

Detecting Extraneous Content in Podcasts

Sravana Reddy

Spotify

sravana@spotify.com

Yongze Yu

Spotify

yongzey@spotify.com

Aasish Pappu

Spotify

aasishp@spotify.com

Aswin Sivaraman*

Indiana University

asivara@iu.edu

Rezvaneh Rezapour*

University of Illinois

at Urbana-Champaign
rezapou2@illinois.edu

Rosie Jones

Spotify

rjones@spotify.com

Abstract

Podcast episodes often contain material extraneous to the main content, such as advertisements, interleaved within the audio and the written descriptions. We present classifiers that leverage both textual and listening patterns in order to detect such content in podcast descriptions and audio transcripts. We demonstrate that our models are effective by evaluating them on the downstream task of podcast summarization and show that we can substantially improve ROUGE scores and reduce the extraneous content generated in the summaries.

1 Introduction

Podcasts are a rich source of data for speech and natural language processing. We consider two types of textual information associated with a podcast episode: the short description written by the podcast creator, and the transcript of its audio, both of which may contain content that is not directly related to the main themes of the podcasts. Such content may come in the form of sponsor advertisements, promotions of other podcasts, or mentions of the speakers' websites and products.

While such content is tightly integrated into the user experience and monetization, it is a source of noise for many natural language processing and information retrieval applications which utilize podcast data. For example (Table 1), an episode of the podcast show *Survival* includes a promotion for an unrelated podcast *Dog Tales* about dogs; a search query for podcasts on dogs should probably not surface the *Survival* episode. Algorithms attempting to connect topics discussed in the podcast to those mentioned in the episode description, such as summarization models, would be confounded by the presence of supplementary material and URLs

* Work done while at Spotify

... sit stay and roll over with excitement for par casts endearing series dog tails. Listen to dog tails free on Spotify or wherever you get your podcast. And now back to the story. Almost immediately after setting sail on June 29th. 1871 Charles Francis Halls Arctic Expedition...
... the focus is on strengthening deficient repertoires, while systematically increasing task demands and difficulty. For more information, visit www.behaviorbabe.com . — This episode is sponsored by Anchor: The easiest way to make a podcast. https://anchor.fm/app

Table 1: Portions of a podcast transcript and episode description which contain extraneous material (in bold).

in the description. Information extraction models looking for entities may mistakenly retrieve sponsor names from advertisements.

In this paper, we introduce the problem of detecting extraneous content (which we sometimes shorthand as EC) in episode descriptions and audio transcripts. We produce an annotated corpus by taking advantage of podcast listening data, construct models to detect extraneous content, and evaluate our models for accuracy of detection, as well as for the downstream task of summarizing podcast transcripts. We also discuss some of challenges that arise while annotating and classifying extraneous content in this domain.

2 Previous Work

A related, well-studied problem is boilerplate detection on web pages, mainly involving the detection of templates, navigational elements, and advertisements (Kohlschütter et al., 2010). Such models tend to rely on the specific structure of web page boilerplate markup and characteristics. There has also been work on detecting promotional content on Wikipedia (Bhosale et al., 2013).

There are primarily two lines of work in advertisement detection and discovery in multimedia. One computes acoustic features over the entire audio to discriminate between the segments of con-

tent and the segments of advertisements (Conejero and Anguera, 2008; Melamed and Kim, 2009; Nguyen et al., 2010). The other fuses multimodal features such as visual cues to segment ad clips from televised and online videos (Lienhart et al., 1997; Duan et al., 2006; Vedula et al., 2017). Our work is closely related to Huang et al. (2018), who analyze consumer engagement on audio advertisements when compared to topical content, as we utilize the user engagement signals to predict extraneous content segments.

3 Datasets and Annotation

To create an annotated dataset, we selected a random set of podcast episodes out of the Spotify Podcast Dataset, a corpus of 105,360 episodes (Clifton et al., 2020). Each episode in the dataset has an automatically generated transcript and a short text description of the episode written by the podcast creators. We annotate both sources, creating ground-truth labels for the extraneous content detection task, using the open source software *doccano* (Nakayama et al., 2018). Annotators were instructed to select spans that correspond to extraneous content, which we defined as ads, social media links, promotions of other podcasts, and show notes that are not directly related to the episode. Respecting sentence boundaries was encouraged, but not required, to allow for cases where extraneous content starts or ends mid-sentence.

3.1 Podcast Episode Descriptions

Annotation of episode descriptions was relatively straightforward. We encountered a few corner cases: for example, an episode may in its entirety be a promotion or an ad; a description that reflects these promotions would be on-topic. In such scenarios, the annotations attempted to be consistent with the annotators’ judgments of the main topics of the episode as far as possible. Examples of annotated episode descriptions are shown in Table 6 in the appendix.

3.2 Podcast Episode Transcripts

Each transcript contains thousands of words, of which extraneous content may make up a small portion. This necessitates a way to sample an informative subset of transcript segments for annotation. We observe that if a region is extraneous to the main content, listeners of the podcast episode may fast-forward past this region or abandon listening.

To this end, we gather listener data for a subset of our corpus from Spotify, an audio streaming platform. We record the proportion of all listeners who begin streaming an episode retained at each second of the episode’s duration. Listening data was aggregated from the date of each episode’s publication – with the most recent episode published in February 2020 – through September 2020. For episodes with a substantial number of listeners, there exist distinct local minima or “dips” in the retention curves (Figure 1), which we posit may correspond to regions of extraneous content.

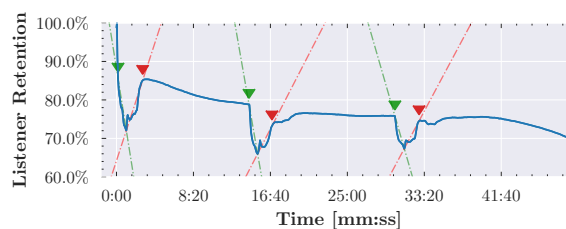


Figure 1: A podcast listener retention curve for a single podcast episode. The dips in the graph suggest potential EC regions. Start- and end-points for each dip are automatically estimated as described in Section 3.2, shown with green and red markers respectively.

To locate the center point of each dip, we first apply `SciPy` peak detection on the negative retention curve (Virtanen et al., 2020). Within ± 2 min of each peak, we calculate the slopes of secant lines passing through every point on the curve. The coordinates which maximize the secant slope within this range correspond to the start/end points of the dips. Green and red secant lines are shown in Figure 1 to illustrate this process.

The transcript is then segmented 60 sec before the starting point and 90 sec after the ending point of each dip, loosening the boundaries of the potential EC regions; these segments are then manually annotated. We note that the transcripts contain noisy text which are artifacts from an automatic speech recognition (ASR) system. An additional challenge stems from identifying native advertising, where podcast creators deliberately script product placement into the context of their content (Einstein, 2015; Hutton, 2015). Considering all cases, we attempted to best estimate the boundaries of the scripted content.

Examples of annotated transcript regions are shown in Table 7 in the appendix. The set of manual annotations is used as a *gold* labeled transcript dataset for our models.

Dataset	# of episodes	# of sentences
<i>Episode Descriptions</i>	1730 (61.58% w/ EC)	10410 (43.80% w/ EC)
<i>Transcripts (Gold)</i>	506 (62.25% w/ EC)	12918 (24.01% w/ EC)

Table 2: Our manually annotated datasets show that the majority of podcast episodes do contain extraneous content, making up a sizeable fraction of all sentences.

Of the annotated dips, 38.4% were found not to contain extraneous content. These largely correspond to episode beginnings, where listeners may skip over the introduction to show, or dynamically inserted ads that are not present in the transcript. The dip boundaries for the rest are relatively accurate against the manual annotations, with an absolute mean error of 16.0 words for the starts, and 35.2 for the ends, motivating the use of the unannotated listener dips as a *silver* training set (described in §4.2).

Sentence-Level Labels Our models for descriptions and transcripts use the sentence as the unit of classification. The annotated data is split into sentences using SpaCy¹. A sentence is labeled as extraneous if more than 50% is annotated as such (Table 2).

4 Extraneous Content Detection

4.1 Sentence-Level Classification

We built separate classifiers for detecting extraneous content in descriptions and transcripts. A pre-trained BERT (Devlin et al., 2019) cased language model² was first fine-tuned on our entire large corpus (of podcast descriptions and transcripts respectively, excluding the test set) to capture the distinctive language use of the podcast domain, and then further fine-tuned for classification on the annotated data to predict whether a sentence is extraneous. We also trained non-neural classifiers (logistic regression and SVMs) with TF-IDF unigram and bigram features (Appendix A.2).

We experimented with single sentence classification in isolation, and with the immediately preceding sentence prepended for context.³

¹<https://spacy.io>

²<https://huggingface.co/bert-base-cased>

³The two sentences are concatenated. In BERT, we include the SEP token to separate them. Where there is no preceding sentence, we use a special `__START__` token.

Dataset	Model	Single Sentence		Sentence with Context	
		Unigram	Bigram	Unigram	Bigram
<i>Episode Descriptions</i>	LR	0.893	0.818	0.858	0.850
	SVM	0.907	0.907	0.898	0.896
	BERT	0.920		0.940	
<i>Transcripts (Gold)</i>	LR	0.651	0.603	0.629	0.638
	SVM	0.675	0.679	0.663	0.670
	BERT	0.710		0.769	

Table 3: F1 score of three sentence-level EC classifiers tested across two datasets.

Results Our experiments used a training/test split of 90/10 for annotated transcripts and 80/20 for episode descriptions. The best results come from BERT where the model sees the previous sentence (Table 3). We observed that classification precision and recall are comparable on the descriptions dataset; however, with transcripts, precision tends to be lower (0.690) than recall (0.870). This suggests that the model mistakenly identifies sentences as extraneous, but has fairly good coverage. This seems to be because many sentences in extraneous content in the training transcripts are seemingly innocuous when taken out of context, confounding the model during training.

Dataset	Model	Doc. (%)	Sen. (%)
<i>Episode Descriptions</i>	Sentence-level BERT	86.4	94.9
	BERT + Change Point	89.6	95.4
	BERT + BiLSTM-CRF	87.6	95.0
<i>Transcripts (Gold)</i>	Sentence-level BERT	43.1	88.7
	BERT + Smoothing	49.1	90.9
<i>Transcripts (Silver)</i>	Sentence-level BERT	60.8	96.6
	BERT + Smoothing	66.7	97.0

Table 4: Classification accuracy of sentence-level and document-level extraneous content detection. Documents come from either the dataset of full episode descriptions, or from transcript segments corresponding to dip regions. A document EC-match occurs when *all* sentence labels agree; this is difficult to achieve with longer segments.

4.2 Document-Level Classification

We classify sentences independently, but extraneous content comes as contiguous groups of sentences. We apply non-parametric kernel regression to post-hoc smooth the sequence of individual sentence classification probabilities in the transcripts. We observe that extraneous content within episode

descriptions often appears as a contiguous block at the end, prompting us to apply a change point detector (Appendix A.3) on the sentence classification probabilities in order to detect the start of the EC block.

We also formulate the problem as sequence tagging at the sentence level, in order to allow the model to learn the label dependencies. For this, we use the BERT pooled sentence embeddings as input to a separate BiLSTM-CRF model. The BiLSTM-CRF improves over sentence-level classification but underperforms the change point detection strategy (Table 4). In future work, we would like to investigate an end-to-end BERT-BiLSTM-CRF model (Dai et al., 2019) or sequential sentence classification models (Cohan et al., 2019).

Expanded Transcript Dataset Since the manually labeled gold set is small, we create a larger silver dataset from 6401 detected listener dips across 4930 episodes by applying the best performing classifier trained on the gold data. To encode information about the dip locations to aid the model, we prepend special tokens ‘in-dip’ and ‘outside-dip’ to each sentence depending on whether the sentence lies within a listening dip.

We strip special tokens from the resulting silver set. This data is then used to train a final classifier that can detect extraneous content regions in podcast transcripts *without* listener dips. We also add a negative sample of sentences that are distant (by at least 5 minutes) from dips in the same episodes, with the assumption that these are likely to be topical.

On the same test set as the previous experiments, the document and sentence level performance increases (Table 4), proving the model benefits from the larger, albeit noisy, training set. This model is applied to the corpus in the downstream task described below.

5 Application to Podcast Summarization

We address the problem of automatically generating episode descriptions from podcast transcripts, a task similar to abstractive summarization. Within this problem, we evaluate the downstream effect of removing extraneous sentences from the training and/or test data. Alternatives to removal (such as using the model’s predictions as auxiliary inputs in the downstream system) are left for future work.

We experiment with two supervised abstractive summarization models both built using BART

(Lewis et al., 2020). The first experiment uses a model pretrained for summarization on the CNN/DailyMail dataset⁴ (Hermann et al., 2015). This model (which we refer to as BART-CNN) evaluates the extent to which extraneous text in the transcripts contribute to the presence of extraneous content in the generated descriptions. In the second experiment, we fine-tune BART-CNN on our corpus of podcasts, similar to the work of Zheng et al. (2020), using the episode transcripts as inputs and descriptions as targets. We refer to this model as BART-PODCASTS; with it, we can evaluate the effects of EC as realistic noise which may contaminate the training data of summarization models.

5.1 Experimental Setup

From the original corpus of 105,360 podcast shows, 6401 were used for training and evaluation of the two EC detection models. We filter the remaining episodes for descriptions which are distinctly short or long. The resulting dataset contains 84,451 episodes sorted by episode publication date. This is split into 82,451 episodes for training, 1000 for validation, and 1000 for evaluation.

As a baseline, we manually remove extraneous content from the episode descriptions within the test set, comparing the ROUGE score of the model outputs against the manually-cleaned descriptions as well as against the original descriptions. Additionally, we manually validate whether the generated outputs for 150 random test episodes contain extraneous content.

5.2 Results

Table 5 shows the full ROUGE-L scores of our experiments. We evaluate quality through ROUGE as well as by manually verifying for the presence of extraneous content in the outputs. BART-CNN is an out of the box summarization model, while BART-PODCASTS is the same model that is fine-tuned on our data of transcripts and descriptions. All numbers are reported on the test split of the corpus. The range of these ROUGE scores is comparable to previous podcast summarization work. Removing extraneous content is clearly beneficial for summary quality: while the baseline models have better ROUGE scores against the original (EC-containing) descriptions, the highest scoring models score better against the clean descriptions compared to the originals. With the original data, both

⁴<https://huggingface.co/facebook/bart-large-cnn>

Model	EC Removal Method	ROUGE-L against original descriptions			ROUGE-L against cleaned descriptions			Amount of EC in output (%)
		Rec.	Pre.	F	Rec.	Pre.	F	
<i>Original Descriptions (no model)</i>	n/a	1.000	1.000	1.000	0.996	0.810	0.872	50.0
<i>BART-CNN</i>	n/a	0.310	0.199	0.204	0.340	0.170	0.195	18.8
	M1	0.299	0.200	0.199	0.330	0.172	0.192	11.5
	M2	0.309	0.188	0.195	0.345	0.165	0.191	4.0
<i>BART-Podcasts</i>	n/a	0.377	0.329	0.315	0.349	0.228	0.248	73.2
	M1	0.269	0.309	0.247	0.308	0.280	0.260	8.7
	M3	0.323	0.276	0.261	0.367	0.247	0.263	15.5
	M4	0.330	0.281	0.269	0.376	0.261	0.268	2.0

Table 5: Evaluation of the effect of extraneous content detection on a downstream summarization task. Accuracies are reported in terms of recall, precision, and F1 score. The models are provided with data pre-processed through one of four EC removal strategies: (M1) cleaned test descriptions, (M2) cleaned test transcripts, (M3) cleaned training descriptions, or (M4) cleaned descriptions and transcripts for both train and test.

BART-CNN and BART-PODCASTS tend to generate descriptions that contain extraneous material. Interestingly, BART-PODCASTS, being trained on the unmodified descriptions, produces even *more* extraneous content (73.2%) than the corresponding original descriptions (50.0%), often generating ads unrelated to the actual sponsors, and non-existent URLs. While post-processing only the output summaries with no change to the model inputs is effective at minimizing extraneous content, it does so at the expense of summary quality, since the resulting summaries are significantly shortened. The best overall performance comes from detecting extraneous content on transcripts and descriptions before model training and application.

6 Conclusion

We introduced the problem of detecting extraneous content in podcast descriptions and transcripts, presented models that leverage textual and listener data, and evaluated them on a downstream summarization task.

We consider our models to be baselines for a new problem with several opportunities for future work. Although we used two separate models for the descriptions and transcripts with the view that the language patterns are different, a joint model or shared components may be able to take advantage of some of the common vocabulary. One could leverage the ‘boilerplate’ nature of some types of extraneous content like ads by detecting repeated sentences and phrases across the corpus. A language model

that is robust to noisy speech transcripts (Lin et al., 2019; Chuang et al., 2019) may improve accuracy on podcast transcripts. Given that extraneous content may appear as pre-recorded audio, or with a different speaking pitch and cadence, acoustic features alongside textual ones may be helpful.

7 Acknowledgments

We thank Ann Clifton and Jussi Karlgren for valuable assistance with the summarization model and evaluations, Richard Mitic, Laurence Pascall, Skylar Brannon, Kristie Savage, and Karishma Gulati for collaborating on identifying listening dips from user logs, Iftekhar Tanveer for helpful discussions, and the anonymous EACL reviewers for their suggestions.

References

- Shruti Bhosale, Heath Vinicombe, and Raymond Mooney. 2013. [Detecting promotional content in Wikipedia](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1851–1857, Seattle, Washington, USA. Association for Computational Linguistics.
- Yung-Sung Chuang, Chi-Liang Liu, and Hung-Yi Lee. 2019. [Speechbert: Cross-modal pre-trained language model for end-to-end spoken question answering](#). *arXiv preprint arXiv:1910.11559*.
- Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020. [100,000 podcasts: A spoken English document corpus](#). In *Proceedings*

- of the 28th International Conference on Computational Linguistics, pages 5903–5917, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. [Pretrained language models for sequential sentence classification](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.
- David Conejero and Xavier Anguera. 2008. Tv advertisements detection and clustering based on acoustic information. In *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, pages 452–457. IEEE.
- Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. 2019. Named entity recognition using bert bilstm crf for chinese electronic health records. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5. IEEE.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ling-Yu Duan, Jinqiao Wang, Yantao Zheng, Jesse S Jin, Hanqing Lu, and Changsheng Xu. 2006. Segmentation, categorization, and identification of commercial clips from tv streams using multimodal analysis. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 201–210.
- Brandon R. Einstein. 2015. Reading Between the Lines: The Rise of Native Advertising and the FTC’s Inability to Regulate It. *Brook. J. Corp. Fin. & Com. L.*, 10:225.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.
- Jason Huang, David Reiley, and Nick Riabov. 2018. Measuring consumer sensitivity to audio advertising: A field experiment on pandora internet radio. *Available at SSRN 3166676*.
- Christopher Hutton. 2015. [Podcasting embraces native advertising](#).
- Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Rainer Lienhart, Christoph Kuhmunch, and Wolfgang Effelsberg. 1997. On the detection and recognition of television commercials. In *proceedings of IEEE international Conference on Multimedia Computing and Systems*, pages 509–516. IEEE.
- H. Lin, T. Lo, and B. Chen. 2019. Enhanced bert-based ranking models for spoken document retrieval. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 601–606.
- I Dan Melamed and Yeon-Jun Kim. 2009. Automatic detection of audio advertisements. In *Tenth Annual Conference of the International Speech Communication Association*.
- Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. [doccano: Text annotation tool for human](#). Software available from <https://github.com/doccano/doccano>.
- MN Nguyen, Qi Tian, and Ping Xue. 2010. Efficient advertisement discovery for audio podcast content using candidate segmentation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010(1):1–12.
- Nikhita Vedula, Wei Sun, Hyunhwan Lee, Harsh Gupta, Mitsunori Ogihara, Joseph Johnson, Gang Ren, and Srinivasan Parthasarathy. 2017. Multimodal content analysis for effective advertisements on youtube. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1123–1128. IEEE.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. [SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python](#). *Nature Methods*, 17:261–272.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. *Transformers: State-of-the-art natural language processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan. 2020. *A baseline analysis for podcast abstractive summarization*.

A Appendix

A.1 Annotation Process and Annotated Examples

Examples of extraneous content regions of episode descriptions are shown in Table 6. and example of annotated listener ‘dip’ regions in the podcast transcripts in Table 7. The categorization into types is only for illustrative purposes and is not used in our model.

A.2 Model Training and Evaluation Details

We modified the sentence splitter in SpaCy to include ---, . . . and the three-space string as delimiters for episode descriptions, based on our observations of common patterns. Speech recognition errors/disfluencies and missing punctuation contribute to a small amount of noise in the sentence segmentations of transcripts and descriptions respectively.

For the bag of words models, we used <https://scikit-learn.org/> implementations with the default parameters and no hyper-parameter tuning.

For all Transformer models, we used the Huggingface library (Wolf et al., 2020). For summarization, we set the maximum length of the target descriptions as 250 tokens for training and generation, and the minimum length to 30 tokens. The models were trained for up to 5 epochs, with early stopping based on ROUGE-2 on the validation set. All other hyperparameters were set as the defaults specified in the Huggingface Transformers code.

For evaluation, we use the FILES2ROUGE implementation (<https://github.com/pltrdy/files2rouge>), after tokenizing both the reference and the outputs using the Moses tokenizer (<https://github.com/alvations/sacremoses>).

A.3 Change Point Detection for Episode Descriptions

As shown in Eq. 1, we can find a position $\hat{\tau}$ which maximizes the negative log-likelihood ratio R_τ of H_1 as the existence of the change point versus H_0 as no change point.

$$\hat{\tau} = \arg \max_{1 \leq \tau \leq N} R_\tau \quad (1)$$

We make the assumption that (1) there is only one change point, and (2) extraneous content appears at end the descriptions. The null hypothesis is that there is no changepoint, while the alternative hypothesis assumes that there is a changepoint at the time $t = \tau$. Here is the hypothesis test:

$$H_0 : \theta_1 = \theta_2 = \dots = \theta_{N-1} = \theta_N \quad (2)$$

$$H_1 : \theta_1 = \dots = \theta_{\tau-1} = \theta_\tau \neq \theta_{\tau+1} = \dots = \theta_N \quad (3)$$

Therefore, the likelihood is given by the probability of observation the data $x = x_1, \dots, x_n$ conditional on H_0 . In other words,

$$L(H_0) = p(x|H_0) = \prod_{i=1}^N p(x_i|\theta_i) \quad (4)$$

And the the likelihood of the alternative hypothesis is,

$$L(H_1) = p(x|H_1) = \prod_{i=1}^{\tau} p(x_i|\theta_i) \prod_{j=\tau+1}^N p(x_j|\theta_j) \quad (5)$$

The log-likelihood ratio R_τ is then,

$$\begin{aligned} R_\tau &= \log \frac{L_{H_0}}{L_{H_1}} \\ &= \sum_{i=1}^{\tau} \log p(x_i|\theta_1) + \sum_{j=\tau+1}^N \log p(x_j|\theta_2) \\ &\quad - \sum_{k=1}^N \log p(x_k|\theta_0) \end{aligned} \quad (6)$$

TYPE	EXAMPLES
Platform Ads	Transformation Church Pastor Mike Todd brings Marked Part 8- You Are Enough --- This episode is sponsored by Anchor: The easiest way to make a podcast. https://anchor.fm/app Support this podcast: https://anchor.fm/m-e8/support
Social media links, podcast promotions, sponsor ads	Crazy new years news, Dallin fantasizes about a Tesla, and then we do some hilarious Mad Libs... you have to hear these. Follow the Dashleys for more! https://thedashleysxxx.com https://facebook.com/thedashleysxxx https://instagram.com/thedashxxx https://youtube.com/thedasxxx hellodasxxxx@gmail.com Check out our new Podcast! http://anchor.fm/takingsxxxx Thank you to Skillshare for sponsoring this podcast! Get 2 months premium membership free here: skillshare.com/dashxxx Thank you to HelloFresh for sponsoring this podcast! Sign up for HelloFresh and get 10 free meals here! URL: hellofresh.com/BIGLITTLELxxxx Promo: BIGLITTLELxxxx — Support this podcast: https://anchor.fm/biglittlexxxx/support

Table 6: Annotated examples from episode descriptions. (Some of the links are masked for privacy.)

TYPE	SUBTYPE	EXAMPLES
Sponsorships / Ads	Products	... the railway tracks a mere four inches a small but deadly act of sabotage. Hey friend, I want to tell you about the new \$3 Little John from Jimmy John’s [...] or with the Jimmy John’s app at participating locations taxes and delivery fees extra. Welcome to today and True Crime a par cast original...
	Platforms	... before we move on to your next topic. We just want to say thanks to our sponsor Anchor. If you haven’t heard about Anchor [...] and it’s everything you need to make a podcast in one place download the free anchor app or go to Anchor. Um to get started, a very somber...
Podcast Promotions	Other Podcasts	... Coming up heading returns to Denmark and Begins the long journey to reclaim his throne. Darkness tragedy pain these things hide within every beloved institution and most people are none the wiser every week. [...] Search The Dark Side of or visit Park ask.com / Dark Side to listen now. Now back to the story...
	Social Media	... manipulates our relationships into deadly results at par cast we are grateful for you our listeners you allow us to do what we love let us know how we’re doing reach out on Facebook and Instagram at par cast and Twitter [...] for more information. It all began with a computer...
Not EC	Episode Introduction	... I’m your host Taylor structure of the live Daily Talk Radio Show [...] It’s getting to meet interesting inspiring and exceptional people and getting them to talk about the heaviest things in the world. So if you love real talk with a...
	Miscellaneous Content	... Not much is known about her personal life prior to her murder if she was so average, why was she living with her aunt and uncle I’m not saying there’s anything wrong with that [...] I’m sorry to say that it looks as though your husband has an advanced...

Table 7: Annotated examples from the podcast transcripts, corresponding to detected dips in listening.