AUDIO WORD SIMILARITY FOR CLUSTERING WITH ZERO RESOURCES BASED ON ITERATIVE HMM CLASSIFICATION

Amélie Royer[(a)]*, Guillaume Gravier*[(b)]*, Vincent Claveau*[(b)]

a – ENS Rennes, IRISA & Inria Rennes, France
b – CNRS, IRISA & Inria Rennes, France

ABSTRACT

Recent work on zero resource word discovery makes intensive use of audio fragment clustering to find repeating speech patterns. In the absence of acoustic models, the clustering step traditionally relies on dynamic time warping (DTW) to compare two samples and thus suffers from the known limitations of this technique. We propose a new sample comparison method, called similarity by iterative classification, that exploits the modeling capacities of hidden Markov models (HMM) with no supervision. The core idea relies on the use of HMMs trained on randomly labeled data and exploits the fact that similar samples are more likely to be classified together by a large number of random classifiers than dissimilar ones. The resulting similarity measure is compared to DTW on two tasks, namely nearest neighbor retrieval and clustering, showing that the generalization capabilities of probabilistic machine learning significantly benefit to audio word comparison and overcome many of the limitations of DTW-based comparison.

Index Terms— zero-resource speech processing, word discovery, audio words clustering, unsupervised learning, acoustic similarity, dynamic time warping

1. INTRODUCTION

Clustering word-like acoustic fragments has proven useful in a number of situations were no annotated resources are available to build models, the so-called 'zero resource' setting. In particular, unsupervised word discovery from acoustic data with zero resources has recently appeared as a new challenge in speech processing. Seminal work on the topic [1] has triggered various approaches, e.g., [2, 3, 4], and led to the recent zero resource speech challenge [5]. This challenge targets the unsupervised discovery of linguistic units from raw speech in an unknown language, with linguistic units being either word-like units or phone-like units. A key ingredient to unsupervised word discovery is clustering of acoustic patterns that are likely to be words. In fact, all approaches in the literature detect potential repeating word-like fragments that are further grouped together to identify meaningful patterns. The clustering step might be explicit [1, 4], or implicit [2].

Clustering word-like acoustic fragments requires a measure of the similarity between two fragments x and y, regardless of the clustering algorithm used. The natural choice with speech signals is obviously the dynamic time warping (DTW) algorithm to account for possible temporal variations. This is for instance the choice made in [1, 2, 3, 4]. But DTW has a number of drawbacks that severely limit its effectiveness. In particular, DTW is very sensitive to spectral variations, as typically found across speakers. The use of posteriorgram representations improves the speaker-dependency of DTW [6], yet pattern comparison remains sensitive to many variations including start and end point detection, spectral variability and significant speech rate variations. On the contrary, probabilistic models, such as hidden Markov models (HMM) and its variants, have proven significantly more robust to these variations but require training data.

In this paper, we propose to implicitly define a similarity between acoustic fragments suited for clustering that takes full advantage of the modeling and generalization capabilities of HMMs, without the need for pre-trained models. The technique is thus perfectly fit for zero resource tasks. The key idea behind this approach is that any supervised classifier naturally produces a partition of the dataspace thus providing a rough notion of similarity. In particular, in recent years, several studies have investigated the use of classifiers trained on randomly generated annotations of the data to uncover similarities between samples [7, 8, 9, 10]. In locality sensitive hashing schemes, samples often falling on the same sides of random hyperplanes are grouped together. Similarly, samples grouped in the same class by a set of randomly trained classifiers are deemed very similar. The advantage over LSH lies in the generalization capability of classifiers, which leads to much more complex space partitions than hyperplanes. We apply here this principle, named *similarity by iterative classification* (SIC), to audio similarity, using HMMs as classifiers to group word-like audio fragments.

[*] Now with Institute of Science and Technology Austria

2. AUDIO SIMILARITY BY ITERATIVE CLASSIFICATION

The key idea for computing the audio similarity between two signals by iterative classification is that, if we consider a set of independent classifiers, the more often two samples are assigned the same label by one of the classifiers, the more likely it is that the two samples are similar. The principle of SIC is thus to generate a significant number of independent classifiers and to count how often two samples are classified together among the set of independent classifiers.

In fact, the reason why a classifier labels two samples similarly is first and foremost because the two samples exhibit structural similarity as modeled by the classifier. Obviously, the type of classifier used must be adapted to the task and able to capture the structural properties of the data. The very principle of SIC was first introduced in [7] and [8] with a similarity based on respectively decision trees and random forests to distinguish synthetic samples from true data. A first extension to time-structured data clustering using conditional random fields was proposed in [9]. This last approach is here adapted to speech signals clustering relying on hidden Markov models, a natural choice for the classification of speech signals, where Markov models are trained directly on the data to be clustered without the need for human-labeled data.

2.1. The SIC algorithm

Let $\mathcal{X} = \{x_1 \dots x_D\}$ be a database of D audio samples. We aim at defining a similarity function $s : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ between pairs of samples taken from $\mathcal{X}$. Following the SIC principle, we need to train a set of independent HMM classifiers on the samples, each classifier providing a different partition of the data space. This is achieved by randomly choosing a subset of the data as training set, on which labels are randomly generated. HMM classifiers are then learned from this synthetic (random) training set and applied on the remainder of the data to generate labels that will further serve as the basis for defining a similarity between any two pair of samples.

Formally, the following process is iterated N times to generate a number of independent HMM classifiers so as to prevent bias towards specific training parameters.

At iteration i, a training set and a test set, Tr_i and Te_i, are extracted from the database such that $Tr_i \cap Te_i = \varnothing$ and $Tr_i \cup Te_i \subset \mathcal{X}$. A synthetic label is randomly generated for each training sample in Tr_i. We denote α_r (resp. α_e) the proportion of training (resp. test) samples, and L the number of unique labels which are randomly assigned to the samples in Tr. Note that prior knowledge could be included during the labeling step in order to refine the process, e.g., if two samples are already known to be very similar, they should be assigned the same synthetic label. In our experiments, such knowledge is however not available, and we only resort to a basic randomized assignation. Based on the randomly generated labels, a classifier c_i is trained on Tr_i and used to classify each sample in Te_i. The classification result defines a similarity score $s_i : Te_i \times Te_i \rightarrow \mathbb{R}$, reflecting the assumption that two samples obtaining the same label share some structural similarity uncovered by the classifier. Formally, we define $s_i(x, y)$ as

$$s_i(x, y) = \mathbb{1}_{c_i(x)=c_i(y)} = \begin{cases} 1 & \text{if } c_i(x) = c_i(y) \\ 0 & \text{if } c_i(x) \neq c_i(y) \end{cases} . \quad (1)$$

After the last iteration, the similarity between two data points x_p and x_q from $\mathcal{X}$ is obtained as the average number of times the two samples have been classified together over the N iterations, i.e.,

$$s^N(x_p, x_q) = \frac{\sum_{i=1}^{N} s_i(x_p, x_q)\ \mathbb{1}_{\{x_p, x_q\ \in\ Te_i\}}}{\sum_{i=1}^{N} \mathbb{1}_{\{x_p, x_q\ \in\ Te_i\}}} \quad (2)$$

where $\sum_{i=1}^{N} \mathbb{1}_{\{x_p, x_q\ \in\ Te_i\}}$ is the number of times x_p and x_q were both in the same test set.

The pseudo-code of the algorithm is given in Algorithm 1. Note that other score functions than (1) could be considered, e.g., using reward/penalty scores instead of a binary decision. We experimented several such variants [11]. However, while they change the overall distribution of the similarities, none of the variants impact the clustering results significantly.

Data: $\mathcal{X}$, α_r, L_{min}, L_{max}
Result: $s : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$
for *i=1 to N* **do**
  $Tr_i \leftarrow \alpha_r|\mathcal{X}|$ random samples from $\mathcal{X}$;
  $Te_i \leftarrow \alpha_e|\mathcal{X}|$ random samples from $\mathcal{X} \setminus Tr_i$;
  $L_i \leftarrow \text{rand}(L_{\text{min}}, L_{\text{max}})$;
  foreach $x_j \in Tr_i$ **do**
    Assign label l_j, where $j = \text{rand}(1, L_i)$;
  $c_i \leftarrow \text{Learn}(Tr_i, (l_j)_j)$; `// Training`
  for $\{x_p, x_q\} \in Te_i \times Te_i$ **do**
    $s(x_p, x_q)$ += $\mathbb{1}_{c_i(x_p)=c_i(x_q)}$;
    $occ(x_p, x_q)$ += 1;
$s \leftarrow s/occ$; `// Normalization`

Algorithm 1: Pseudo-code for SIC

2.2. About randomization

The randomization of the learning parameters at each iteration is an essential part of the algorithm to avoid bias towards specific characteristics of the data in the final similarity. Regarding the training and test sets, we keep their proportions constant throughout the iterations and only vary their composition. The value of α_r is determined according to the size

of the dataset and to the maximum number of synthetic labels we consider, so as to ensure enough samples in each training class on average. As for the test samples, we simply use the remaining samples, i.e., $\alpha_e = 1 - \alpha_r$ and hence $Te = \mathcal{X} \setminus Tr$.

The number of synthetic labels at iteration i, L_i, is chosen at random within an interval $[L_{\min}, L_{\max}]$. In practice, the value of L_i is positively correlated with the granularity and discriminative power of the similarity, as raising the number of labels increases the classification grain. Clearly, having a large number of labels will make it unlikely that two samples be classified together unless they are highly similar. As a consequence, the similarity will be significantly greater than zero only for samples that are indeed very close one from another. All other distances will tend towards zero.

Randomization should also be considered in the classifier setting. We use hidden Markov models for which we vary the topology at each iteration, alternating between two types of chains. `HMM type 1` designates a linear Markov chain with loop transitions on each emitting state and direct forward transitions while `HMM type 2` additionally features skip transitions. In addition to changing the topology, the number of states and the acoustic features are also chosen randomly. See Sec. 3.1 for details on the impact of these parameters.

3. EXPERIMENTS

Similarity by iterative classification is evaluated within two tasks, namely nearest neighbor retrieval and clustering applied to audio words.

3.1. Experimental Setting

Word-like audio samples were extracted from a subset of the ESTER2 dataset [12], which contains audio streams extracted from various French radio news shows. We considered all words that can be extracted from the reference transcript, filtering out potential outliers. We excluded samples with a length inferior to 0.2 seconds, as well as all words with less than 10 occurrences. As clustering is purely acoustic, possible homophones were merged in a single category. The resulting database contains 13,477 audio samples for 543 unique clusters. The main difficulty of the task lies in the high variability of speaker and recording conditions (radio studio, outdoors, phone conversation...) among the samples of a given class. The fact that words were taken from broadcast news speech and extracted from their context also adds to the difficulty because of context removal and coarticulation.

We use Mel frequency cepstral coefficients as a classical representation of speech signals. For DTW, we use MFCC with first and second order derivatives and perform cepstral mean removal. On the contrary, following the same randomization process as earlier, we vary the type of coefficients extracted for the MFCC features at each iteration of SIC. Note that using variance normalization or more robust features would certainly slightly improve the results, however both for SIC and for the baseline. We thus chose to experiment with difficult features to show the robustness of SIC in adverse conditions. We also set the main parameters values as $\alpha_r = 0.4$, $L_{\min} = 100$ and $L_{\max} = 200$, which on average ensures roughly 40 samples in each synthetic class. While a higher number of synthetic labels could better capture the high granularity of the ground-truth clustering (543 classes), it would lead to higher computation times and memory usage.

SIC is compared to a standard DTW similarity in terms of nearest neighbor retrieval, where the neighbors of a sample are the members of its ground-truth class, and in terms of clustering, relying on Markov clustering [13]. Markov clustering operates on a graph connecting all points with edges weighted by the similarity between the samples, obtained either with SIC or with DTW. The interest of the nearest neighbor retrieval evaluation is that it avoids the dependency to a particular clustering algorithm and parameter setting. These results should thus be considered as a more objective assessment of the similarity performance than the one obtained via clustering. Yet, the clustering results shows the benefits in a more realistic task.

For all experiments, we report the mean average precision (mAP), average f-score at rank 1 and 100 for the nearest neighbor retrieval task. The mAP evaluates the precision (relatively to the ranks of the ground-truth neighbors in the list of neighbors ranked according to the similarity measure considered), while the f-measure captures both recall and precision. For the clustering task, we report standard evaluation measures comparing the resulting clusters with ground-truth classes, that is: adjusted rand index, V-measure, normalized mutual information and adjusted purity. The adjusted purity characterizes how pure the clusters are based on the number of different classes appearing in a cluster. The adjusted Rand index uses pairs counting and takes into account both correctly and incorrectly classified pairs of samples. Finally, the V-measure and normalized mutual information are both based on entropy and information theory notions, rather than pairs counting. A complete presentation and discussion of these scores can be found in [14].

3.2. Results

We first present in Tab. 1 a comparison of various SIC runs (2,000 iterations each) with different HMM topologies. Label "type 1/2" denotes runs where one of the two topologies is chosen at random at each iteration. We also indicate the total number of states in the HMM, which is usually constant when type 2 is present, as the skip transitions allow for various lengths of Markov chains. Finally, given that the minimum length of the samples is 0.2s and the feature rate is 100Hz, the maximum number of states in the HMMs is set to 20. Tab. 1 shows that increasing the number of states in the HMM improves the results for all the evaluation measures.

Setting / Measure	Type 1/2 7 states	Type 1/2 10 st.	Type 1/2 12 st.	Type 1/2 14 st.	Type 1/2 20 st.	Type 1 random(7;20)	Type 2 20 st.
mAP	16.49	18.27	19.86	20.61	**20.63**	20.53	20.20
f@1	57.66	59.47	61.79	**62.86**	62.46	62.64	62.44
f@100	14.56	15.89	16.82	**17.28**	17.19	17.13	17.02
Adj. Rand Index	0.130	0.149	0.133	0.136	0.135	0.107	**0.153**
V-measure	0.597	0.612	0.616	0.619	**0.623**	0.619	0.621
Norm. Mutual Info	0.585	0.598	0.601	0.604	**0.608**	0.604	**0.608**
Adj. Purity	0.476	0.524	0.552	**0.556**	**0.556**	0.543	0.539

Table 1. Influence of the randomization on the HMM topology. Bold entries indicate the best result for each evaluation metric.

Similarity / Evaluation	DTW	SIC
mAP	3.11	20.61
f@1	14.05	62.86
f@100	4.65	17.28
Adjusted Rand Index	0.003	0.135
V-measure	0.177	0.623
Normalized Mutual Info	0.154	0.608
Adjusted Purity	0.117	0.556
Clusters found	542	542

Table 2. Clustering and retrieval results comparison of the DTW and SIC similarity on the ESTER2 dataset

However the topology of the HMM itself has no significant influence as the results are roughly the same for type 1, type 2 or type1/2 runs with 14-20 states.

Results comparing SIC and DTW are reported in Tab. 2, where SIC was estimated over 2,000 iterations with type 1/2 HMMs having 14 states. We observe that SIC clearly outperforms DTW for the different evaluation metrics. The advantage of SIC over DTW is clearly due to the fact that the exploitation of adequate classifiers, even if trained with artificially generated labels, allows us to build a similarity measure with a more complex internal representation of the data, thus better capturing the resemblance existing between the samples. Detrimental to DTW is also the scaling of scores between distinct pairs of samples. On the contrary, SIC does not face score calibration issues. An in-depth analysis of the results shows that both similarities perform better on rare words, e.g., person's names. A possible explanation to this observation lies in the burstiness phenomenon: Rare words tend to appear in specific contexts, while common words are more likely to occur in more various contexts and hence exhibit greater variability among their samples. While this phenomenon benefits to both SIC and DTW, SIC improvement over DTW is overall much higher for these rare words, leading us to believe that SIC benefits more than DTW from situations with limited variability between occurrences of the sample from the same class.

In terms of computational cost, the main bottleneck for SIC are the numerous training and testing phases with the different classifiers. For our experiments, we developed a parallel implementation of SIC and ran it on several 8 cores and 48GB RAM nodes. As an order of magnitude, running 50 iterations of SIC on one single node required on average 9.53GB RAM, 1h12 of actual elapsed time (*real*), and 4h48 of CPU time cumulated over all processors (*sys+usr*). This also raises the question of the convergence speed of the algorithm. While the results reported here were obtained for 2,000 iterations to ensure convergence, we observed in practice that the similarity usually reaches a stable point around 1,000 iterations. In [11], we present a more complete analysis of the convergence speed and also propose an on-the-fly stopping criterion for the SIC algorithm, based on the evolution of the average entropy of the similarity throughout the iterations.

4. CONCLUSION

This paper shows that efficient supervised classification algorithms can be diverted to define a similarity between audio samples. The similarity is said to be implicit as no direct comparison of the samples is explicitly made, as opposed to what is done with DTW. Results on audio word comparison demonstrates that the modeling and generalization capabilities of supervised models yield significantly better measures of similarity than direct pattern comparison. Clearly, full advantage can be taken of advanced hidden Markov modeling—vocal tract length normalization, speaker adaptation, DNN posteriors, etc.—to yield more accurate similarity. The bottleneck remains the computation time but the process is highly parallel and, as we demonstrated, would benefit from a massively parallel GPU implementation. A key point is that the method is applicable to any audio data and reaches beyond retrieval and clustering. HMMs are here trained on the dataset to cluster but could as well be trained beforehand on unlabeled data. This opens many opportunities, e.g., in template-based speech recognition or in low-resource languages where annotated data are scarce.

5. REFERENCES

[1] Alex Park and James R. Glass, "Unsupervised word acquisition from speech using pattern discovery," in *IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2006.

[2] Armando Muscariello, Guillaume Gravier, and Frédéric Bimbot, "Audio keyword extraction by unsupervised word discovery," in *Annual Conf. of the Intl. Speech Communication Association*, 2009.

[3] Aren Jansen, Kenneth Church, and Hynek Hermansky, "Towards spoken term discovery at scale with zero resources," in *Annual Conf. of the Intl. Speech Communication Association*, 2010.

[4] Rémi Flamary, Xavier Anguera, and Nuria Oliver, "Spoken wordcloud: Clustering recurrent patterns in speech," in *Intl. Workshop on Content-Based Multimedia Indexing*, 2011.

[5] Marteen Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The zero resource speech challenge 2015," in *Annual Conf. of the Intl. Speech Communication Association*, 2015.

[6] Yaodong Zhang, R. Salakhutdinov, Hung-An Chang, and J. Glass, "Resource configurable spoken query detection using deep Boltzmann machines," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2012.

[7] Bing Liu, Yiyuan Xia, and Philip S Yu, "Clustering through decision tree construction," in *Intl. Conf. on Information and Knowledge Management*, 2000.

[8] Tao Shi and Steve Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, 2006.

[9] Vincent Claveau and Abir Ncibi, "Knowledge discovery with CRF-based clustering of named entities without a priori classes," *Conference on Intelligent Text Processing and Computational Linguistics CICLing*, pp. 417–433, 2014.

[10] Alexis Joly and Olivier Buisson, "Random maximum margin hashing," in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2011.

[11] Amélie Royer, Vincent Claveau, Guillaume Gravier, and Teddy Furon, "Knowledge discovery in multimedia content by diversion of supervised learning techniques," Tech. Rep., Inria, 2015.

[12] Sylvain Galliano, Édouard Geoffrois, Guillaume Gravier, Jean-François Bonastre, Djamel Mostefa, and Khalid Choukri, "Corpus description of the ESTER evaluation campaign for the rich transcription of French broadcast news," in *Language Resources and Evaluation Conf.*, 2006, pp. 315–320.

[13] Stijn van Dongen, "A cluster algorithm for graphs," Tech. Rep., Centre for Mathematics and Computer Science, Amsterdam, 2000.

[14] Nguyen Xuan Vinh, Julien Epps, and James Bailey, "Information theoretic measures for clusterings comparison," *Journal of Machine Learning Research*, 2010.