IMPROVING THE PREDICTION OF THERAPIST BEHAVIORS IN ADDICTION COUNSELING BY EXPLOITING CLASS CONFUSIONS

Zhuohao Chen[1], *Karan Singla*[1], *James Gibson*[1], *Dogan Can*[1], *Zac E Imel*[2], *David C. Atkins*[3], *Panayiotis Georgiou*[1], *Shrikanth Narayanan*[1]

[1]Signal Analysis and Interpretation Lab, University of Southern California, Los Angeles, CA, USA
[2]Department Educational Psychology, University of Utah, Salt Lake City, UT, USA
[3]Department of Psychiatry and Behavioral Sciences, University of Washington, Seattle, WA, USA

[1]sail.usc.edu, [2]zac.imel@utah.edu, [3]datkins@u.washington.edu

ABSTRACT

In this work we address the problem of joint prosodic and lexical behavioral annotation for addiction counseling. We expand on past work that employed Recurrent Neural Networks (RNNs) on multimodal features by grouping and classifying subsets of classes. We propose two implementations: One is hierarchical classification, which uses the behavior confusion matrix to cluster similar classes and makes the prediction based on a tree structure. The second is a graph-based method which uses the result of the original classification just to find a certain subset of the most probable candidate classes, where the candidate sets of different predicted classes are determined by the class confusions. We make a second prediction with simpler classifier to discriminate the candidates. The evaluation shows that the strict hierarchical approach degrades performance, likely due to error propagation, while the graph-based hierarchy provides significant gains.

Index Terms— behavioral signal processing, multimodal, class confusions, class hierarchy, graph-based

1. INTRODUCTION

Identifying communicative behaviors in counseling conversations is a challenging and important task. Better understanding of behaviors in psychotherapy interactions could enable better treatment by establishing metrics for therapy quality as well as tracking patient progress. In this work, we investigate the problem of improving classification of behaviors in psychotherapy by exploiting confusions between behaviors of interest. We evaluate our proposed methodology, using the data from a particular type of psychotherapy called Motivational Interviewing (MI).

Motivational interviewing is a client-oriented counseling method that helps people resolve ambivalent feelings and insecurities to find the internal motivation they need to change their behavior. This approach is extensively used in treating alcohol and drug abuse problems. Some recent studies employ deep learning frameworks for predicting therapist behaviors in MI with lexical features [1, 2, 3, 4]. The work in [5] presents a multimodal approach for modeling utterance-level behaviors and reveals that using prosodic features in addition to lexical features outperforms single modality models.

In a multiclass scenario, we assume that the classes are statistically independent of each other, which rarely happens in practice. In our task, this assumption is unrealistic since behavioral codes are human-defined and not orthogonal, and the number of classes is not small. Motivated by the point that in multi-class classification, classifying a subset of classes is generally less challenging and more accurate, we proposed two approaches to tackle this problem for psychology behavior predictions in this paper. The first is a class hierarchy approach which predicts class labels following a tree structure [6, 7, 8]. The second is a graph-based approach that is two-step: first a classifier will predict the label of a given utterance; based on the predicted label it will perform a second classification using a simpler classifier that distinguishes the predicted class from classes which are likely to be misclassified as the predicted label. Both methods require the information of the class confusions, which are generated using baseline models for predicting the therapist behaviors in counseling sessions. The evaluations of the prediction are measured by the average F1 score.

2. DATASETS

The data we use comes from Motivational Interviewing sessions presented in [9, 10]. Some previous works using this dataset are described in [1, 2, 4, 5]. There are 337 transcribed sessions coded by experts at the utterance level with behavioral labels following the Motivational Interviewing Skill Code (MISC) manual [11]. The original MISC has 19 behavioral codes. Can et al. and Xiao et al. proposed different ways of clustering the codes in order to address the sparsity of some codes in the data [1, 12]. In this paper, we take the strategy proposed by Xiao et al. grouping all counselor codes into 8 categories. We remove backchannels without timestamps which cannot be aligned and split the data into training and testing sets by sessions with roughly 2:1 ratio. The statistics of data are shown in Table 1.

Table 1: Frequency of samples for misc codes

Code	Description	#Train	#Test
FA	Facilitate	1194	496
GI	Giving Information	12241	4643
RES	Simple Reflection	4594	1902
REC	Complex Reflection	3613	1235
QUC	Closed question (Yes/No)	4393	2066
QUO	Open question (Wh- type)	3871	1445
MIA	MI adherent	2948	1521
MIN	MI non-adherent	890	433
Total		33744	13741

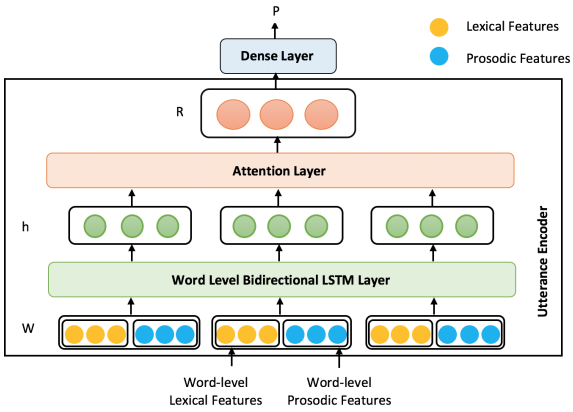
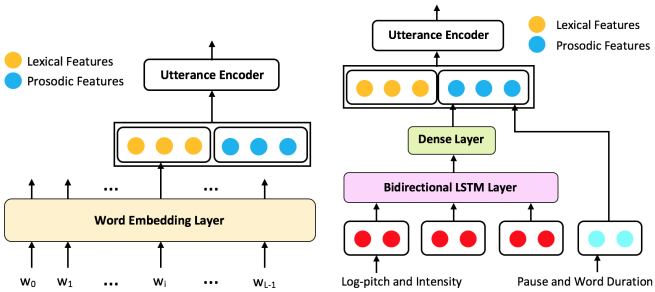


Fig. 1: Architecture for utterance encoder

3. MODEL DESCRIPTION



(a) Lexical features (b) Prosodic features

Fig. 2: Word-level lexical and prosodic features.

We consider each utterance as a sequence $w = \{w_0, w_1, ...w_{L-1}\}$, where L is the number of words in the utterance. Each w is represented by its lexical information (word embeddings) along with the corresponding prosodic information from the time aligned audio signal. We then assume a function $c = f(w)$ mapping w to a MISC code $c \in \{1, 2, ..., C\}$, where C is the number of classes (behavioral codes). We aim to find a function $f*$ that minimizes the error between the predicted and annotated codes.

We employ a multimodal approach for this task. The model architecture is shown in Fig. 1. The bottom layer is a bidirectional LSTM layer with 256 dimensions. For each word, the lexical features and prosodic features are concatenated and then fed into the LSTM layer. The attention mechanism above the hidden layer of LSTM is used for accessing the internal memory of the system, which helps learn the importance of different words for the meaning of the sentence. The attention mechanism is configured in the same way as described in [13]. The dense layer which has 256 input dimensions on the top takes the output of the recurrent layer and generates the prediction vector of all possible MISC codes. We name this lexical and prosodic combination model Combo-LP. If prosodic features are removed, the model is reduced to a single lexical modality (Single-L).

3.1. Lexical embeddings

Each word w_i is mapped to a 100-dimension feature vector via a word embedding layer [14]. The embedding layer is pre-trained by the training utterances plus the general psychotherapy corpus [15]. The architecture of it is shown in Fig. 2a. We threshold the word sequence length to 50, which covers approximately 99% of the utterances. Longer utterances are processed by tail truncation, while shorter ones are padded with zeros.

3.2. Prosodic Features

The prosodic features we use include pause, word duration, and embeddings of pitch and intensity.

3.2.1. Pause and Word Duration:

Both features are extracted via the aligned word information. The pause of the i^{th} word is defined as the duration of the end of the word w_i and the start of the word w_{i+1}. The pause feature of a word is normalized by dividing the actual pause length by the average pause length of the same speaker in one session, clipped to a maximum value of 5. We also normalize the features of word duration in the same way as the pause.

3.2.2. Embeddings of Pitch and Intensity:

We use the logarithmic value of pitch which is more relevant to what we perceive as pitch. The intensity is presented by the first MFCC coefficient which denotes an overall measure of signal loudness. Both features are extracted from 25 ms frames with 10 ms shift using the Kaldi speech recognition toolkit [16, 17]. Embeddings of pitch and intensity are implemented as shown in Fig. 2b. We feed sequences of frame-level log-pitch and intensity features into a bidirectional LSTM layer of 8 dimensions. Another dense layer of 10 dimensions is set above to produce the embedding vectors which keeps updating during training. The pitch and intensity values are computed every 10 ms. In our data 99% of words are shorter than 1 second. We thus extract the context windows from $(k-50)^{th}$ frame to $(k+50)^{th}$ frame for each word, assuming the k^{th} frame is located at the center of the word. We also experimented with zero padding for shorter words but it proved to perform worse thus no padding was employed. The 10-dim embedding vectors, together with pause and word duration form the prosodic, 12-dimensional, feature vector.

3.3. Training

For optimization, we use Adam [18] with a learning rate of 0.001 which is decayed by a factor of 0.9. We train the LSTM model up to 40 epochs with an early stopping strategy and only save the model with the lowest validation loss. Two dropout layers are set for the word-level LSTM layer and the dense layer on the top with the rate of 0.3. To deal with the class imbalance problem, we assign weights for each class inversely proportional to their class frequencies.

4. EXPLOITING CLASS CONFUSIONS

4.1. Hierarchical Classification

The hierarchical classification utilizes a hierarchical division of the output space, it clusters the similar classes and decomposes the multiclass problem into a hierarchy of simpler classification problems. This application is taken in many fields including human behavior [19, 20, 21]. In this article, the correlations between classes are determined by the confusion matrices. We use 10-fold cross-validation

on training set to get confusion matrices $M_0, M_1, ...M_9$ and compute their mean by $\overline{M} = \sum_i M_i/10$. Then we define the distance between the classes to measure how similar they are. There are multiple ways to compute the class distance using confusion matrix, here we apply the method in [22]. We first normalize $\overline{M}$ by:

$$Q_{ij} = \frac{\overline{M}_{ij}}{\sum_k \overline{M}_{ik}} \tag{1}$$

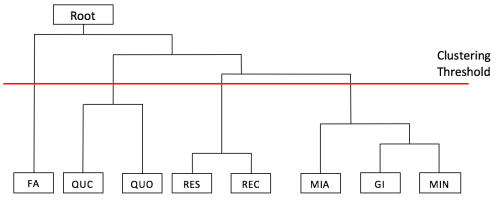
where Q_{ij} shows the ratio of class i being misclassified as class j. Then we define the distance between the classes i and j:

$$d_{ij} = \begin{cases} 1 - \frac{Q_{ij}+Q_{ji}}{2}, & i \neq j \\ 0, & i = j \end{cases} \tag{2}$$

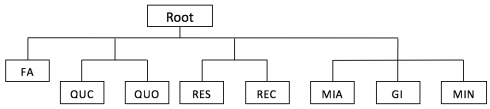
From the definition, we observe that $d_{ij} \in [0, 1]$. The more similar the two classes i and j are, the smaller the value of d_{ij} is.

Classes are clustered using a single-link hierarchical agglomerative clustering (HAC) algorithm based on their distances [23]. The hierarchical structure of the clustering result for Combo-LP model is shown in Fig. 3a, where the class label is determined hierarchically by 7 binary classifiers.

We also construct a flattened hierarchy to investigate reducing error compounding. For both Single-L and Combo-LP model, we set the same threshold for emerging classes and fatten the hierarchy under each node at the 1^{st}-level in the original hierarchy (nodes that are at distance 1 from the root). Fig. 3b presents the flattened hierarchy of Combo-LP model.



(a) Original hierarchy



(b) Flattened hierarchy

Fig. 3: Hierarchical and flattened code structures.

4.2. Graph-based Approach

The general idea of the graph-based approach is to select the subset of classes as candidates given the predicted label of the flat classifier, and then distinguish the candidate classes by a simpler classifier. Where the candidate set of classes for each predicted label is determined by class confusions. An early work trying this approach is called GraphSVM presented in [6]. Recent research uses a similar approach to learn fine-grained features to distinguish a subset of classes that boost the performance of image classification of the basic CNN model [24].

Unlike [6] that focuses on the percentage of misclassified documents, we focus on misc codes. We obtain the flat multiclass classifier F for all classes by training the baseline model and generate the average confusion matrix $\overline{M}$. Then we compute:

$$p_{ij} = \frac{\overline{M}_{ij}}{\sum_k \overline{M}_{kj}} = \hat{P}(\omega \ \in \Omega_i \ | F(\omega) = j) \tag{3}$$

where ω is an instance of the sample space, Ω_i presents the space of class i and p_{ij} is the estimated conditional probability of how likely the instance ω belongs to class i when the flat classifier F predicts its class label as j.

The pseudocode of the algorithm is shown in Algorithm 1. For simplicity, we use class 1, class 2, ...class 8 to denote the class FA, GI, RES, REC, QUC, QUO, MIA, MIN respectively. For each class j, we initialize the candidate class set by $S(j) = \{j\}$ and add any class i to $S(j)$ when the inequalities at line 4 are satisfied. The parameter T is the threshold for selecting the classes which are similar to the predicted class. We tried different values for the threshold T and found $T = 0.12$ is the best. Moreover, the values between between 0.04 and 0.2 give the similar results. In the rest of this paper we employed $T = 0.12$ for both Single-L and Combo-LP baseline models. The constraint $\overline{M}_{ij} > \overline{M}_{ji}$ is used to keep the balance of the precision and recall and prevent reducing too many true positive samples in the second prediction.

For any candidate set $S(j)$ containing more than one element, we train the candidate classifier H_j with the training data belonging to all classes in $S(j)$.

During testing, we make the first prediction using the flat classifier F. Assuming the predicted class is k, if $|S(k)| > 1$, we make the second prediction by classifier H_k, otherwise we don't need the refinement and just keep the previous predicted class label.

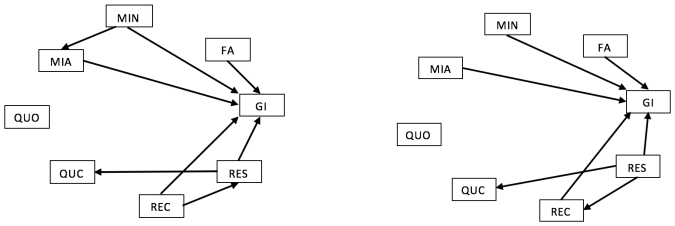
Fig. 4 presents candidate sets graph for Single-L and Combo-LP models with each class pointing to any other classes in its candidate set. In this graph-based approach, we take the advantages of both the flat classifier for all classes, which prevents the error from compounding, and the smaller classifiers of fewer classes, which can predict more accurately.

Algorithm 1 Graph-based Approach

1: Initialize p_{ij} and $S(j) = \{j\}$ for all $j \in \{1, 2, ..., 8\}$
2: **for** $j = 1, 2, ..., 8$ **do**
3: **for** $i \in 1, 2, ..., 8$ **do**
4: **if** $p_{ij} > T$ and $\overline{M}_{ij} > \overline{M}_{ji}$ **then**
5: $S(j) = \{i\} \bigcup S(j)$
6: Train the candidate classifier H_j of the classes in $S(j)$ for all j that $|S(j)| > 1$
7: **for** each instance x **do**
8: predict the class label c_l by using the flat classifier F(x)
9: **if** $|S(c_l)| > 1$ **then**
10: make the final prediction of x by using $H_{c_l}(x)$

5. EXPERIMENTAL EVALUATION

We first compute the average F1 and accuracy over 10 runs for the Single-L and Combo-LP models respectively. Then we perform the original hierarchy, flattened hierarchy and graph-based approaches



(a) Single-L (b) Combo-LP

Fig. 4: Candidate set graph, $T = 0.12$.

on them. For the original hierarchy and flattened hierarchy methods we both generate 10 sequences of the classifiers for the Single-L and Combo-LP hierarchy architectures respectively and calculate their mean metric values. For the graph-based approach, we fix the threshold with the optimal value $T = 0.12$ and also get 10 results for both baseline models by repeating the procedures from line 6 to 10 of Algorithm 1 with the same classifiers used for computing the average metrics of the original models.

5.1. Overall Performance Comparison

The overall performances of different methods are presented in Table 2 whereas the standard deviations of average F1 scores are in the brackets. The Combo-LP outperforms the Single-L in terms of both average F1 and accuracy for all approaches. The Combo-LP also significantly improves the best multimodal approach in [5] with p-value < 0.01 whose mean and standard deviation for this task are 56.41% and 0.63% over 10 runs. Because the model in [5] simply computes the mean and standard deviation of word-level prosodic features across the frames which fails to learn the variation within a word. When comparing different approaches we discover that the graph-based pattern achieves the best F-score when applied to the same baseline model. The two hierarchy methods both degrade the flat classification, especially the original hierarchy which gets the worst results for both Single-L and Combo-LP. The graph-based Combo-LP performs best with an average F1 of 58.76% and accuracy of 63.28%. It has significant gains over any other approaches with p-value < 0.01. The graph-based Single-L approach also shows statistical significance p-value < 0.01.

Table 2: Overall performance comparison

	Average F1 (%)		Accuracy (%)	
Approach	Single-L	Combo-LP	Single-L	Combo-LP
Flat Classifier	56.12 (0.45)	57.48 (0.71)	59.83	60.08
Original Hierarchy	54.76 (0.62)	55.05 (0.84)	59.17	59.53
Flattened Hierarchy	55.70 (0.58)	56.16 (0.64)	59.39	59.73
Graph-based	57.52 (0.31)	58.76 (0.42)	62.04	63.28

5.2. Graph-based Results of Each Class

Table 3 shows how F1 scores change when applying the graph-based algorithm on baseline models. The graph-based approach improves the predictions of Combo-LP for most classes including FA, GI, REC, QUC, MIA and MIN, reduces the F-score of RES with 0.46%. For the Single-L, it only reduces the average F1 of class REC and increases the other classes except for QUO. As shown in Fig. 4, in both models the candidate set of QUO has no other code and it is not in any other candidate set either, so the average F1 of QUO will not change. We conclude that the graph-based model benefits the predictions of majority classes.

Table 3: F-score for each Class

	F-score {%}			
Class	Combo-LP	Graph-based Combo-LP	Single-L	Graph-based Single-L
FA	73.10	75.49	68.54	71.47
GI	65.83	71.25	65.48	69.64
RES	48.53	48.07	45.68	48.80
REC	44.73	46.65	45.77	44.30
QUC	71.41	71.69	70.70	71.08
QUO	80.43	80.43	80.07	80.07
MIA	53.98	54.38	52.35	52.87
MIN	21.88	22.09	20.39	21.94

Table 4: Data of candidate sets in graph-based Combo-LP

Class	Candidate Set	F1-Cand (%)	F1-Flat (%)	Proportion (%)
FA	{FA, GI}	90.76	69.47	82.97
RES	{GI, RES, REC, QUC}	72.52	57.63	90.12
REC	{GI, REC}	79.70	55.28	64.18
MIA	{GI, MIA}	75.39	59.91	85.45
MIN	{GI, MIN}	62.37	45.86	61.61

5.3. Discussion of Two Approaches

The results of the original hierarchy and flattened hierarchy demonstrate that the class hierarchy suffers a lot from error compounding. From Fig. 4 we find that most nodes linked with at least one other node and they compose of a connected graph. Thus in class hierarchy we always group the classes into different spaces, which are not orthogonal, and there are many misclassified samples in early stage which degrades the classifications at deeper levels.

In Table 4, we display further information of candidate sets with more than one element in graph-based Combo-LP. We use F1-Cand to denote the average F1 of candidate classifications and show the means of the original F1 scores of candidate classes in the F1-Flat column. In the last column, we present the proportion of the samples in any of the candidate classes to the frequency of the predicted labels. Comparison of F1-Cand and F1-Flat shows that the candidate classifiers have evident gains over the original classifiers. We also discover that candidate sets within several classes consist of a majority of the samples given the predictions, which means the effects of the other classes are limited. This evidence demonstrates why the graph-based algorithm helps improve the performance.

6. CONCLUSION

In this paper, we employed a modified multimodal deep learning framework to predict behaviors of therapist in MI addiction sessions and explored using class confusion to improve accuracy. Experimental results demonstrate that applying the graph-based algorithm with multimodal features approach achieves the best performance while strict hierarchical classification makes the prediction worse due to error propagation. We also showed that the graph-based approach can benefit predictions consistently for most classes. In addition, we discussed why these two approaches have different results based on data of candidate sets and confusion matrices. In the future, we plan to apply the graph-based approach to multilabel classification and modeling empathy in addiction counseling. These tasks are at turn level which includes multiple utterances.

7. REFERENCES

[1] Bo Xiao, Dogan Can, James Gibson, Zac E Imel, David C Atkins, Panayiotis G Georgiou, and Shrikanth S Narayanan, "Behavioral coding of therapist language in addiction counseling using recurrent neural networks.," in *Interspeech*, 2016, pp. 908–912.

[2] James Gibson, Dogan Can, Bo Xiao, Zac E Imel, David C Atkins, Panayiotis Georgiou, and Shrikanth Narayanan, "A deep learning approach to modeling empathy in addiction counseling," *Commitment*, vol. 111, pp. 21, 2016.

[3] Mehedi Hasan, Alexander Kotov, April Idalski Carcone, Ming Dong, Sylvie Naar, and Kathryn Brogan Hartlieb, "A study of the effectiveness of machine learning methods for classification of clinical interview fragments into a large number of categories," *Journal of biomedical informatics*, vol. 62, pp. 21–31, 2016.

[4] James Gibson, Dogan Can, Panayiotis Georgiou, David C Atkins, and Shrikanth S Narayanan, "Attention networks for modeling behaviors in addiction counseling," in *Proc. Interspeech*, 2017.

[5] Karan Singla, Zhuohao Chen, Nikolaos Flemotomos, James Gibson, Dogan Can, David Atkins, and Shrikanth Narayanan, "Using prosodic and lexical information for learning utterance-level behaviors in psychotherapy," *Proc. Interspeech 2018*, pp. 3413–3417, 2018.

[6] Shantanu Godbole, Sunita Sarawagi, and Soumen Chakrabarti, "Scaling multi-class support vector machines using inter-class confusion," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 513–518.

[7] Shailesh Kumar, Joydeep Ghosh, and Melba M Crawford, "Hierarchical fusion of multiple classifiers for hyperspectral data analysis," *Pattern Analysis & Applications*, vol. 5, no. 2, pp. 210–220, 2002.

[8] Yangchi Chen, Melba M Crawford, and Joydeep Ghosh, "Integrating support vector machines in a hierarchical output space decomposition framework," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*. IEEE, 2004, vol. 2, pp. 949–952.

[9] David C Atkins, Mark Steyvers, Zac E Imel, and Padhraic Smyth, "Scaling up the evaluation of psychotherapy: evaluating motivational interviewing fidelity via statistical text classification," *Implementation Science*, vol. 9, no. 1, pp. 49, 2014.

[10] John S Baer, Elizabeth A Wells, David B Rosengren, Bryan Hartzler, Blair Beadnell, and Chris Dunn, "Agency context and tailored training in technology transfer: A pilot evaluation of motivational interviewing training for community counselors," *Journal of substance abuse treatment*, vol. 37, no. 2, pp. 191–202, 2009.

[11] William R Miller, Theresa B Moyers, Denise Ernst, and Paul Amrhein, "Manual for the motivational interviewing skill code (MISC)," *Unpublished manuscript. Albuquerque: Center on Alcoholism, Substance Abuse and Addictions, University of New Mexico*, 2003.

[12] Doğan Can, David C Atkins, and Shrikanth S Narayanan, "A dialog act tagging approach to behavioral coding: A case study of addiction counseling conversations," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[15] Zac E Imel, Mark Steyvers, and David C Atkins, "Computational psychotherapy research: Scaling up the evaluation of patient–provider interactions.," *Psychotherapy*, vol. 52, no. 1, pp. 19, 2015.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[17] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2494–2498.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] Zhongzhe Xiao, Emmanuel Dellandrea, Weibei Dou, and Liming Chen, "Hierarchical classification of emotional speech," *IEEE Transactions on Multimedia*, vol. 37, 2007.

[20] Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan, "Emotion recognition using a hierarchical binary decision tree approach," *Speech Communication*, vol. 53, no. 9-10, pp. 1162–1171, 2011.

[21] Yuan-Pin Lin, Chi-Hong Wang, Tien-Lin Wu, Shyh-Kang Jeng, and Jyh-Horng Chen, "EEG-based emotion recognition in music listening: A comparison of schemes for multiclass support vector machine," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 489–492.

[22] Daniel Silva-Palacios, Cèsar Ferri, and María José Ramírez-Quintana, "Probabilistic class hierarchies for multiclass classification," *Journal of computational science*, vol. 26, pp. 254–263, 2018.

[23] Robin Sibson, "Slink: an optimally efficient algorithm for the single-link cluster method," *The computer journal*, vol. 16, no. 1, pp. 30–34, 1973.

[24] Zhenhua Wang, Xingxing Wang, and Gang Wang, "Learning fine-grained features via a CNN tree for large-scale classification," *Neurocomputing*, vol. 275, pp. 1231–1240, 2018.