*INTERSPEECH 2017*
August 20–24, 2017, Stockholm, Sweden

# Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization

*Guillaume Wisniewksi, Hervé Bredin, Grégory Gelly, Claude Barras*

LIMSI, CNRS, Univ. Paris-Sud, Université Paris Saclay, 91 403 Orsay, France
firstname.lastname@limsi.fr

## Abstract

Real-time speaker diarization has many potential applications, including public security, biometrics or forensics. It can also significantly speed up the indexing of increasingly large multimedia archives. In this paper, we address the issue of low-latency speaker diarization that consists in continuously detecting new or reoccurring speakers within an audio stream, and determining when each speaker is active with a low latency (*e.g.* every second). This is in contrast with most existing approaches in speaker diarization that rely on multiple passes over the complete audio recording. The proposed approach combines speaker turn neural embeddings with an incremental structure prediction approach inspired by state-of-the-art Natural Language Processing models for Part-of-Speech tagging and dependency parsing. It can therefore leverage both information describing the utterance and the inherent temporal structure of interactions between speakers to learn, in supervised framework, to identify speakers. Experiments on the Etape broadcast news benchmark validate the approach.

## 1. Introduction

In this paper, we address the issue of online speaker diarization that consists in continuously detecting new or reoccurring speakers within an audio stream, and determining when each speaker is active in an online fashion. This differs from most state-of-the-art speaker diarization systems that follow a batch process, performing multiple agglomerative clustering passes over the complete audio recording [1].

More precisely, we focus on very low latency (*e.g.* one second) online speaker diarization. This issue has been addressed before (see [2] for a survey), but usually with higher or variable latency. *Liu and Kubala* [3] compared online speaker clustering with a hierarchical batch clustering but relied on a (reference) speaker turn segmentation, delaying the decision to the end of each turn. *Oku et al.* [4] looked for speaker changes at phoneme boundaries and performed speaker clustering with a 2 to 20 seconds latency, resulting in better speaker adapted models for speech transcription.

*Markov and Nakamura* designed an online speaker diarization system based on Gaussian Mixture Models (GMM) and observed a doubling of the error rate when the latency is reduced from 5 seconds to only one second [5]. This is likely due to the fact that GMMs are not well suited for modeling very short speech segments. Our first contribution is therefore to replace (multi-)gaussian modeling by neural embeddings that we recently found to significantly outperform state-of-the-art approaches for short speech segments comparison [6]. Section 2 describes how we extend our previous work with a new multi-level architecture and a better training procedure.

Our second contribution is to consider diarization as an incremental sequence labeling task, a popular framework in the Natural Language Processing (NLP) community that reduces structure prediction to a sequence of local *actions* predicted by a multi-class classifier that incrementally builds the output structure. This approach is at the heart of state-of-the-art NLP models for many sequence labeling tasks [7, 8] and of many parsers [9, 10, 11]. As discussed in Section 3, it has both advantages of being particularly adapted to online speaker diarization, and allowing to model the inherent temporal structure of interactions between speakers. This follows our previous work [12] on supervised speaker identification in TV series. There, we showed that structured prediction techniques (taking the sequence of speech turns into account) outperform standard approaches processing speech turns independently from each other. The main difference over [12] is that we switch from supervised speaker identification to unsupervised speaker diarization, removing the need for prior biometric models.

## 2. Neural embedding

In this section, we improve over our previous work [6] on neural embedding of short speech segment using Long Short-Term Memory network (LSTM). This type of approaches aims at projecting speech sequences into a high-dimensional space where sequences from one speaker (respectively two different speakers) are close to (resp. far from) each other. We propose a novel network architecture and the associated *"angular proximity"* loss that, when combined, facilitate training and improve performance. More details about this approach can be found in [13], where it is used for spoken language identification.

### 2.1. Network architecture

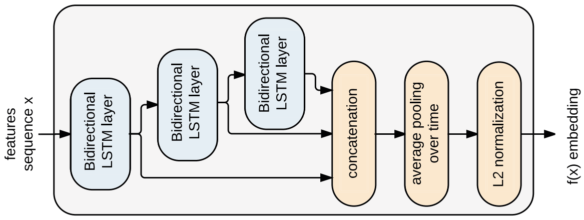

Figure 1: *Multi-level network for sequence embedding.*

As depicted in Figure 1, the proposed network is composed of three stacked bi-directional LSTM layers whose output sequences are concatenated to obtain an output combining multiple levels of abstraction. This architecture also facilitates the training by linking all the LSTM layers directly to the output. The output sequence is averaged over time, giving a unique output vector for the whole sequence. A final L2-normalization

http://dx.doi.org/10.21437/Interspeech.2017-1067

layer ensures the output embedding lives on the unit hypersphere.

### 2.2. *"Angular proximity"* loss

Because embeddings are forced to live on the unit hypersphere, the information resides in the direction of the embedding, not the norm. Therefore, inspired by the triplet [14] and the center [15] losses, we designed the *"angular proximity"* loss that jointly:

- learns a unique representative embedding $\boldsymbol{r}_i$ for every speaker $i \in [1, n]$ in the training set,
- minimizes the angular offset $\theta_i(\boldsymbol{z}) = \arccos(\boldsymbol{r}_i \cdot \boldsymbol{z})$ between embeddings $z$ (uttered by speaker $i$) and their representative embedding $\boldsymbol{r}_i$,
- maximizes the angular offset with the representative embeddings of all other speakers.

It is defined as follows:

$$\mathcal{L}(\boldsymbol{z}, s) = \sum_{i \in [1,n]}^{i \neq s} \sigma(\theta_s(\boldsymbol{z}) - \theta_i(\boldsymbol{z})) \qquad (1)$$

where $\boldsymbol{z}$ is the embedding of a sequence uttered by speaker $s$, and $\sigma$ is the logistic function which brings faster and better convergence by focusing the training efforts on the cases that are close to the boundaries between speakers.

In practice, representative embeddings $\boldsymbol{r}_i$ are initialized randomly, and trained together with the parameters of the network using backpropagation and SMORMS3 gradient descent algorithm [16]. Though representative embeddings can serve as speaker models for later supervised speaker identification, they are not used in the rest of the paper: the diarization approach is fully unsupervised in terms of prior biometric models.

## 3. Incremental structure prediction

Considering speaker diarization as an incremental sequence labeling task has several advantages. First, it allows us to identify speakers continuously with a low latency as decisions are taken *"on the fly"*, as soon as a new audio segment has been observed. Second, by using a discriminative classifier, it is possible to consider arbitrary features describing both the speech utterance and the inherent temporal structure of interactions between speakers. Finally, this approach formulates diarization in a supervised learning framework. While we do not consider the actual speaker identities (labels are arbitrary integers that can only be used to decide if two utterances have been pronounced by the same person), the proposed method is trained from annotated audio streams.

### 3.1. Inference

Algorithm 1 and Figure 2 describe how diarization can be formulated as an incremental sequence labeling task. The audio stream is segmented into segments of fixed length (1 second in our experiments). After observing a new segment, a multi-class classifier is used to choose one among $n$ classes (or actions) to attribute this utterance to:

- one of the $n-1$ speakers that have been previously detected ($n-1$ ADD actions);
- a new speaker that will be referred to as the $n$-th speaker (NEW action);

The classifier can use features describing both the past observations (*e.g.* the similarity between current and past neural embeddings) and the past decisions (or history, *e.g.* the number of speakers identified so far). Note that the number of possible actions increases as new speakers are detected.

**Algorithm 1** Inference of our low-latency diarization system.

```
$h$ = EMPTYLIST();
while The audio stream is not empty do
    $\mathbf{x}$ = NEXTOBSERVATION()
    $\mathcal{A}$ = GETACTIONS($h$)
    $a^*$ = PREDICTMULTICLASS($\mathcal{A}, \mathbf{x}, \mathbf{w}, h$)
    $h$.PUSH($(\mathbf{x}, a^*)$)
end while
```

Choosing an action results in an update of the history $h$, that stores the sequence of actions made so far. In particular, this history allows to determine the number of speakers that has already been detected and the utterances pronounced by them: this is used to determine the set of possible actions and the features describing them.

In practice, a standard multi-class SVM is trained to choose one of the $n$ possible actions $\mathcal{A}$ according to the current history. There is one weight vector $\mathbf{w}_a$ and one feature function $\phi_a(\mathbf{x})$ for each possible action $a$. Given an observation $\mathbf{x}$, the SVM predicts the highest-scoring action:

$$a^* = \arg\max_{a \in \mathcal{A}} \mathbf{w}_a \cdot \phi_a(\mathbf{x}) \qquad (2)$$

To take into account the fact that the number of actions is increasing as new speakers are detected, we assume that all ADD actions share the same parameters: there are in fact only two weight vectors in our SVM, one for computing the score of all ADD actions and one for computing the score of the NEW action. The feature vectors are obviously adapted to each action.

### 3.2. Training

The main challenge faced by incremental approaches is that the past predictions influence the distribution of the future features violating the crucial assumption that examples are independent and identically distributed. The Algorithm 2 is an *imitation learning* method [7, 17] designed to avoid this problem by integrating learning and inference: the classifier is trained in an online fashion (*i.e.* the weight vector is updated each time an new utterance is observed) on a distribution of a possibly suboptimal sequence of actions obtained by running the system itself. It is closely related to the training algorithm of dependency parsers [18].

**Algorithm 2** Training of a low-latency diarization system

```
$h$ = EMPTYLIST();
while The audio stream is not empty do
    $\mathbf{x}, y$ = NEXTOBSERVATION()
    $\mathcal{A}$ = GETACTIONS($h$)
    $o$ = ORACLE($y, h$)
    $a^*$ = PREDICTMULTICLASS($\mathcal{A}, \mathbf{x}, \mathbf{w}, h$)
    $\mathbf{w}$ = UPDATE($\mathbf{w}, \mathbf{x}, o$)
    $h$.PUSH($(\mathbf{x}, a^*)$)
end while
```

More precisely, at each step, the algorithm computes the oracle action that would associate the received observation to

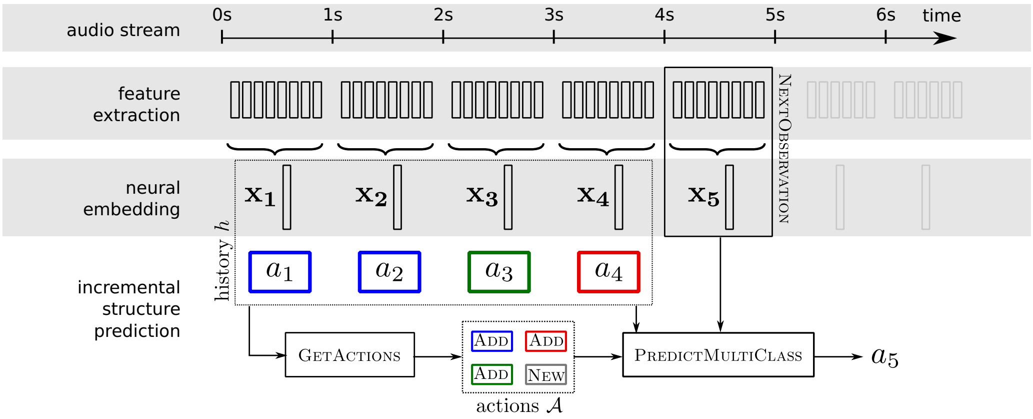

Figure 2: *Incremental sequence labeling formulation of a low-latency speaker diarization task.*

the correct speaker and compare it to the predicted actions. The weight vector of the classifier is then updated if necessary (*i.e.* if the loss of the considered classifier is not null). Importantly, the history is updated with the predicted action and not the gold action so that the training data model the fact that the sequence of past actions will be imperfect at test time.

Detecting whether an error occurs when the SVM makes a prediction for an observation $\mathbf{x}$ with gold action $y$ is done by computing the multi-class hinge-loss [19] defined as:

$$\ell(y, \mathbf{x}, \mathbf{w}) = \max\left(0, 1 + \phi_r(\mathbf{x}) \cdot \mathbf{w}_r - \phi_y(\mathbf{x}) \cdot \mathbf{w}_y\right) \quad (3)$$

where $r = \arg\max_{a \in \mathcal{A} \setminus \{y\}} \mathbf{w}_a \cdot \phi_a(\mathbf{x})$ is the erroneous action with the highest score. This loss is null only when the correct action has been predicted with a large enough confidence. After receiving a new observation, the weight vector of the SVM is updated as follows:

$$\forall a, \mathbf{w}'_a = \mathbf{w}_a - \eta_t \nabla_a \quad (4)$$

where $t$ is the number of observations received so far, $\eta_t = \frac{1}{\mu \cdot t}$ is the learning rate and $\nabla_a$ is a sub-gradient of the objective function that defines the task of learning a SVM (*i.e.* minimize the regularized empirical risk):

$$\min_{\mathbf{w}} \frac{\mu}{2} ||\mathbf{w}||^2 + \frac{1}{m} \sum_{\mathbf{x}, y} \ell(y, \mathbf{x}, \mathbf{w}) \quad (5)$$

where $\mu$ is the regularization constant, $||\mathbf{w}||^2 = \sum_a ||\mathbf{w}_a||^2$ the norm of the weight vector and $m$ the total number of observations. As explained in [20], $\nabla_a = \mu \cdot \mathbf{w}_a$ if the loss (Eq. (3)) is null. If the loss is above zero, then:

$$\nabla_a = \begin{cases} \mu \cdot \mathbf{w}_a - \phi_a(\mathbf{x}) & \text{if } a = y \\ \mu \cdot \mathbf{w}_a + \phi_a(\mathbf{x}) & \text{if } a = r \\ \mu \cdot \mathbf{w}_a & \text{otherwise} \end{cases} \quad (6)$$

This update corresponds to a step of a stochastic gradient descent optimizing the objective function of Equation (5).

### 3.3. Features

The features used to decide which action must be performed are strongly inspired by features used in online clustering [21] and structure features used in speaker identification task [12].

As explained in Section 3.1, we maintain throughout the inference a history of all the past actions. Thanks to this history, it is possible to associate each speaker recognized to the list of the utterances she (presumably) pronounced and compute several quantities that can describe her, such as the *center*, defined as the mean of the observations attached to her or the *radius* defined as the average of the distance between the observations referring to this speaker and her center.

For a given action, the feature function $\phi_l$ associated to the $l$-th speaker defines three different kind of features[1]:

- **features describing the observation** such as the distance between the current observation and the previous ones, distance between the current observation and the center of this speaker, a Boolean indicating whether the observation is further than the radius of this speaker, ...
- **features describing interactions between speakers** such as the number of seconds since this speaker has last spoken, a Boolean indicating whether this speaker was the last one to speak, whether there was a silence or not, ...
- **features characterizing the speaker** such as the number of utterances attributed to this speaker, the increase in intra-cluster distance variance or the increase of the speaker radius.

Note that none of these features rely on the actual identity of the speaker during training. Labels are only used to decide whether two observations were uttered by the same speaker.

## 4. Experiments

### 4.1. Datasets

We relied on the REPERE[2] corpus [22] to pre-train the recurrent neural network used for extracting neural embeddings. It is composed of various TV shows (around news, politics and people) from two French TV channels for a total of 137h (50 of which are manually annotated). In practice, we used a subset made of 37.5h of speech uttered by a total of 1,178 different speakers to train the network.

Speaker diarization experiments were performed on the TV subset of the ETAPE[3] dataset [23]. It contains 29h of TV broadcast (18h for training, 5.5h for development and 5.5h for test)

[1] Please refer to the source code, available at `https://perso.limsi.fr/wisniews/recherche/`, to see the full set of features used in our experiments

[2] `islrn.org/resources/360-758-359-485-0/`

[3] `islrn.org/resources/425-777-374-455-4/`

from three French TV channels (also around news, politics and people).

### 4.2. Neural embedding

In this paragraph, we first evaluate the intrinsic quality of the proposed neural embedding.

**Implementation details.** 59-dimensional acoustic features are extracted every 10ms on a 25ms window using Yaafe toolkit [24]: 19 Mel-Frequency Cepstral Coefficients (MFCC), their first and second derivatives, and the first and second derivatives of the energy. The neural network embeds 1s-long sequences (*i.e.* 100 frames) into 192-dimensional embeddings (three levels with 64 dimensions each, as shown in Figure 1). It is trained for 400 epochs on the REPERE dataset, using mini-batches of 60 sequences from 20 different speakers (3 sequences per speaker). An epoch ends when every speaker has been seen at least once by the network.

**Results.** 100 sequences are extracted randomly for each of the 61 speakers in the ETAPE development set. The *"same/different"* experiment consists in a binary classification task: given any two of those sequences, decide whether they were uttered by the same speaker, or two different speakers. This is achieved by thresholding the computed distance between their embeddings. While TristouNet embeddings [6] achieved 17.3% equal error rate (EER) on sequences of 1 second, they are significantly outperformed by our proposed multi-level approach, which brings the EER down to 12.2%.

### 4.3. Low-latency speaker diarization

In this paragraph, we evaluate the overall low-latency speaker diarization system on the test set of ETAPE dataset (5.5h).

**Evaluation metric.** While the diarization error rate (DER)[4] is the *de facto* standard metric for comparing different diarization approaches [1], it is usually not enough to understand the type of errors commited by the system. Therefore, we also report purity [25] and coverage [26] which are two dual evaluation metrics providing additional insight on the behavior of the system. Over-segmented results (*e.g.* too many speaker clusters) tend to lead to high purity and low coverage, while under-segmented results (*e.g.* when two speakers are merged into one large cluster) lead to low purity and higher coverage. We rely on *pyannote.metrics* [27] open-source toolkit to compute those metrics.

**Implementation details.** The ETAPE training set (18h) is used to train the SVM multi-class classifier, while the development set (5.5h) is used to tune two hyper-parameters: the regularization constant $\mu$ and the number of steps in the stochastic gradient descent (Equation (4)). Their value is chosen using a grid search to optimize the DER on the development set of the ETAPE corpus. Note that we rely on the reference speech/non-speech segmentation in all experiments.

**Baseline.** We compare our low-latency approach with our offline in-house multi-stage speaker diarization system [28]. It relies on the sequence of a few module: segmentation into acoustically homogeneous segments using Gaussian divergence, Bayesian Information Criterion (BIC) clustering, Viterbi resegmentation and Cross-Likelihood Ratio (CLR) clustering. It is only one DER point worse than the best performing system of the official ETAPE benchmark submitted by LIUM [29].

Table 1: *Diarization error rate (DER, %), purity (Pur., %) and coverage (Cov., %) on the ETAPE development and test sets.*

| **Subset** | **Low-latency (ours)** | | | **Offline baseline** [28] | | |
|---|---|---|---|---|---|---|
| | DER | Pur. | Cov. | DER | Pur. | Cov. |
| dev. | 20.9 | 84.1 | 66.4 | 11.8 | 89.3 | 87.6 |
| test | 25.1 | 82.9 | 72.8 | 12.4 | 88.5 | 85.6 |

**Results.** Table 1 reports the results on the ETAPE development and test subsets. Our low-latency approach degrades prediction performance by around 10 DER points. Assessing to which extent this drop in performance is a problem for downstream applications is an open question. The result of the proposed approach are not surprising. Indeed, by design, it only considers a very small audio window, while the baseline realizes two passes over the entire stream. Moreover, the baseline automatically segments the audio stream into homogeneous segments and can therefore use longer, more consistent segments while the proposed approach uses windows of fixed length.

Comparing the purity and coverage of the two approaches show that the the proposed method tends to detect more speakers than the baseline: while their purity is almost the same, the coverage of the former approach is much smaller than the one of the latter approach. In practice, on the test set, the low-latency method detects almost 2.5 times more speakers than there are in the reference.

## 5. Conclusion

We have developed a very low latency (1 second) online speaker diarization system based on the combination of neural embeddings and incremental structure prediction. Experiments on the ETAPE broadcast news benchmark validate the approach, with only around 10 DER points increase when compared to a state-of-the-art offline system that does multiple passes on the complete audio files. We have also significantly improved the discriminative power of neural embedding of short (1 second) speech segments, going from 17.3% in our previous work [6] down to 12.2% identification error rate.

A detailed analysis of the influence of the latency on the overall performance of the system remains to be done, in order to find the best compromise between latency and performance. For instance, for applications where offline speaker diarization is not technically feasible (*e.g.* with very long audio files) but needs to be indexed very accurately, one may use the online speaker diarization approach using a higher latency. It should also be possible to run several online speaker diarization systems in parallel, with various levels of latency, to jointly benefit from the fast decision-taking capabilities of low-latency systems and higher accuracy of the ones with higher latency.

## 6. Acknowledgements

This work was partly supported by ANR through the ODESSA (ANR-15-CE39-0010) and MetaDaTV (ANR-14-CE24-0024) projects.

[4] All reported DER are computed using a 250ms collar.

# 7. References

[1] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356 –370, Feb. 2012.

[2] M. Moattar and M. Homayounpour, "A review on speaker diarization systems and approaches," *Speech Communication*, vol. 54, pp. 1065–1103, 2012.

[3] D. Liu and F. Kubala, "Online speaker clustering," in *2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003, pp. 572–575.

[4] T. Oku, S. Sato, A. Kobayashi, S. Homma, and T. Imai, "Low-latency speaker diarization based on Bayesian information criterion with multiple phoneme classes," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4189–4192.

[5] K. Markov and S. Nakamura, "Improved novelty detection for online GMM based speaker diarization." in *Proc. of the 9th Annual Conference of the International Speech Communication Association (Interspeech 2008)*, 2008, pp. 363–366.

[6] H. Bredin, "TristouNet: Triplet Loss for Speaker Turn Embedding," in *ICASSP 2017, IEEE International Conference on Acoustics, Speech, and Signal Processing*, New Orleans, USA, March 2017.

[7] H. Daumé, III and D. Marcu, "Learning as search optimization: Approximate large margin methods for structured prediction," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 169–176. [Online]. Available: http://doi.acm.org/10.1145/1102351.1102373

[8] L. Shen, G. Satta, and A. Joshi, "Guided learning for bidirectional sequence classification," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, 2007, pp. 760–767. [Online]. Available: http://aclweb.org/anthology/P07-1096

[9] M. Collins and B. Roark, "Incremental parsing with the perceptron algorithm," in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, Barcelona, Spain, July 2004, pp. 111–118. [Online]. Available: http://www.aclweb.org/anthology/P04-1015

[10] J. Nivre, "Algorithms for deterministic incremental dependency parsing," *Comput. Linguist.*, vol. 34, no. 4, pp. 513–553, 2008. [Online]. Available: http://dx.doi.org/10.1162/coli.07-056-R1-07-027

[11] L. Aufrant and G. Wisniewski, "PanParser: a Modular Implementation for Efficient Transition-Based Dependency Parsing," LIMSI-CNRS, Technical Report, 2016.

[12] E. Knyazeva, G. Wisniewski, H. Bredin, and F. Yvon, "Structured Prediction for Speaker Identification in TV Series," in *Interspeech 2015, 16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, September 2015.

[13] G. Gelly and J.-L. Gauvain, "Spoken Language Identification using LSTM-based Angular Proximity," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[15] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," in *European Conference on Computer Vision (ECCV)*, 2016.

[16] S. Funk, "Rmsprop loses to smorms3," 2015, http://sifter.org/~simon/journal/20150420.html.

[17] S. Ross and J. A. D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 661–668.

[18] Y. Goldberg and J. Nivre, "Training deterministic parsers with non-deterministic oracles," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 403–414, 2013. [Online]. Available: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/145

[19] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, Mar. 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=944790.944813

[20] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 807–814. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273598

[21] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1009783824328

[22] A. Giraudel, M. Carr, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard, "The REPERE corpus: a multimodal corpus for person recognition." in *LREC*, 2012, p. 11021107. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/707_Paper.pdf

[23] G. Gravier, G. Adda, N. Paulson, M. Carré, A. Giraudel, and O. Galibert, "The ETAPE corpus for the evaluation of speech-based TV content processing in the French language," in *LREC - Eighth international conference on Language Resources and Evaluation*, Turkey, 2012, p. na. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00712591

[24] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software." in *ISMIR 2010, 11th International Society for Music Information Retrieval Conference*, 2010, pp. 441–446.

[25] M. Cettolo, "Segmentation, classification and clustering of an Italian broadcast news corpus," in *Content-Based Multimedia Information Access-Volume 1*, 2000, pp. 372–381.

[26] J.-L. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data." in *ICSLP 1998, 5th International Conference on Spoken Language Processing*, vol. 98, no. 5, 1998, pp. 1335–1338.

[27] H. Bredin, "pyannote.metrics: a Toolkit for Reproducible Evaluation, Diagnostic, and Error Analysis of Speaker Diarization Systems," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017. [Online]. Available: http://pyannote.github.io/pyannote-metrics

[28] C. Barras, X. Zhu, S. Meignier, and J. L. Gauvain, "Multi-Stage Speaker Diarization of Broadcast New," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1505–1512, Sep. 2006.

[29] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Interspeech*, 2013.