results will be analyzed separately for utterance and Job Member levels.

Table 2 reports the results obtained from Utterance-Level NC. The first row shows the results from the NC state-of-the-art (i-vector framework), serving as a means of comparison. Results show that the x-vector framework outperforms the baseline in all languages under study, performing the best for the European Portuguese scenario (with an F1 of 0.91). However, its performance is still lacking for the use case in consideration, with 0.19 EER for the pt-BR set. One of the factors that may have hindered the performance was the lack of fine-tuning of some model parameters, such as the regularization coefficient. While the results of the h-vector framework alone are inconclusive (performing better than the x-vectors for some languages, and worse for others), the same is not true for its counterpart with attention mechanism. The h-vector formulation with self-attention mechanism surpassed both the UBM-GMM (i-vector, the state-of-the-art) and TDNN (x-vector) models, achieving a maximum F1 of 0.96 for the pt-PT set. It is also important to highlight the performance improvements achieved for the pt-BR data (0.32 vs. 0.11 EER and 0.69 vs. 0.90 F1).

To conclude on the performance of the Job Member-Level NC component, a system simulating the arrival of new utterances was set up. Job Member nativeness decisions were computed for different amounts of utterances, ranging from 1 utterance per Job Member, until the maximum available. The selection of which utterances to include in each step was random. The simulation was ran 1,000 times, and results per number of utterances were averaged. Figure 4 shows the results achieved. As in the utterance level component, the best results were observed for the pt-PT set, needing only three utterances to successfully classify all Job Members. American and British English achieved the next best performance, requiring eight utterances for accurate classification. Also as in the case of utterance classification, the worst performance was achieved for the Brazilian Portuguese data (eleven utterances needed). The results in this section take into account the test set comprising 45 Job Members for pt-PT, 59 for pt-BR, 121 for en-US, 100 for en-GB, and 96 for en-IN.

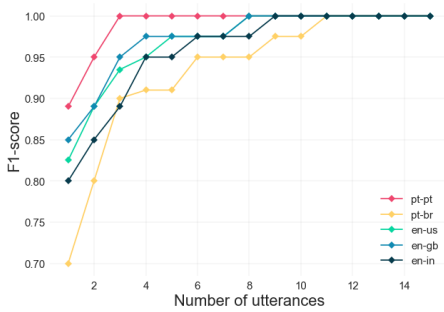

Figure 4: *Example of the simulation experiment for the h-vector (CNN + attention) framework.*

5. Conclusions

In this paper, we presented our work towards the deployment of a nativeness detection module into a crowdsourcing platform for quality control of speech collections. A corpus with more than 130 hours of audio with 25 languages was built to train four different embedding frameworks. We reported performance of Nativeness Classification on five different language-locale pairs along these four frameworks (utterance and Job Member-level). The h-vector solution with attention mechanism outperformed all approaches (including the i-vector baseline). Results show that the number of utterances needed to successfully verify Job Member nativeness vary across language, although having a ceiling of eleven utterances. In the field of crowdsourcing, this information can be used to optimize the speech collection pipeline, preventing substandard work from an early point, thus saving on both time and cost of the collection.

Future work includes the extension of the framework to a larger set of target languages and a deeper investigation of attention based extractor methods.

6. Acknowledgements

This work has been partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020.

7. References

[1] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech communication*, vol. 9, no. 4, pp. 351–356, 1990.

[2] J. Freitas, J. Ribeiro, D. Baldewijns, S. Oliveira, and D. Braga,

"Machine learning powered data platform for high-quality speech and nlp workflows." in *INTERSPEECH*, 2018, pp. 1962–1963.

[3] V. Muntés-Mulero, P. Paladini, J. Manzoor, A. Gritti, J.-L. Larriba-Pey, and F. Mijnhardt, "Crowdsourcing for industrial problems," in *International Workshop on Citizen in Sensor Networks*. Springer, 2012, pp. 6–18.

[4] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[5] J. Surowiecki, *The wisdom of crowds*. Anchor, 2005.

[6] L. Kilman, A. Zekveld, M. Hällgren, and J. Rönnberg, "The influence of non-native language proficiency on speech perception performance," *Frontiers in Psychology*, vol. 5, p. 651, 2014.

[7] E. Shriberg, L. Ferrer, S. S. Kajarekar, N. Scheffer, A. Stolcke, and M. Akbacak, "Detecting nonnative speech using speaker recognition approaches." in *Odyssey*, 2008, p. 26.

[8] J. Lopes, I. Trancoso, and A. Abad, "A nativeness classifier for TED talks," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5672–5675.

[9] M. Mehrabani, J. Tepperman, and E. Nava, "Nativeness classification with suprasegmental features on the accent group level," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[10] E. Ribeiro, J. Ferreira, J. Olcoz, A. Abad, H. Moniz, F. Batista, and I. Trancoso, "Combining multiple approaches to predict the degree of nativeness," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] F. Hönig, A. Batliner, and E. Nöth, "Automatic assessment of non-native prosody–annotation, modelling and evaluation," in *International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, 2012, pp. 21–30.

[12] A. Rajpal, T. B. Patel, H. B. Sailor, M. C. Madhavi, H. A. Patil, and H. Fujisaki, "Native language identification using spectral and source-based features." in *INTERSPEECH*, 2016, pp. 2383–2387.

[13] M. Senoussaoui, P. Cardinal, N. Dehak, and A. L. Koerich, "Native language detection using the i-vector framework." in *INTERSPEECH*, 2016, pp. 2398–2402.

[14] A. N. Uddin, M. A. Rahman, M. Islam, M. A. Haque *et al.*, "Native language identification using i-vector," *arXiv preprint arXiv:1811.05540*, 2018.

[15] G. George, M. R. Haas, and A. Pentland, "Big data and management," 2014.

[16] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors." in *Odyssey*, 2018, pp. 105–111.

[17] X. Miao, I. McLoughlin, and Y. Yan, "A new time-frequency attention mechanism for tdnn and cnn-lstm-tdnn, with application to language identification." in *INTERSPEECH*, 2019, pp. 4080–4084.

[18] Y. Shi, Q. Huang, and T. Hain, "H-vectors: Utterance-level speaker embedding using a hierarchical attention model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7579–7583.

[19] J. Valk and T. Alumäe, "Voxlingua107: a dataset for spoken language recognition," *arXiv preprint arXiv:2011.12998*, 2020.

[20] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[22] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," 10 2015.

[23] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1. IEEE, 2005, pp. I–741.