Recognizing Song Mood and Theme: Leveraging Ensembles of Tag Groups

Michael Vötter, Maximilian Mayerl, Günther Specht, Eva Zangerle
Universität Innsbruck, Austria
{firstname.lastname}@uibk.ac.at

ABSTRACT

In this year's MediaEval *Emotions and Themes in Music* task, the goal was to assign emotion and theme tags to songs. In this paper, we describe our (Team UIBK-DBIS) approach to solving this task. We extend the neural network model approach of our last year's submission, based on a Convolutional Recurrent Neural Network (CRNN), by building an ensemble model that utilizes spectral features. Our approaches achieve a ROC-AUC score between 0.626 and 0.707 on the provided test set.

1 INTRODUCTION

The *Emotions and Themes in Music* task of the MediaEval 2020 workshop requires to detect a song's mood and theme based on audio descriptors of the song. The prediction is done in a multi-label fashion where a total of 56 mood and theme tags are available in the dataset collected from Jamendo. The dataset as well as the split (split-0) used for the task were created by Bogdanov et al. [2]; details can be found in the overview paper [1].

Our neural network ensemble approach is an extension of our Convolutional Recurrent Neural Network (CRNN) approach [6] used to solve last year's task. We confirm last year's findings, that augmenting samples by choosing multiple random windows from the provided mel-spectrograms improves the results of both CNN and CRNN models. Further, we show that building an ensemble model of tag groups achieves improved F_1 results. The underlying code is available on GitHub[1].

2 APPROACH

Our models for MediaEval 2019 [6] have shown varying prediction results depending on the target tag—possibly because mood and theme are two different concepts with some overlap or correlation. Hence, we hypothesize that a single model trained to predict tags for both has to learn different concepts at the same time, potentially degrading accuracy. Inspired by ensemble models, we propose to use multiple independent models. As it was infeasible to create an ensemble in a fully fledged one-versus-rest fashion due to computational constraints, we split the emotion and theme tags into tag groups. This allows training one model per group. For the final predictions as required for the task, these models have to be combined into an ensemble.

2.1 Data Preprocessing

We used two different ways of pre-processing the provided mel-spectrograms for our model, which are both based on the windowing approach introduced by Mayerl et al. [6]. Both use a window size of 1366.One pre-processing strategy used the center window approach, where one sample is taken from the center of each song. The other pre-processing strategy was used to augment data for underrepresented tags, and uses windows taken from random positions within the song. As the most frequent tag occur around 14 times more often than the least frequent ones, we first extracted 14 random windows per song. Based on the tag counts in the dataset, we then categorize each song into one of four different categories. To assign each song to a category, we defined decision boundaries at $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$ of the count of the most frequently occurring tag. The first group contains songs that have a tag (the one with the highest overall count assigned to this song) assigned that has a count $\geq \frac{1}{2}$ of the maximum value, the second group contains remaining songs with a tag count $\geq \frac{1}{3}$ of the maximum value, etc. For these groups, we kept 1, 2, 3 or 4 randomly selected samples, respectively, of each song contained in the train or validation set. This procedure results in a train and validation set with better balance between tags. In the following, the strategy using center windows is referred to as *raw*, and the other strategy is referred to as *augmented*.

2.2 Ensemble Model

The motivation for building an ensemble model stems from the fact that the given tags cover two partly overlapping concepts, namely mood (emotions) and theme (topic). Further, we have seen substantial differences when comparing the per-tag scores of our last year's submissions [6]. To build an ensemble, we split the given tags into different groups and train one base model (CNN or CRNN) per group. We propose the following three splitting strategies:

- *linear*: Splits the tags in two equally sized consecutive groups based on lexicographical order.
- *performance*: Splits the tags into two equally sized groups ordered by the best scores (F_1 and PR-AUC).
- *manually*: Uses two or three tag groups that were manually assigned. The split into the two groups *mood* and *theme* was determined by majority vote among four different human judges. As a tie breaker (necessary for the tags *love* and *upbeat*), we used a coin flip. The three group split was created by assigning all tags with a kappa score of one to the respective *mood* or *theme* group and all others to an *uncertain* group (exact groups see README on GitHub).

Each model hence predicts a disjoint subset of target tags based on the given mel-spectrogram input. To obtain the final predictions, we merge those predictions to get the overall ensemble prediction.

[1] https://github.com/dbis-uibk/MediaEval2020

MediaEval'20, December 14-15 2020, Online

2.3 CNN Model

We use a CNN model as originally introduced by LeCun et al. [5]. Our CNN model uses the mel-spectrogram as input. Similar to the CRNN model of [6], we use a padding layer right after the input layer with a width of 1440, where the input with of 1366 is zero-padded left and right with the same size. This padding layer is followed by five blocks each containing two successive 2d-convolution layers with ELU activation, followed by a max-pooling layer and a dropout layer. The convolution layers use a kernel size of 3x3 while the max pooling layer uses a pooling size of 2x2. The dropout rate is 0.1, and the number of filters per block is 32, 32, 64, 64, and 64, respectively. After that, we use a dense layer with a width of 256 and ELU activation followed by a dropout layer with a dropout rate of 0.2, followed by another dense layer using sigmoid activation and a width of 56 to fit the expected output shape. We train our model using the RMSprop optimizer with categorical cross-entropy loss. This results in a network being able to predict probabilities per tag. Binary tags are predicted by setting a threshold per tag that gets optimized by finding the "elbow" in the ROC curve computed on the validation set as already done in [6].

2.4 CRNN Model

This model is a slight adaption of the model used in last year's submission of Mayerl et al. [6]. The model consists of convolutional layers followed by recurrent and dense layers, based on the architecture introduced by Choi et al. [3]. In total, there are four convolutional blocks each consisting of a 2D-convolution layer, followed by a batch normalization layer, ELU activation, a max pooling layer and a dropout layer. After these two blocks, two GRU layers are used that are followed by dropout layer with dropout rate of 0.3 and a dense layer with a size of 56 to fit the required output shape. We train this model using the Adam [4] optimizer with a categorical cross-entropy loss in contrast to the binary cross-entropy used last year. Again the threshold used for binary tag predictions is set using the ROC curve method described in Section 2.3.

2.5 Submissions

Based on the setup described above, we submitted the following five runs where each model is trained for 20 epochs:

- *Run #1*: CRNN ensemble model using manual mood and theme split, and trained on augmented data.
- *Run #2*: CRNN ensemble model using manual mood and theme split, and trained on raw data (center windowing).
- *Run #3*: CRNN ensemble model using manual mood, theme and uncertain split, and trained on augmented data.
- *Run #4*: CRNN ensemble model using two splits based on the F_1-score, and trained on augmented data.
- *Run #5*: CRNN model trained on augmented data.

3 RESULTS AND ANALYSIS

In addition to the results for the submitted runs described in Section 2.5, we included further results in Table 1. All models contained in Table 1 are trained for 20 epochs. The name of each approach encodes the used dataset and model. The first letter specifies the used dataset: *a* stands for the augmented dataset (Section 2.1), while *r* means that the raw dataset without augmentation was used. This is followed by the type of model, *CNN* (Section 2.3 or *CRNN* (Section 2.4). If the model name is prefixed with *e* this means that the given model was used in an ensemble (Section 2.2). In case of ensemble models, the remainder of the approach name specifies the splitting strategy (Section 2.2) and how many splits (last token of the approach name) were used.

Table 1: Evaluation results. Submitted runs are in bold.

Approach	ROC-AUC	PR-AUC	F_1 (micro)	F_1 (macro)
Popularity	*0.500*	*0.032*	*0.057*	*0.003*
VGG-ish	*0.726*	*0.108*	*0.177*	*0.166*
a-cnn	0.643	0.069	0.091	0.089
a-ecnn-manually-2	0.637	0.069	0.087	0.086
r-cnn	0.637	0.069	0.089	0.089
r-ecnn-manually-2	0.626	0.066	0.088	0.088
a-crnn	**0.704**	**0.097**	**0.100**	**0.104**
a-ecrnn-linear-2	0.703	0.093	0.103	0.109
a-ecrnn-manually-2	**0.689**	**0.090**	**0.106**	**0.111**
a-ecrnn-manually-3	**0.683**	**0.086**	**0.101**	**0.103**
a-ecrnn-f1-2	**0.685**	**0.090**	**0.100**	**0.100**
a-ecrnn-pr-auc-2	0.685	0.078	0.099	0.097
r-crnn	0.707	0.089	0.103	0.107
r-ecrnn-linear-2	0.697	0.089	0.104	0.106
r-ecrnn-manually-2	**0.695**	**0.089**	**0.098**	**0.105**
r-ecrnn-manually-3	0.691	0.086	0.102	0.106
r-ecrnn-f1-2	0.689	0.093	0.099	0.107
r-ecrnn-pr-auc-2	0.696	0.088	0.097	0.105

From Table 1, we can see that using the augmented dataset leads to slightly improved results over the raw dataset. Hence, we were able to reproduce last year's findings of Mayerl et al. [6]. Further, we see that the CRNN model outperforms the CNN model across all setups. Additionally, the results show that for most of the metrics, the ensemble method using two or three manual splits of the tags outperforms splittings based on performance of a simple linear split. The submitted CRNN model trained on the augmented dataset with tags manually split into mood and theme shows the best results for F_1. Additional analyses showed that this performance gain stems from a better precision score, as this approach also has the best precision score (not shown in the table) as well as a good recall score. In contrast, the CRNN model (without using an ensemble strategy) shows the best results for ROC-AUC (on the raw dataset) and PR-AUC (on the augmented dataset). Further analyses is need to find the cause for that difference.

4 SUMMARY AND OUTLOOK

Our proposed approach of using an ensemble design, is inspired by the varying performance of last year's models when looking at the per-tag scores. We showed that for traditional scoring metrics such as F_1, the ensemble models are able to outperform a plain CRNN approach. Nevertheless, we could not outperform the VGG-ish baseline. Potential future work includes changing the models contained in the ensemble. For example, using different types of models for different groups of tags would be an option. Further, the ensemble may be extended by training multiple models per group or even multiple models on different groups with e.g., a majority vote for the final predictions.

REFERENCES

[1] Dmitry Bogdanov, Alastair Porter, Philip Tovstogan, and Minz Won. 2020. MediaEval 2020: Emotion and Theme Recognition in Music Using Jamendo. In *Proc. of the MediaEval 2020 Workshop*. Online, 14-15 December 2020.

[2] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. 2019. The MTG-Jamendo Dataset for Automatic Music Tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*. Long Beach, CA, United States.

[3] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2392–2396.

[4] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).

[5] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1, 4 (1989), 541–551.

[6] Maximilian Mayerl, Michael Vötter, Hsiao-Tzu Hung, Boyu Chen, Yi-Hsuan Yang, and Eva Zangerle. 2019. Recognizing Song Mood and Theme Using Convolutional Recurrent Neural Networks. In *Proc. of the MediaEval 2019 Workshop*. Sophia Antipolis, France, 27-30 October 2019.