Reproducing "Topic Modeling on Podcast Short-Text Metadata"

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

this is the abstract

**Additional Keywords and Phrases:** Podcast short-text metadata, …

1. 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.

1. 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.

* 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.

* 1. Experiment setup

1. Comparing The Results

In the paper the researchers used topic coherence as their prime evaluation metric. This metric (also denoted as CV reflects the coherence or quality of a topic. A higher CV 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 CV 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 CV for each topic.

* 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 (aword ∈ {0.2, 0.3, 0.4, 0.5} × aent ∈ {0.3, 0.4}) with varying number of topics (K ∈ {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  iTunes | 50.20  49.30 | 53.10  47.20 | 48.50  50.30 | 53.30  52.50 | 53.20  52.80 | 56.40  52.40 | 52.50  50.60 | 56.30  50.50 |
| 50 | Deezer  iTunes | 48.90  43.30 | 49.20  49.50 | 52.10  52.50 | 50.90  49.50 | 51.50  50.10 | 52.60  51.90 | 56.30  46.50 | 60.60  52.00 |
| 100 | Deezer  iTunes | 51.40  49.50 | 50.80  50.70 | 51.50  49.00 | 55.30  49.20 | 52.20  50.60 | 48.10  49.90 | 50.80  46.70 | 54.90  48.70 |
| 200 | Deezer  iTunes | 48.40  47.00 | 50.60  51.30 | 49.80  48.20 | 51.60  49.80 | 50.00  51.10 | 49.00  47.40 | 55.40  49.00 | 53.30  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 (aword, aent, 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.

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

* 1. 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 out results differed significantly from the ones 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  iTunes | False  False | False  False | False  False | False  False | False  False | False  False | False  False | False  False |
| 50 | Deezer  iTunes | False  True | False  False | False  False | False  False | False  False | False  False | False  False | False  False |
| 100 | Deezer  iTunes | False  False | False  True | False  False | False  False | False  False | False  False | False  False | False  False |
| 200 | Deezer  iTunes | False  False | False  False | False  False | False  False | False  False | False  False | False  False | False  False |

**WRITE SOME THINGS ABOUT THE BOOLEAN TABLES; BUT ALSO AOUT THE ABSOLUTE VALUES**

1. 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 CV metric, without scripts for a more thorough evaluation.

1. COnclusion

Reproducing the paper highlighted NEiCE's potential to improve topic coherence in short-text metadata while exposing critical challenges in reproducibility. Despite code and data limitations, our results aligned closely with the original study's trends, validating NEiCE’s effectiveness.

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.).