EXPLORING PRE-TRAINED NEURAL AUDIO REPRESENTATIONS FOR AUDIO TOPIC SEGMENTATION

Iacopo Ghinassi[◇,⋆] *Matthew Purver*[◇,†] *Huy Phan*[◦] *Chris Newell*[‡]

[⋆]Centre for Digital Music / [◇]Cognitive Science Research Group,
School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[†]Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[◦] Amazon Alexa, Cambridge, MA, USA
[‡] BBC R&D, UK

ABSTRACT

Recent works have shown that audio embeddings can improve automatic topic segmentation of formats such as radio shows. In this work we expand the work in that direction by showing how and which publicly available, pre-trained neural audio embeddings can perform the task, without the need of any further fine-tuning of the audio encoders. The ranking of the encoders suggest that neural encoders pre-trained for speaker diarization and general purpose audio classification are the best suited to be used as features, beating non-neural baselines. We show that we can obtain perfect results on a newly created random dataset similar to the one used in previous work. We also show for the first time results on real-world data, proving that our method can be applied to actual radio shows with good results, but the choice of audio encoders is extremely important in order to achieve those. Finally, by releasing the datasets we used we make the contribution of providing the first (to our knowledge) publicly available, free of charge datasets for audio topic segmentation of media products.

Index Terms— topic segmentation, neural audio embeddings

1. INTRODUCTION

The task of topic segmentation is concerned with segmenting a long document into topically coherent segments [1]. The input document can be composed of text, audio, video or a mixture of them. Many approaches have been developed during past years to tackle this problem. Recent works have explored the use of information from the audio stream alone in segmenting, for example, radio shows (see Fig. 1). The use of just audio implies that no transcripts are needed, therefore saving resources and possible causes of noise, where the transcripts were automatically generated. In this context, [2] has shown that the use of audio embeddings can outperform previous approaches. In doing so, the authors left open several points. Firstly, the encoders used to extract the embeddings were trained on the same corpus used for segmentation. Because of this, problems of efficiency and performance arise: new classification models need to be fitted each time to extract audio embeddings and different models might have a big impact on segmentation performance. Secondly, non-neural baselines for audio embeddings were not presented, leaving an open question about whether using neural networks for extracting audio embeddings is indeed the most effective approach.

[◦]The work was done when H. Phan was at the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK and the Alan Turing Institute, UK and prior to joining Amazon.

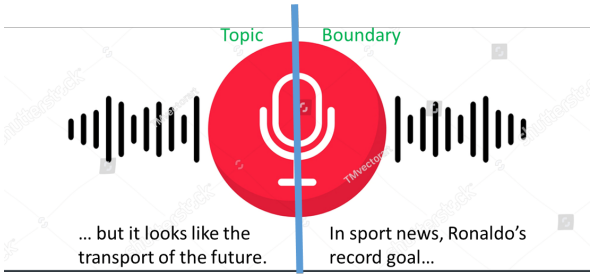

Fig. 1. Segmenting an audio file into topically coherent segments implies recognising when the underlying topic shifts. In our case, we propose a method based on audio information only.

Thirdly, the corpus used to evaluate topic segmentation was not publicly released and statistics about the corpus were not included either, leaving open questions about the replicability of the proposed approach. Fourthly, the corpus used to evaluate topic segmentation was artificially constructed by concatenating random portions of radio programmes in a setting similar to [3]. The use of artificial datasets for evaluating topic segmentation attracted criticism in the past years [4], leading to the need of real-world evaluation to confirm the utility of audio embeddings for the task.

In this work we address the above problems by introducing a topic segmentation system that works on embeddings derived from open-source, pre-trained audio encoders and by introducing three new datasets. We explore different research questions:

RQ1: Do audio representations from pre-trained neural networks generally yield better results than traditional features?

RQ2: Which audio representation is more appropriate for the segmentation task?

RQ3: Is our method effective in real-world datasets and how does the performance vary?

We answer these questions by using a set of different pre-trained neural and non-neural features as input for a BiLSTM network to recognise the boundaries of topic segments.

We show good segmentation results on real-world data and that pre-trained neural representations are effective for the task, but performance varies greatly according to architecture and pre-training objectives of the encoders. We achieve a perfect score on our random dataset and, by releasing the two real-world datasets, we open the field to real-world use of the proposed system.

2. RELATED WORK

2.1. Audio Topic Segmentation System

The direct use of audio to segment an audio stream into coherent segments has been researched for a long time. When audio is involved, the coherence of two consecutive audio frames is defined in terms of change of speakers, channel or environment [5]. In the context of media products' segmentation, a change in these three factors is likely to represent a change in the content of the audio stream, indicating a possible topic shift. Attempts to directly segment the acoustic input into topically coherent segments also exists [6–8]. Various works on topic segmentation using audio generally employed unsupervised methods developed for text-based topic segmentation such as the TextTiling algorithm [9] and similar methods exploiting vector similarities [7,10]. More recently, supervised approaches, mostly in the form of recurrent neural networks, are usually preferred [2].

The main focus of research into topic segmentation using audio has been the choice of which acoustic features to use. Different types of hand-crafted acoustic features have been explored. One obvious feature that has been applied early on is the binary feature indicating whether a portion of the audio stream consists of just silence [11]. Features that can indicate a change in speaker are useful for the task of segmentation as well. Previous literature has, in fact, used the identity of speakers (e.g., if the current speaker is predicted to be the anchorman in a news show) as features [12]. Mel-frequency cepstral coefficients (MFCCs) have also been directly used for audio topic segmentation [5–7, 13]. Other typical audio features for topic segmentation are the ones related to prosody like F0 contour, speech rate and power. Such features have been used, for example, by [8, 12, 14] and they have their theoretical justification in the fact that certain prosodic patterns have been shown to correlate with a change in topic in multiple scenarios [15].

Another set of acoustic features that have been more recently used in topic segmentation relate to neural audio classification. In this context, [2] trained a neural audio classifier to extract acoustic embeddings to be used in a long short-term memory (LSTM) network for segmenting radio shows. These features were even more effective than using text features in that context. The fact that these type of features have not been thoroughly investigated for topic segmentation gives a lot of scope for experimentation, without necessarily having to train audio encoders from scratch.

2.2. Datasets for Audio Topic Segmentation

In terms of data, many datasets have been proposed in the literature for evaluating topic segmentation of texts [3, 16, 17]. This is not the case, however, for audio topic segmentation. In this context, the only (to our knowledge) publicly and free of charge dataset that is annotated for topic segmentation system is the one released by [18], which include a number of academic meetings. That domain, however, is quite restricted and for the more specific domain of media products segmentation, much fewer datasets have been developed and none has been publicly released. The most famous examples of such datasets are those released for the various editions of the TDT challenges [19] and two editions of the TRECVID challenge [20]. Generally, it is quite common for authors to use private datasets created by the authors themselves [21] when experimenting on TV newscasts and TV programmes, which can be explained both with the fact that the TDT datasets must be paid for and that they are by now quite old (about 20 years). Given all these considerations, together with the analysis of new and better features for audio topic segmentation we release three new publicly available and free of charge datasets for audio topic segmentation, which we describe in more details below. In our case, we manage to overcome copyright limitations by releasing the audio embeddings extracted for each programme, but not the original audio. This strategy has the advantage of allowing the experimentation and advancement of the field, while protecting the interests of the copyright owners and the privacy of participants in the programme.

3. METHODOLOGY

3.1. Pre-Trained Audio Embeddings

We explore different pre-trained audio encoders and manually engineered audio embeddings. In the choice of such encoders we followed the work of [22] that evaluated different neural encoders for a variety of audio tasks. We used the following pre-trained neural encoders:

X-Vectors (XVEC): this architecture was proposed by [23] in the context of speaker diarization, obtaining state-of-the-art results at the time of publication. Here we used the pre-trained X-vectors model implemented by SpeechBrain[1].

Openl3 (OP): this model was proposed by [24] as a relatively lightweight, publicly available pre-trained model for audio classification. We used the official implementation[2].

Wav2Vec2 (WAV): the Wav2Vec family of neural audio encoders was proposed as a way of translating the self-supervised language modelling objective from the text domain in the audio domain [25]. Wav2vec2 is the most recent model following this approach and it achieved state-of-the-art performance in various speech-related tasks, especially in under-resourced settings [26]. Here we used the pre-trained model released by huggingface[3].

CREPE (CR): this model was proposed in the context of F0 tracking, reaching state-of-the-art upon its release [27]. We used the pre-trained Pytorch implementation[4].

3.2. Non-neural Baselines

We include two non-neural baselines obtained by using manually engineered features:

Prosodic (PR): this set of features reflects the ones from [28]. They include means and standard deviations of pause durations, F0 contour, pitch jump (difference of F0 with respect to the previous frame), mel-frequency bins and their delta values for a total of 167 features per audio embedding.

MFCC: this set of features includes the mean and standard deviation of the first 50 mel-frequency cepstral coefficients, as well as their first-order delta values.

3.3. Topic Segmentation Model

3.3.1. Basic unit extraction

In order to extract the audio embeddings and train a topic segmentation system, we first need to pre-process the audio file and divide it in smaller portions, which will be passed individually to the encoder. A straightforward approach is that of [2] which simply extracts the audio embeddings from non-overlapping 1-second portions of audio. This approach leads to a severe sparsity problem, as the number

[1] https://huggingface.co/speechbrain/spkrec-xvect-voxceleb

[2] https://github.com/marl/openl3

[3] https://huggingface.co/docs/transformers/model_doc/wav2vec2

[4] https://github.com/maxrmorrison/torchcrepe

Table 1. Dataset details and statistics (durations are expressed in seconds). Where applicable, we report mean and 95% confidence interval.

Dataset	Total Audio Files	Total Number of Segments	Avg. File Segments	Avg. File Duration	Avg. Segment Duration
BMAT-ATS	100	995	9.95 ± 0.05	393.06 ± 13.82	41.52 ± 1.29
NonNews-BBC	57	399	7 ± 0.52	2131.19 ± 0.52	511.18 ± 39.13
RadioNews-BBC	48	702	12.77 ± 1.19	1631.75 ± 227.08	138.18 ± 12.29

of positive units will remain the same while the number of negative examples grow linearly with the length of the input audio.

The effect of this class imbalance is unclear in the work of [2], given the absence of dataset statistics. We decided to segment the audio in non-overlapping 1-second portions as well, but to mitigate the sparsity problem we employed a different training loss, described in more details in the experimental details section.

3.3.2. Audio encoding

Once having pre-segmented the input file in non-overlapping 1-second portions and having an encoder enc, we obtain the initial sequence of audio embeddings $\mathcal{X}_i = enc(a_i)$, where a_i denotes a 1-second audio unit, and reduce them to one embedding with $x_i = pool(\mathcal{X}_i)$. Since the different encoders used take in different audio frame sizes for embedding extraction, the length of the embedding sequence $\mathcal{X}_i$ extracted for each audio portion depends on a specific encoder. The pooling function $pool$ reduces the embedding sequence to a single embedding per audio portion, so that $x_i \in \mathbf{R}^{1\times d}$, with d being the number of elements in the embedding yielded by the given encoder. We consider different pooling functions. In the case of XVEC, PR and MFCC the pooling function is simply the identity function $pool(\mathcal{X}_i) = \mathcal{X}_i \equiv x_i$ as they already summarise all of the input by default. In the case of the other three encoders, OP, WAV, and CR, we experimented with the following:

STD: $pool(\mathcal{X}_i) = mean(\mathcal{X}_i) \oplus std(\mathcal{X}_i)$ with $mean$ representing average, std the standard deviation and $\oplus$ concatenation.

MAX $pool(\mathcal{X}_i) = max(\mathcal{X}_i)$ with max representing max pooling across different vector's dimensions.

LAST: $pool(\mathcal{X}_i) = \mathcal{X}_i[N]$ where N is the length of the sequence $\mathcal{X}_i$, thus $\mathcal{X}_i[N]$ corresponds to the last embedding vector in the sequence. By using the last element from each sequence we aim to maximise the potential distance between the different audio units.

DELTA: $pool(\mathcal{X}_i) = \mathcal{X}_i[N] - \mathcal{X}_{i+1}[N]$. In this last setting the pooling operation computes the difference of the last embedding of the current sequence $\mathcal{X}_i$ and the last embedding of the next sequence $\mathcal{X}_{i+1}$, and thus, modelling directly in the feature space the difference between the two audio units in the audio file and possibly aiding the segmentation model in finding areas of maximal divergence.

3.3.3. Segmentation model

Finally, the embeddings x_i are passed into a bidirectional LSTM (BiLSTM) network, followed by a dense layer and sigmoid activation function for classification. At inference time, a boundary is output when the output score is greater than a threshold θ, that was set to 0.5. The segmentation model is illustrated in Fig. 2.

3.4. Data

Previous work using audio embeddings neither publicly released the dataset that was used nor included statistics about it. Moreover, it was specified that the dataset was built using a process similar to [3], i.e., concatenating random portions of radio shows and treating the

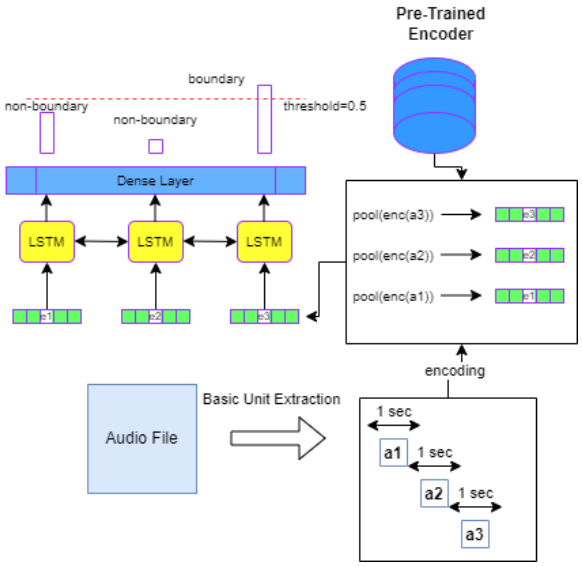

Fig. 2. Our proposed segmentation procedure.

concatenation points as topic boundaries. The random concatenation of different audio portions might prove unrealistic in the audio domain as drastic variations in the signal power or in the overall spectral envelope are easier to capture than, e.g., random concatenation of words, where some semantic understanding is still needed. To overcome these limitations we introduce three new datasets.

BMAT-ATS: This dataset replicates the random setting of [3], but it has been built on top of the publicly available BMAT dataset [29] and can therefore be re-created given the released recipe. The dataset has been built by using the English news segments contained in the original BMAT dataset. Such segments have been previously segmented by splitting each file where there are intervals of foreground music. In this way the length of individual files ranged from 5 seconds to 1 minute. 100 random audio files have then been created by concatenating 10 such smaller files randomly and by avoiding repetitions in the patterns.

NonNews-BBC: This dataset consists of 57 magazine-style radio programmes from BBC Radio channels covering different non-news topics. Topic boundaries were manually annotated by experts and the length of each audio file range from 20 to 60 minutes approximately. Given copyright limitations, we release just the extracted embeddings and the relative ground-truth labels from this dataset.

RadioNews-BBC: This dataset consists of a selection of 48 news bulletins from local, national and World Service radio channels by the BBC. Topic boundaries were manually annotated by experts and the length of each audio file range from 9 to 60 minutes approximately. Given copyright limitations, we release the extracted embeddings and the relative ground-truth labels from this dataset.

The statistics of the three datasets are presented in Table 1.

Table 2. Boundary similarity scores B-F1, B-Precision and B-Recall (%). **Bold** indicates best in column. $^{+}$,* indicate statistical significance ($p < 0.05$): $^{+}$= significantly better than best non-neural baseline; *= significantly worse than best configuration. The higher the better.

Dataset	**BMAT-ATS**			**NonNews-BBC**			**RadioNews-BBC**		
Encoder	***B-F1***	***B-Precision***	***B-Recall***	***B-F1***	***B-Precision***	***B-Recall***	***B-F1***	***B-Precision***	***B-Recall***
PR	35.66*	23.43	89.57	32.02*	30.24	39.42	24.12*	21.50	29.05
MFCC	95.90	93.65	98.61	33.15*	25.56	58.28	**60.76**	72.16	55.19
XVEC	94.19	**100**	89.32	**49.88**$^{+}$	**56.47**	**60.77**	59.29	78.98	49.05
OP + STD	82.06*	87.79	77.70	36.87*	55.86	28.47	54.86	85.87	41.25
WAV + STD	94.11	93.53	95.48	30.27*	33.09	29.86	51.76	70.68	41.63
CREPE + STD	**96.61**	98.63	94.91	29.75*	29.79	35.03	47.68*	48.77	48.95
OP + MAX	86.60*	83.87	91.60	28.93*	40.22	24.95	55.15*	81.57	42.34
WAV + MAX	91.51	90.07	93.88	24.29*	33.85	20.61	45.41*	52.67	40.93
CREPE + MAX	93.26	97.02	90.60	30.45*	33.88	32.91	48.77*	81.05	36.50
OP + LAST	92.56	90.17	96.08	36.99	46.67	33.30	51.09	**92.47**	35.94
WAV + LAST	48.54*	38.41	73.37	5.00*	3.19	11.47	16.61*	11.14	35.08
CREPE + LAST	21.57*	14.50	86.76	1.99*	1.44	3.12	8.36*	5.16	53.16
OP + DELTA	95.40	92.15	**99.30**	30.40*	29.51	35.37	35.80*	24.96	**77.25**
WAV + DELTA	18.60*	12.97	52.51	0*	0	0	6.02*	3.67	18.32
CREPE + DELTA	12.42*	7.42	54.15	0*	0	0	2.59*	1.43	14.27

4. EXPERIMENTAL SETUP

For all settings, the segmentation model was designed to have 2 BiLSTM layers with a total of 512 hidden units per layer (i.e. 256 hidden units per direction). In optimising the model, we used a learning rate of 0.001, Adam optimiser [30] and the focal loss function. Focal loss was proposed by [31] in the context of object detection but it has been proved to be useful also for topic segmentation in the text domain [32]. The loss includes a weighting that does not just re-balance the classes (as we have very imbalanced classes, as shown above), but that also assign more weight to examples that were "hard", in the sense of closer to the decision boundary, as those are probably the ones that the system will need to be more careful in classifying. For the loss parameters we used $\alpha = 0.9$ to assign more weight to positive class in training and we kept the default value $\gamma = 2$ for the weighting of "hard" examples[5]. We adopted an early stopping mechanism when no improvement on validation data was observed over 50 epochs and we set a maximum of 1000 training epochs. We applied dropout before and after the recurrent layers and we chose the best dropout probabilities $p1$ and $p2$ based on validation results, with the search space $p1 \in \{0, 0.2, 0.5\}$ for dropout before the recurrent layers and $p2 \in \{0, 0.2, 0.5\}$ for dropout after.

The models were evaluated using a variant of precision, recall and F1 scores named boundary similarity [33]. Standard accuracy metrics, in fact, are considered too strict for topic segmentation [34]. Because of this, boundary similarity was recently proposed as a way to account for near misses by introducing a loss modulated on the standard minimum edit distance algorithm, where predicted and ground truth boundaries are seen as two strings and the cost of editing the prediction to become the ground truth string is converted into a confusion matrix from which standard F1, precision and recall can be computed [33]. For computing the metric we have used the standard *segeval* python library [35]; as per standard F1, precision and recall the resulting accuracy metrics are named B-F1, B-Precision and B-Recall respectively when reporting our results.

For each configuration, we used pre-defined train, validation and test splits, where the validation and test folds are about 15% of the original dataset and the training set amount to the remaining 70%.

We also report the p-value associated with the null hypothesis of each encoder not leading to test results significantly different from the best performing non-neural baseline from the same dataset: this way we test whether neural audio embeddings can improve over traditional features (RQ1). We obtained the p-values by running a two sample t-test between the B-F1 test scores from the each configuration and the B-F1 test scores from the best-performing non-neural baselines, that in every case turned out to be MFCC. Similarly, we report whether each configuration is significantly worse than the best performing one by using the B-F1 scores of the best setting and comparing them with all other settings' via a two sample t-test.

5. RESULTS

Table 2 shows the results for all configurations. In describing the results, we first turn to the real-world datasets, NonNews-BBC and RadioNews-BBC. X-vectors are consistently the best audio embeddings when looking at B-F1 score for NonNews-BBC and they are the best among neural embeddings for RadioNews-BBC. If we look at NonNews-BBC dataset, especially, we can observe how XVEC is the only neural audio encoders that is significantly better than the MFCC baseline, even though OP+LAST is not significantly worse than XVEC for this dataset if we set the null hypothesis rejection threshold to 0.05 (it is significantly worse when the threshold is 0.10). Surprisingly, using MFCC in RadioNews-BBC proves to be the most effective approach; even though the results are not significantly better than XVEC, this evidence suggests that neural embeddings are not always better than traditional non-neural features, directly contradicting the claim of [2].

Furthermore, other neural embeddings such as OP+LAST do perform better than MFCC in NonNews-BBC, but probably because of a large variance in test results, this improvement does not seem statistically significant. The results from RadioNews-BBC confirm this as most neural configuration, even though marginally worse than MFCC, are not significantly worse, reflecting the high variability in test results. In both NonNews-BBC and RadioNews-BBC the precision of neural encoders tend to be bigger, suggesting that the model tends often to undergenerate topic boundaries, probably as a result

[5]The parameter names follow the implementation we used, available at https://pytorch.org/vision/stable/_modules/torchvision/ops/focal_loss.html

of overfitting to the majority class (i.e. no boundary).

Especially OpenL3 seems to follow this pattern, having the highest precision scores in RadioNews-BBC and positioning itself as second best neural encoder overall, followed by CREPE and, last, Wav2Vec2. Manually engineered prosodic features always position in the middle for NonNews-BBC datasets, being better than WAV and CREPE but worse than XVEC, OP and MFCC. When looking at RadioNews-BBC those manually engineered features underperform also with respect to the best settings of Wav2Vec2 and CREPE.

Whereas the poor performance of Wav2Vec2 is not striking in the sense that the general purpose audio encoder was pre-trained on mainly speech data and was reported to improve performance on tasks that are not similar to the topic segmentation one (e.g. automatic speech recognition), we might have expected CREPE and PR to perform better than they did, considering the importance of pitch as a topic shifting signal recorded by previous literature [36]. The three encoders mostly show results that are significantly worse than the best neural embeddings for both real-world datasets.

Different encoders also seem to benefit from different pooling strategies. On one hand, OpenL3 on NonNews-BBC perform the best when using the LAST pooling strategy and perform the worst with MAX, while CREPE+MAX is the best configuration for CREPE and WAV+STD is the best one for Wav2Vec2. More in details, using the LAST and DELTA pooling strategies seems to increase recall: this is likely related to the fact that the last embedding of each boundary 1-second chunks will already be positioned in the next topic segment making negative and positive samples more similar and causing the model to overgenerate topic boundaries. The DELTA pooling strategy seem to be overall the worst setting, as it totally fails when combined with WAV and CREPE and it underperforms also when applied to OpenL3 embeddings. A notable exception to this is the recall score of OP+DELTA for RadioNews-BBC, which is the highest for the dataset. It can probably be explained with what noted above about LAST and DELTA pooling effect on recall more generally. Similar observations hold for RadioNews-BBC but with the difference that OP+MAX places itself first relatively to other pooling approaches on the same embeddings, confirming the fact that the best encoder and pooling technique might change according to the data they are used on.

Experiments on BMAT-ATS clearly shows that random datasets are easier than the real-world datasets, casting doubts about whether they provide a trustworthy benchmark. The best setting in terms of B-F1 is CREPE+STD, contradicting the bad performance of this encoder for real-world datasets. XVEC reaches perfect precision, confirming it among the best approaches, while OP+DELTA is the best setting in terms of recall, probably as a result of what previously observed for the LAST and DELTA pooling. Most of the configurations, however, manage to reach over 90% for all the metrics, leading to the results being mostly not significantly different from each other.

Overall, we have shown that the proposed method can be successfully applied to real-world data (RQ3) and that pre-trained neural representations can in certain cases outperform non-neural ones in topic segmentation (RQ1): this is however dependent upon the domain and the encoder being used, contradicting the general claim that neural embeddings are always better. We, then, empirically showed which a pre-trained encoder might work the best for this purpose (RQ2), reaching the conclusion that X-vectors seem to be consistently the best option, even though not significantly better than alternatives such as OpenL3. Our results, therefore, confirmed that speaker diarization and sound classification are relevant to topic segmentation in the audio domain. Future research might explore different encoders from these two domains further.

At the same time, our results challenged previous literature by demonstrating that features related to pitch often under-perform with respect to simple MFCCs and that neural audio embeddings are not always the best choice of feature. The release of our two datasets, then, can foster more research to highlight the limit and the potential of such features in real world topic segmentation of media products.

6. CONCLUSION

In this work, we have introduced three new datasets for audio topic segmentation and experimented with different audio encoding techniques to extract audio embeddings for audio-based topic segmentation. The main question we aimed to answer is whether using pre-trained neural encoders could yield results comparable to the ones described in [2] and which neural encoder is more suitable for the task. At the same time, given pitfalls from previous work, we aimed to establish whether the use of audio embeddings for topic segmentation was still effective in real-world scenarios and whether neural encoders were actually better than more traditional features.

Our results confirmed that neural representations can yield improvements in topic segmentation over non-neural features on real-world data, but not for any given domain, as MFCCs are the best option in segmenting news podcasts. Still, neural features for speaker diarization such as X-vectors perform always as good if not better than MFCCs and future research might consider using similar, more advanced features to improve over simpler baselines.

We also made the crucial contribution of releasing three datasets for audio topic segmentation of radio shows. To the best of our knowledge, they are the first datasets of their kind to be freely and publicly available and this is particularly important given the fact that results on synthetic data was proven to be higher on average, therefore not reflecting actual performance in a real-world scenario.

Future research can build upon our observations and our released datasets to further improve the state of the art by, for example, using more advanced architectures or combining different features.

7. ACKNOWLEDGEMENTS

We acknowledge financial support from several sources: the Slovenian Research Agency via research core funding for the programme Knowledge Technologies (P2-0103), and the UK EPSRC via the projects Sodestream (Streamlining Social Decision Making for Improved Internet Standards, EP/S033564/1) and ARCIDUCA (Annotating Reference and Coreference In Dialogue Using Conversational Agents in games, EP/W001632/1).

8. REFERENCES

[1] M. Purver, "Topic segmentation," in *Spoken Language Understanding*. John Wiley & Sons, Ltd, 3 2011.

[2] O. Berlage, K. M. Lux, and D. Graus, "Improving automated segmentation of radio shows with audio embeddings," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 751–755.

[3] F. Choi, "Linear text segmentation : approaches, advances and applications," in *Proceedings of CLUK 3*, 2000.

[4] M. Georgescul, A. Clark, and S. Armstrong, "An analysis of quantitative aspects in the evaluation of thematic segmentation algorithms," in *Proc. 7th SIGdial Workshop on Discourse and Dialogue (SIGdial06)*, 2006.

[5] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, vol. 6, 1998.

[6] I. Malioutov, A. Park, R. Barzilay, and J. Glass, "Making sense of sound: Unsupervised topic segmentation over acoustic input," in *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[7] H. Chen, L. Xie, W. Feng, L. Zheng, and Y. Zhang, "Topic segmentation on spoken documents using self-validated acoustic cuts," *Soft Computing*, vol. 19, 2015.

[8] G. Tür, A. Stolcke, D. Hakkani-Tür, and E. Shriberg, "Integrating prosodic and lexical cues for automatic topic segmentation," *Computational Linguistics*, vol. 27, 2001.

[9] M. A. Hearst, "Multi-paragraph segmentation expository text," in *Proc. 32nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Jun. 1994, pp. 9–16.

[10] F. Y. Y. Choi, P. Wiemer-hastings, and J. Moore, "Latent semantic analysis for text segmentation," *Proc. 2001 Conference on Empirical Methods in Natural Language Processing*, vol. 102, 2001.

[11] C. Wang, Y. Wang, H. Y. Liu, and Y. X. He, "Automatic story segmentation of news video based on audio-visual features and text information," in *Proc. International Conference on Machine Learning and Cybernetics*, vol. 5, 2003.

[12] A. Rosenberg and J. Hirschberg, "Story segmentation of broadcast news in English, Mandarin and Arabic," in *Proc. the Human Language Technology Conference of the NAACL*, 2006, pp. 125–128.

[13] S. S. Cheng and H. M. Wang, "A sequential metric-based audio segmentation method via the bayesian information criterion," in *Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, 2003.

[14] M. Galley, K. R. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 562–569.

[15] J. Hirschberg and C. Nakatani, "Acoustic indicators of topic segmentation," in *Proc. International Conference on Speech and Language Processing*, 1998.

[16] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant, "Text segmentation as a supervised learning task," vol. 2, 2018.

[17] S. Arnold, R. Schneider, P. Cudré-Mauroux, F. A. Gers, and A. Löser, "SECTOR: A neural model for coherent topic segmentation and classification," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 169–184, 2019.

[18] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The icsi meeting corpus," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 2003, pp. I–I.

[19] J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. V. Mulbregt, "A hidden markov model approach to text segmentation and event tracking," vol. 1, 1998.

[20] W. Kraaij, A. F. Smeaton, and P. Over, "Trecvid 2004 - an overview," 2004.

[21] Émilie Dumont and G. Quénot, "Automatic story segmentation for tv news video using multiple modalities," *International Journal of Digital Multimedia Broadcasting*, vol. 2012, 2012.

[22] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, "Hear: Holistic evaluation of audio representations," in *Proc. NeurIPS 2021 Competitions and Demonstrations Track*, 2022, pp. 125–145.

[23] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2018-April, 2018.

[24] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3852–3856.

[25] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech*, 2019, pp. 3465–3469.

[26] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 449–12 460.

[27] J. W. Kim, J. Salamon, P. Q. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 161–165.

[28] E. Tsunoo, P. Bell, and S. Renals, "Hierarchical recurrent neural network for story segmentation," in *Proc. Interspeech*, vol. 2017-August, 2017.

[29] B. Meléndez-Catalán, E. Molina, and E. Gómez, "Open broadcast media audio from tv: A dataset of tv broadcast audio with relative music loudness annotations," in *Transactions of the International Society for Music Information Retrieval*, 2019, pp. 43–51.

[30] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd International Conference on Learning Representations (ICLR)*, 2015.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[32] R. Iikura, M. Okada, and N. Mori, "Improving bert with focal loss for paragraph segmentation of novels," in *Proc. 17th International Conference on Distributed Computing and Artificial Intelligence*, 2021, pp. 21–30.

[33] C. Fournier, "Evaluating text segmentation using boundary edit distance," in *Proc. 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 1702–1712.

[34] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Machine Learning*, vol. 34, 1999.

[35] C. Fournier, "Evaluating text segmentation," Master's thesis, University of Ottawa, 2013.

[36] M. Yeung, B.-L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, 1996, pp. 296–305.