STREAM: Simplified Topic Retrieval, Exploration, and Analysis Module

Anton Frederik Thielmann and **Arik Reuter** and **Benjamin Säfken**
Institute of Mathematics
Clausthal University of Technology

Christoph Weisser and **Manish Kumar**
BASF
Ludwigshafen, Germany

Gillian Kant
Centre for Statistics
University of Göttingen

Abstract

Topic modeling is a widely used technique to analyze large document corpora. With the ever-growing emergence of scientific contributions in the field, non-technical users may often use the simplest available software module, independent of whether there are potentially better models available. We present a Simplified Topic Retrieval, Exploration, and Analysis Module (STREAM) for user-friendly topic modelling and especially subsequent interactive topic visualization and analysis. For better topic analysis, we implement multiple intruder-word based topic evaluation metrics. Additionally, we publicize multiple new datasets that can extend the so far very limited number of publicly available benchmark datasets in topic modeling. We integrate downstream interpretable analysis modules to enable users to easily analyse the created topics in downstream tasks together with additional tabular information. The code is available at the following link: `https://github.com/AnFreTh/STREAM`

1 Introduction

Identifying latent topics within extensive text corpora is a fundamental task in the field of Natural Language Processing (NLP) and has been of larger scientific interest since the early 2000s (Hofmann, 2001; Blei et al., 2003). Especially with the emergence of contextualized embeddings, extraction algorithms and topic models continue to evolve and achieve increasingly impressive results in terms of topic coherence (Larochelle and Lauly, 2012; Srivastava and Sutton, 2017; Chien et al., 2018; Wang et al., 2019; Dieng et al., 2020). Even, methodologically simpler methods achieve state-of-the-art results by leveraging document and word-embeddings (Sia et al., 2020; Grootendorst, 2022; Angelov, 2020).

The publication of open source software like Gensim (Řehůřek and Sojka, 2010), the Natural Language Tool Kit (nltk) (Bird et al., 2009) or SpaCy (Vasiliev, 2020) have enabled researchers to apply such models in various fields, including education (Granić and Marangunić, 2019), offsite construction (Liu et al., 2019), bioinformatics (Liu et al., 2016), communication sciences (Maier et al., 2018), finance (Thormann et al., 2021) and numerous other applications (e.g., (Hall et al., 2008; Daud et al., 2010; Boyd-Graber et al., 2017; Kant et al., 2022; Thielmann et al., 2021; Hannigan et al., 2019; Tillmann et al., 2022)).

The OCTIS (optimizing and comparing topic models is simple) (Terragni et al., 2021a) framework in particular has found favor in the scientific community and made fitting and evaluating sophisticated topic models easy and efficient. However, OCTIS lacks the methodologically simpler yet very performant models such as clustering based topic extraction (Sia et al., 2020; Angelov, 2020) and the user-centric implementation of BERTopic (Grootendorst, 2022). Especially the user-friendly implementation and visualization possibilities of BERTopic allow non-technical users to easily analyze their document corpora and visualize their results which has led to a variety of use cases especially in the social sciences (e.g. (Falkenberg et al., 2022; Jeon et al., 2023; Zankadi et al., 2023)).

We thus contribute the STREAM (**S**implified **T**opic **R**etrieval **E**xploration and **A**nalysis **M**odule) software package. It gets its acronym not only from the easy to use, user-centric topic modelling, evaluation and exploration implementation but also from the integration of downSTREAM models to analyze topic contributions to regression or classification problems.

The core of the STREAM package is built on top of the OCTIS framework and allows seamless integration of all of OCTIS' multitude of models, datasets, evaluation metrics and hyperparameter optimization techniques.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 435–444
August 11-16, 2024 ©2024 Association for Computational Linguistics

1.1 Contributions

The contributions of STREAM can be summarized as follows:

- STREAM integrates multiple clustering based topic models into the OCTIS framework (see the Appendix for a full list of all available models).
- Through interactive visualization methods, STREAM allows easy exploration and analysis of all models.
- We publicize multiple multi-modal datasets to enable researchers to compare their models beyond the standard topic modeling datasets, such as 20NewsGroups and Reuters (Mitchell, 1999; Lewis, 1997).
- STREAM integrates interpretable downstream modeling by introducing a Neural Additive Topic Model (NAM) (Agarwal et al., 2021) that incorporates the documents topic-prevalences along further structural variables into an interpretable downstream regression or classification model.

2 Model Fitting and OCTIS Integration

STREAM is effectively built upon the core concepts of the OCTIS package and inherits from the *AbstractModel*, *AbstractMetric* and *OctisDataset* classes. Thus, all models, evaluation metrics, visualization functions, datasets and downstream models are perfectly integrable with all of OCTIS' models and metrics.

Datasets Creating custom datasets including tabular data is as simple as running the following few lines of code:

```
from stream.data_utils import TMDataset
df = pd.read_csv("your_data.csv")

dataset = TMDataset()
dataset.create_load_save_dataset(
    data=df,
    dataset_name="your_name",
    save_dir="save directory",
    doc_column="text", #column name where documents
     are stored
    label_column="popularity"
    )
```

All textual data is preprocessed according to the users specifications of the preprocessing pipeline and therefore, e.g., lower cased, stopwords removed and lemmatized. In the specified directory, the necessary files and a .csv file storing the tabular data are saved.

Model fitting Fitting a model (here e.g. a simple Kmeans clustering topic model) can subsequently be done simply by running the following code:

```
from stream.models import KmeansTM

model = KmeansTM(num_topics=20)
model_output = model.train_model(dataset)
```

Depending on the model, the hyperparameters can easily be adjusted. Note, that all STREAM datasets are fully usable with all OCTIS models and users can thus easily fit e.g. a LDA (Blei et al., 2003) or ETM (Dieng et al., 2020) on the TMDataset class.

Evaluation STREAM offers multiple new, intruder-word based topic evaluation metrics (Thielmann et al., 2024b) alongside classical NPMI coherence scores (Lau et al., 2014), computed over the complete documents and not over sliding windows, and also Embedding based Coherence metrics (Terragni et al., 2021b). See the Appendix for an overview over all available metrics. The evaluation of a model can thus be done by simply running:

```
from stream.metrics import ISIM
metric = ISIM(dataset)
metric.score(model_output)
```

2.1 Available Datasets

In addition to the implemented models, metrics and downstream tasks, we publicize multiple datasets suited for topic model comparison.

- Multiple **Spotify** datasets comprised of the songs' lyrics and various tabular features, such as the *popularity*, *danceability* or *acousticness* of the songs.
- A new **Reddit** dataset, which is filtered for "Gamestop" (GME) from the Subreddit "r/wallstreetbets". The data is taken from the thread "What are your moves tomorrow?". It is covering the time around the GME short squeeze of 2021.
- A new **Stocktwits** dataset also filtered for "Gamestop" (GME). It is covering the time around the GME short squeeze of 2021.
- In addition, we upload the preprocessed **Reuters** and **Poliblogs** (Roberts et al., 2018) datasets that are well suited for comparing topic model outputs.

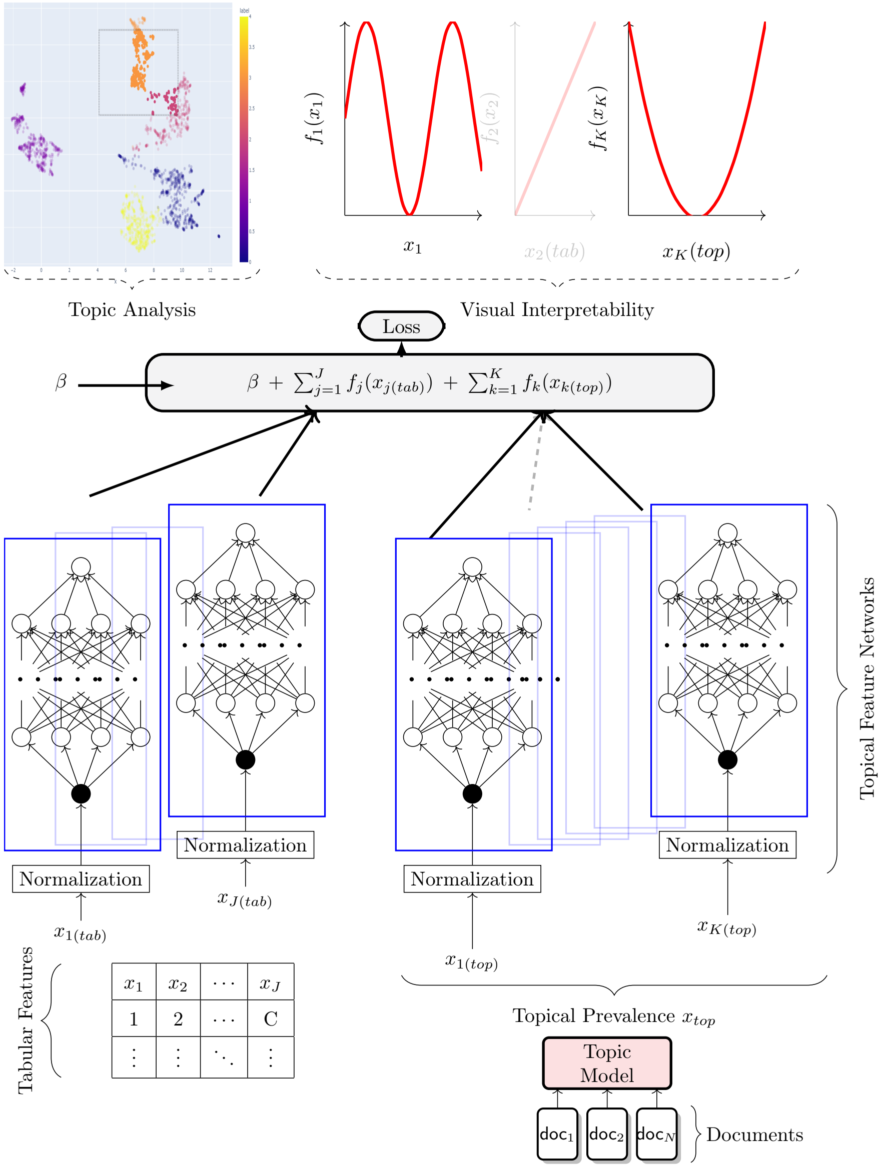

Figure 1: STREAM model architecture. After fitting a topic model, a downstream NAM can be fit and analyzed.

Table 1: Overview over preprocessed datasets that are available in STREAM. Additionally, the OCTIS datasets, *BBC-News*, *20 Newsgroups*, *M10*, *DBLP* are available.

Name	# Docs	# Words
Reuters	8,929	24,803
Reddit_GME	21,549	21,309
Poliblogs	13,246	70,726
Spotify_most_popular	4,538	53,181
Spotify_least_popular	4,374	111,738
Spotify_random	4,185	80,619
Stocktwits_GME	11,114	19,383
Stocktwits_GME_large	136,138	80,435

2.2 Topic Analysis

One of the core concepts of topic modelling is the subsequent qualitative and visual analysis of the created topics. In addition to the available topic-word-lists and matrices, STREAM implements multiple visualization methods to easily analyze the created topics. Besides classical wordclouds, the created topic clusters, topical distances, or top word distributions can be interactively visualized.

```
from stream.visuals import visualize_topic_model
visualize_topic_model(model, port=8050)
```

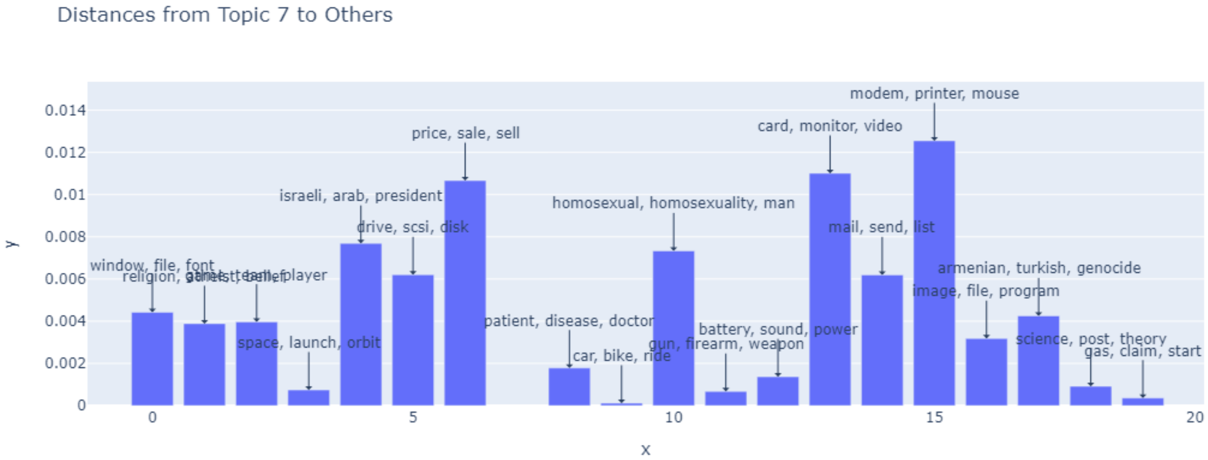

Figure 2: Topical distances of all topics towards an interactively selected topic. The distances are calculated based on topical centroids and cosine similarities in the embedding space.

3 Downstream Tasks

While the visual analysis of topics is often very helpful in analyzing a large corpus, the contents of documents often also have effects on other variables. Roberts et al. (2018) e.g. introduced a model that captures the effects of additional tabular variables on topics. STREAM offers the possibility to analyze the effects of topics and additional tabular variables on any given target variable, via implementing a downstream NAM[1]. The general form of a NAM can be written as:

$$\mathbb{E}(y) = h\left(\beta + \sum_{j=1}^{J} f_j(x_j)\right), \tag{1}$$

where $h(\cdot)$ is the activation function used in the output layer, e.g. linear activation for a simple regression task or softmax activation for a classification task. $x \in \mathbb{R}^j$ are the input features, β describes the intercept. The shape-functions are expressed as $f_j : \mathbb{R} \rightarrow \mathbb{R}$ and represent the Multi-Layer Perceptron (MLP) corresponding to the j-th feature. The model structure of a simple NAM is given in Figure 3.

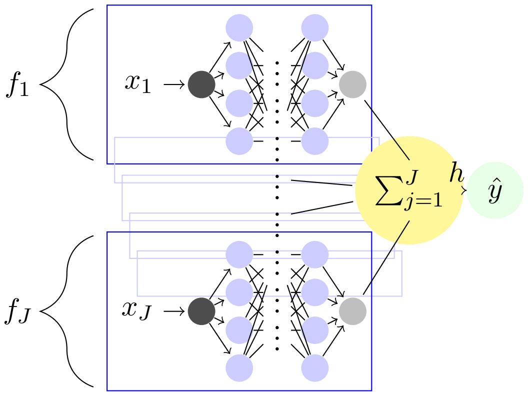

Figure 3: Architecture of a classical NAM. All features are fit independently through a Multi-Layer Perceptron and summed before the activation function and final output layer

Further, let $\boldsymbol{x} \equiv (\boldsymbol{x}_{tab}, \boldsymbol{x}_{doc})$ denote the categorical and numerical (continuous) structural features $\boldsymbol{x}_{tab}$ and $\boldsymbol{x}_{doc}$ denote the documents. After fitting a topic model (see section 2), STREAM extracts the documents topical prevalences and thus "creates" $\boldsymbol{z} \equiv (\boldsymbol{x}_{tab}, \boldsymbol{x}_{top})$, a probability vector over the documents and topics. Note, that $x_{j(tab)}^{(i)}$ denotes the j-th tabular feature of the i-th observation and $x_{k(top)}^{(i)}$ denotes document i-th topical prevalence for topic k. In order to preserve interpretability the available downstream model is given by:

$$h(\mathbb{E}\left[y\right]) = \beta + \sum_{j=1}^{J} f_j(x_{j(tab)}) + \sum_{k=1}^{K} f_k(x_{k(top)}), \tag{2}$$

Thus, the visualization of shape-function f_k shows

[1] see an example in the appendix

the impact topic k has on a target variable y and the visualization of f_j shows the impact of tabular feature j. With the given datasets and examples available in STREAM, this could represent the effect a topic created from the Spotify dataset and a songs duration have on a songs popularity. With a fitted topic model (see section 2), fitting a downstream model is straight forward leveraging the pytorch trainer class. Subsequently, all shape functions can easily be visualized similar to the plots introduced by Agarwal et al. (2021).

```python
from pytorch_lightning import Trainer
from stream.NAM import DownstreamModel

# Instantiate the DownstreamModel
downstreammodel = DownstreamModel(
    trained_topic_model=topic_model, #your trained
     topic model
    target_column='day', #specify your target column
    task='regression', #or 'classification'
    dataset=dataset,
    batch_size=128,
    lr=0.0005
)
```

```python
# Use PyTorch Lightning's Trainer to train and
     validate the model
trainer = Trainer(max_epochs=10)
trainer.fit(downstreammodel)

# Plotting
from stream.visuals import plot_downstream_model
plot_downstream_model(downstream_model)
```

4 Conclusion

In this paper, we present the STREAM framework. A user-friendly topic modeling module for creating datasets, training and evaluating topic models, visualizing results and fitting interpretable downstream models. The proposed framework is a python library and closely interacts with the existing OCTIS framework from Terragni et al. (2021a).

Future adaptations could include the integration of further more performant or e.g. distributional downstream models (Chang et al., 2022; Luber et al., 2023; Thielmann et al., 2024a) to further allow researchers to analyze the effect a topic has on a regression or classification task.

5 Limitations

We present a python package for topic modeling. While all implemented models, visualizations and the downstream models are straightforward, the actual interpretation of the results and figures is still done by the user. Given that especially textual data might include a lot of noise or harmful language, we must therefore stress the users to be careful in their final assessment of their created results. Additionally, while NAMs (Agarwal et al., 2021) offer visual interpretability, they do not allow for statistical significance as the more theoretical Generalized Additive Models (Wood, 2017) or direct causal inference.

References

Suman Adhya, Avishek Lahiri, Debarshi Kumar Sanyal, and Partha Pratim Das. 2022. Improving contextualized topic models with negative sampling. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pages 128–138, New Delhi, India. Association for Computational Linguistics.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. 2021. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711.

Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Jordan Boyd-Graber, Yuening Hu, David Mimno, et al. 2017. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. 2022. NODE-GAM: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations*.

Jen-Tzung Chien, Chao-Hsi Lee, and Zheng-Hua Tan. 2018. Latent dirichlet mixture model. *Neurocomputing*, 278:12–22.

Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. 2010. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of computer science in China*, 4(2):280–301.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Max Falkenberg, Alessandro Galeazzi, Maddalena Torricelli, Niccolò Di Marco, Francesca Larosa, Madalina Sas, Amin Mekacher, Warren Pearce, Fabiana Zollo, Walter Quattrociocchi, et al. 2022. Growing polarization around climate change on social media. *Nature Climate Change*, 12(12):1114–1121.

Andrina Granić and Nikola Marangunić. 2019. Technology acceptance model in educational context: A systematic literature review. *British Journal of Educational Technology*, 50(5):2572–2593.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

David Hall, Dan Jurafsky, and Christopher D Manning. 2008. Studying the history of ideas using topic models. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 363–371.

Timothy R Hannigan, Richard FJ Haans, Keyvan Vakili, Hovig Tchalian, Vern L Glaser, Milo Shaoqing Wang, Sarah Kaplan, and P Devereaux Jennings. 2019. Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, 13(2):586–632.

Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1):177–196.

Timo Honkela. 1997. *Self-organizing maps in natural language processing*. Ph.D. thesis, Citeseer.

Eunji Jeon, Naeun Yoon, and So Young Sohn. 2023. Exploring new digital therapeutics technologies for psychiatric disorders using bertopic and patentsberta. *Technological Forecasting and Social Change*, 186:122130.

Gillian Kant, Levin Wiebelt, Christoph Weisser, Krisztina Kis-Katos, Mattias Luber, and Benjamin Säfken. 2022. An iterative topic model filtering framework for short and noisy user-generated data: analyzing conspiracy theories on twitter. *International Journal of Data Science and Analytics*, pages 1–21.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. *Advances in Neural Information Processing Systems*, 25.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Daniel Lee and H Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

David D Lewis. 1997. Reuters-21578 text categorization collection data set.

Guiwen Liu, Juma Hamisi Nzige, and Kaijian Li. 2019. Trending topics and themes in offsite construction (osc) research: The application of topic modelling. *Construction innovation*.

Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou. 2016. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, 5(1):1–22.

Mattias Luber, Anton Thielmann, and Benjamin Säfken. 2023. Structural neural additive models: Enhanced interpretable machine learning. *arXiv preprint arXiv:2302.09275*.

Mattias Luber, Anton Thielmann, Christoph Weisser, and Benjamin Säfken. 2021. Community-detection via hashtag-graphs for semi-supervised nmf topic models. *arXiv preprint arXiv:2111.10401*.

Daniel Maier, Annie Waldherr, Peter Miltner, Gregor Wiedemann, Andreas Niekler, Alexa Keinert, Barbara Pfetsch, Gerhard Heyer, Ueli Reber, Thomas Häussler, et al. 2018. Applying lda topic modeling in communication research: Toward a valid and reliable methodology. *Communication Methods and Measures*, 12(2-3):93–118.

Tom Mitchell. 1999. Twenty Newsgroups. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5C323.

Hamed Rahimi, David Mimno, Jacob Hoover, Hubert Naacke, Camelia Constantin, and Bernd Amann. 2024. Contextualized topic coherence metrics. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1760–1773, St. Julian's, Malta. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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)*.

Arik Reuter, Anton Thielmann, Christoph Weisser, Benjamin Säfken, and Thomas Kneib. 2024. Probabilistic topic modelling with transformer representations. *arXiv preprint arXiv:2403.03737*.

Margaret Roberts, Brandon Stewart, Dustin Tingley, Kenneth Benoit, Maintainer Brandon Stewart, LinkingTo Rcpp, et al. 2018. Package 'stm'. *R Package Version*, 1(3):3.

Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1728–1736, Online. Association for Computational Linguistics.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*.

Yee Teh, Michael Jordan, Matthew Beal, and David Blei. 2004. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17.

Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. 2021a. Octis: Comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270.

Silvia Terragni, Elisabetta Fersini, and Enza Messina. 2021b. Word embedding-based topic similarity measures. In *International Conference on Applications of Natural Language to Information Systems*, pages 33–45. Springer.

Anton Thielmann, René-Marcel Kruse, Thomas Kneib, and Benjamin Säfken. 2024a. Neural additive models for location scale and shape: A framework for interpretable neural regression beyond the mean. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1783–1791.

Anton Thielmann, Arik Reuter, Quentin Seifert, Elisabeth Bergherr, and Benjamin Säfken. 2024b. Topics in the haystack: Enhancing topic quality through corpus expansion. *Computational Linguistics*, pages 1–36.

Anton Thielmann, Christoph Weisser, Thomas Kneib, and Benjamin Säfken. 2023. Coherence based document clustering. In *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*, pages 9–16. IEEE.

Anton Thielmann, Christoph Weisser, and Astrid Krenz. 2021. One-class support vector machine and lda topic model integration—evidence for ai patents. In *Soft computing: Biomedical and related applications*, pages 263–272. Springer.

Anton Thielmann, Christoph Weisser, and Benjamin Säfken. 2024c. Human in the loop: How to effectively create coherent topics by manually labeling only a few documents per class. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8395–8405.

Marah-Lisanne Thormann, Jan Farchmin, Christoph Weisser, René-Marcel Kruse, Benjamin Säfken, and Alexander Silbersdorff. 2021. Stock price predictions with lstm neural networks and twitter sentiment. *Statistics, Optimization & Information Computing*, 9(2):268–287.

Arne Tillmann, Lindrit Kqiku, Delphine Reinhardt, Christoph Weisser, Benjamin Säfken, and Thomas Kneib. 2022. Privacy estimation on twitter: Modelling the effect of latent topics on privacy by integrating xgboost, topic and generalized additive models. In *2022 IEEE Smartworld, Ubiquitous Intelligence Computing, Scalable Computing Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous Trusted Vehicles*, pages 2325–2332.

Yuli Vasiliev. 2020. *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press.

Rui Wang, Deyu Zhou, and Yulan He. 2019. Atm: Adversarial-neural topic model. *Information Processing & Management*, 56(6):102098.

Christoph Weisser, Christoph Gerloff, Anton Thielmann, Andre Python, Arik Reuter, Thomas Kneib, and Benjamin Säfken. 2023. Pseudo-document simulation for comparing lda, gsdmm and gpm topic models on short and sparse text using twitter data. *Computational Statistics*, 38(2):647–674.

Simon N Wood. 2017. *Generalized additive models: an introduction with R*. CRC press.

Hajar Zankadi, Abdellah Idrissi, Najima Daoudi, and Imane Hilal. 2023. Identifying learners' topical interests from social media content to enrich their course preferences in moocs using topic modeling and nlp techniques. *Education and Information Technologies*, 28(5):5567–5584.

A Appendix

A.1 Available Models

Multiple topic model/document clustering and subsequent topic extraction models are available in STREAM. Additionally, STREAM inherits from all models available in OCTIS. Thus, the following models are available:

Table 3: Available Models

Name	Implementation
WordCluTM	STREAM
CEDC	STREAM
DCTE	STREAM
KMeansTM	STREAM
SomTM	STREAM
CBC	STREAM
CTMneg	STREAM
TNTM	STREAM
CTM	OCTIS
ETM	OCTIS
HDP	OCTIS
LDA	OCTIS
LSI	OCTIS
NMF	OCTIS
NeuralLDA	OCTIS
ProdLDA	OCTIS

The SomTM is described in Honkela (1997). WordCluTM follows the word clustering approach introduced by Sia et al. (2020). CEDC is described in Thielmann et al. (2024b). The KMeansTM is similar to Grootendorst (2022) and often used as a fast-compute benchmark model. DCTE is a semi-supervised few-shot model introduced in Thielmann et al. (2024c). TNTM is introduced in Reuter et al. (2024). CTMneg is based on CTM (Bianchi et al., 2021) and introduced by Adhya et al. (2022). CBC is the only model of the STREAM models not based on document embeddings and focuses on coherence scores between documents, described in Thielmann et al. (2023) with adaptations from Luber et al. (2021). The neural topic models implemented in OCTIS and thus also available in STREAM are the CTM introduced by Bianchi et al. (2021), the ETM (Dieng et al., 2020), NeuralLDA and ProdLDA introduced by Srivastava and Sutton (2017). Further models are LDA (Blei et al., 2003), HDP (Teh et al., 2004), LSI (Landauer et al., 1998) and classical NMF (Lee and Seung, 2000).

A.2 Available Datasets

The available datasets are described in the paper in section 2.1. Since most of STREAMs models are centered around Document embeddings (Reimers and Gurevych, 2019), STREAM comes along with

Table 2: Comparison between STREAM and the most well-known topic modeling libraries

Features	STREAM	OCTIS	Gensim	STTM	PyCARET	MALLET	TOMODAPI
Pre-processing tools	✓	✓	✓		✓	✓	✓
Pre-processed datasets	✓	✓	✓	✓	✓	✓	✓
Pre-embedded datasets	✓						
Classical topic models	✓	✓	✓	✓	✓	✓	✓
Neural topic models	✓	✓					✓
Clustering topic models	✓						
Coherence metrics	✓	✓	✓	✓	✓	✓	
Diversity metrics	✓	✓					
Significance metrics	✓	✓					
Classification metrics	✓	✓		✓	✓	✓	✓
Intruder word metrics	✓						
Downstream Model	✓						
Visualization	✓						
Hyper-parameters tuning	BO	BO	MLE	grid-search	MLE		

a set of pre-embedded datasets. Once a user fits a model that leverages document embeddings, the embeddings are saved and automatically loaded the next time the user wants to fit any model with the same set of embeddings, thus enabling very fast model fitting and comparison.

Table 4: Dataset Overview

Name	# Docs	# Words	# Features
Reuters	8,929	24,803	-
Reddit_GME	21,549	21,309	6
Poliblogs	13,246	70,726	4
Spotify_most_popular	4,538	53,181	14
Spotify_least_popular	4,374	111,738	14
Spotify	4,185	80,619	14
Stocktwits_GME	11,114	19,383	3
Stocktwits_GME_large	136,138	80,435	3

A.3 Available Metrics

In addition to the metrics from OCTIS, STREAM offers the following available topic evaluation metrics: ISIM, INT and ISH are all intruder based metrics proposed by Thielmann et al. (2024b). Embedding Coherence is similarly implemented as by Terragni et al. (2021b) without the normalization of the embeddings. NPMI describes classical NPMI scores proposed by Lau et al. (2014) and Embedding Coherence is similar to the Coherence metrics from Terragni et al. (2021b). Expressivity and Embedding Topic Diversity are both diversity metrics calculated in the embedding space. Future developments could include e.g. metrics proposed by Rahimi et al. (2024) or Weisser et al. (2023).

- **Intruder Metrics**
  - **ISIM**: Average cosine similarity of top words of a topic to an intruder word.
  - **INT**: For a given topic and a given intruder word, Intruder Accuracy is the fraction of top words to which the intruder has the least similar embedding among all top words.
  - **ISH**: Calculates the shift in the centroid of a topic when an intruder word is replaced.
- **Diversity Metrics**
  - **Expressivity**: Cosine Distance of topics to meaningless (stopword) embedding centroid.
  - **Embedding Topic Diversity**: Topic diversity in the embedding space.
- **Coherence Metrics**
  - **Embedding Coherence**: Cosine similarity between the centroid of the embeddings of the stopwords and the centroid of the topic.
  - **NPMI**: Classical NPMi coherence computed on the source corpus.

A.4 Downstream task

As a demonstration of the downstream task, we have simulated some simple data. We have created three data generating topics, consisting of *fruits*, *vehicles* and animals. The documents are generated by having a random draw with 60% out of one specified topic and the remaining 40% out random topics. Additionally, we have generated two continuous variables and made the target variable a function of two effects of the continuous variables as well as an effect of the number of words

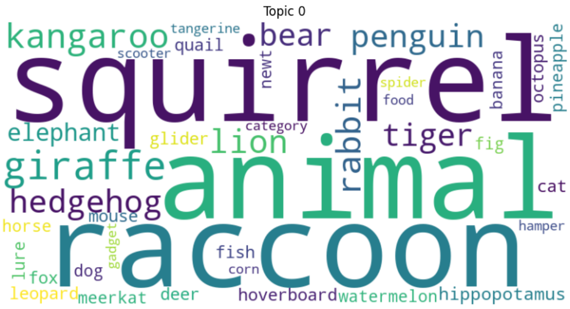

Figure 4: The *animal* topic as detected by the CDEC model and visualized via a wordcloud generating function available in STREAM.

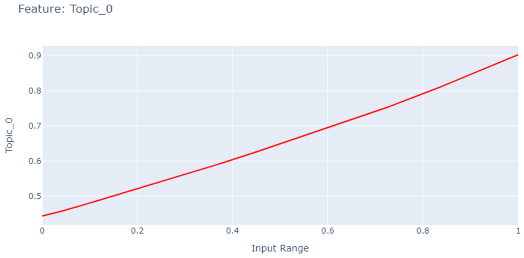

Figure 5: The effect of the *animal* topic on the target variable. Put simple: The "more" a document is about animals, the larger y gets.

taken from each generated topic. We subsequently fit a CDEC (Thielmann et al., 2024b) model and extracted the topics. The animal topic is depicted in Figure 4.

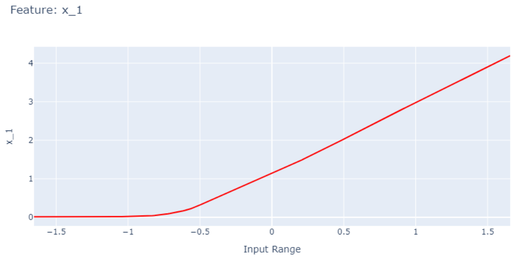

Figure 6: The numerical effect of feature x_1 on y visualized with a function available in STREAM. The visualizations closely follow the ones created by Agarwal et al. (2021).

The downstream model is then simply specified as defined in equation 2. The continuous feature effects are accurately detected and visualized in figures 6 and 7. The topic effects, one continuous 5 and one more complicated 8 are also accurately depicted. It is clearly recognizable, that the animal topic from figure 4 has a continuous positive effect on the target variable whereas the effect of the second topic is more refined and roughly follows a squared function.

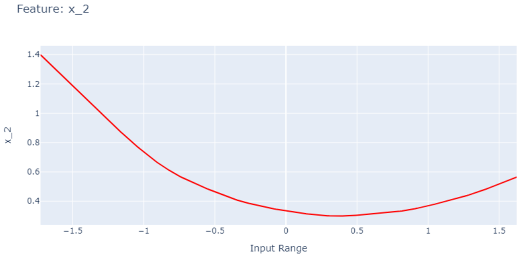

Figure 7: The numerical effect of feature x_2 on y visualized with a function available in STREAM. The visualizations closely follow the ones created by Agarwal et al. (2021).

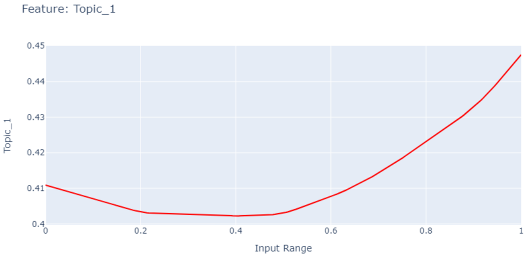

Figure 8: A more complicated topical effect of topic 1, *vehicles* on the target variable.