Concept Over Time Analysis: Unveiling Temporal Patterns for Qualitative Data Analysis

Tim Fischer[†], **Florian Schneider**[†], **Robert Geislinger**[†],
Florian Helfer[‡], **Gertraud Koch**[‡], **Chris Biemann**[†]
[†] Language Technology Group, Department of Informatics, Universität Hamburg, Germany
[‡] Institute of Anthropological Studies in Culture and History, Universität Hamburg, Germany
{florian.schneider-1 OR firstname.lastname}@uni-hamburg.de

Abstract

In this system demonstration paper, we present the *Concept Over Time Analysis* extension for the Discourse Analysis Tool Suite. The proposed tool empowers users to define, refine, and visualize their concepts of interest within an interactive interface. Adhering to the Human-in-the-loop paradigm, users can give feedback through sentence annotations. Utilizing few-shot sentence classification, the system employs Sentence Transformers to compute representations of sentences and concepts. Through an iterative process involving semantic similarity searches, sentence annotation, and fine-tuning with contrastive data, the model continuously refines, providing users with enhanced analysis outcomes. The final output is a timeline visualization of sentences classified to concepts. Especially suited for the Digital Humanities, *Concept Over Time Analysis* serves as a valuable tool for qualitative data analysis within extensive datasets. The chronological overview of concepts enables researchers to uncover patterns, trends, and shifts in discourse over time.

1 Introduction

The Discourse Analysis Tool Suite (DATS) (Schneider et al., 2023) serves as an open-source operational platform designed for conducting digital qualitative discourse analysis within the realm of Digital Humanities (DH). Developed through collaborative efforts, the tool is tailored specifically for DH researchers. Its purpose is to democratize access to cutting-edge machine learning technologies derived from Computer Vision and Natural Language Processing. This initiative enables non-expert users to efficiently handle and analyze unstructured, large multi-modal data.

While the overarching design of the platform is catered to Grounded Theory-based research (Strauss and Corbin, 1990), particularly in alignment with approaches like the Sociology of Knowledge Approach to Discourse (Keller, 2011), numerous features extend its utility across various disciplines. Noteworthy functionalities include the automated pre-processing of multi-modal data (text, image, audio, and video), data exploration, and both manual and automatic annotation.

In this paper, we extend DATS with *Concept Over Time Analysis*, a machine-learning-based feature to identify concepts of interest in a large dataset and analyse their occurrences over time. The proposed system allows users to define concepts, assists them in identifying relevant sentences, refine the concept representation and finally visualize the results over time. The interactive User Interface (UI) follows the Human-in-the-loop paradigm (Holzinger, 2016), encouraging users to give feedback by annotating samples, which improves concept representations and analysis results.

The proposed system performs few-shot sentence classification, mapping each concept to a class. We use Sentence Transformers (Reimers and Gurevych, 2019) to compute representations of sentences and concept descriptions. Initially, a similarity search provides semantically relevant sentences for each concept that can be classified into concepts by the user. With few labeled examples, a contrastive dataset is generated to fine-tune a Sentence Transformer model. This is employed to update concept representations and similarity search results, providing the user with better suggestions. This iterative refinement step is repeated until the user is content. Finally, sentences classified to a concept with confidence above a specified threshold are visualized in a timeline.

The *Concepts Over Time Analysis* enables qualitative data analysis within large datasets. A chronological overview of custom concepts like events, topics, opinions or practices allows researchers to identify patterns, trends, and shifts in discourse over time. This temporal perspective can reveal how ideas, narratives, or cultural practices evolve, providing insights into the dynamics of discour-

sive trends. By visualising such concepts along a timeline, researchers can identify clusters of related events, helping to create a nuanced understanding of how cultural phenomena unfold over time. Further, a timeline analysis can aid in contextualising and connecting different elements within a large dataset, making it easier to trace the lineage of ideas or the influence of past discourses on current ones. Though inspired by DH and discourse analysis, the proposed system is versatile, extending its utility to related analyses such as visualizing topics or sentiments over time.

While the *Concepts Over Time Analysis* is a powerful feature on its own, its integration within DATS can enable users to merge findings with existing analyses, utilize Memos or the Logbook for documentation and reflection, and identify concepts to be analyzed using search and filter functionalities.

The contributions of this paper are threefold: First, we develop the *Concept Over Time Analysis* extension, which facilitates the identification of concepts in large datasets and a chronological overview. Second, we describe a typical usage scenario, outlining the interaction with the system. Finally, we perform experiments on three datasets, demonstrating the feasibility of our system.

The DATS and *Concepts Over Time Analysis* are open-source and available on GitHub[1] where links to a demonstration and a video are provided.

2 Related Work

Qualitative Data Analysis tools that are frequently used in the DH include CATMA (Gius et al., 2022), MAXQDA and Atlas.ti. While offering various analysis tools, at the time of writing, none of them include means to semi-automatically identify concepts in large datasets or visualize concepts in a chronological overview. With the proposed extensions, DATS aims to close that gap.

However, there exist many tools and libraries that offer interesting timeline analysis functionalities. BERTopic (Grootendorst, 2022), a topic modeling tool, leverages Sentence Transformers to compute document or sentence embeddings, yielding easily interpretable topic representations through clustering. With Dynamic Topic Modeling, the tool offers a collection of techniques to analyse the development of topics over time. While BERTopic identifies prevalent topics automatically, our extension empowers users to define and analyze their own concepts or topics of interest across time.

SCoT (Haase et al., 2021) is an interactive web application to analyse the sense-clusters of a word and their evolution over time. SentiView (Wang et al., 2013) is an interactive visualization system for sentiment analysis. It visualizes changes over time of various attributes and relationships between demographics of interest as well as participants' sentiments on popular topics. Open Discourse is a web-platform for the analysis of the plenary minutes of the german federal parliament. It includes a topic analysis feature to investigate the political discourse over time, filterable by speaker, gender, party and other attributes.

LabelSleuth (Shnarch et al., 2022) serves as a no-code platform, making NLP accessible for a broad audience. It facilitates integrated model training and an intuitive, active learning-powered annotation interface. In contrast, our proposed extension supports the training of multi-class sentence classification models and enables a timeline analysis of the results.

3 Concept Over Time Analysis

The *Concept Over Time Analysis* extension is a interactive, machine-learning-based tool that semi-automatically identifies relevant sentences of user-defined concepts, adjusts accordingly to feedback, and visualizes concept occurrences in a timeline.

Its goal is to provide users with a chronological overview of their concepts of interest, with as few steps as possible. Integrating the feedback process into the workflow was a key requirement for the development of the UI.

3.1 Concepts

Concepts are defined by Strauss et al. (1996) as the basic building blocks of a theory. In grounded theory, open coding represents the analytical process through which concepts are identified and developed in terms of their properties and dimensions.

In this work, we employ the broad term *concept* to encompass various usage scenarios, such as categories, topics, sentiments, opinions, etc. In our terms, a concept refers to anything that serves to semantically group or classify sentences.

3.2 User Workflow

The *Concept Over Time Analysis* extension provides versatile functions for diverse applications. Illustratively, we present key features through a

[1] github.com/uhh-lt/dwts

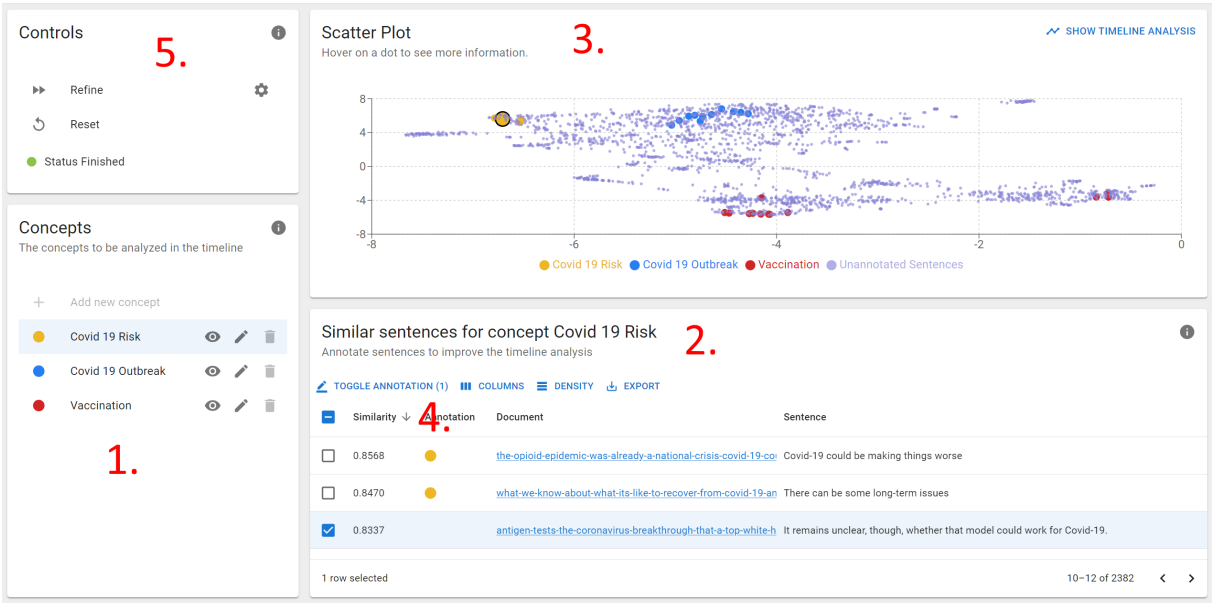


Figure 1: The annotation view of the *Concept Over Time Analysis* extension: (1) Define, edit, delete & toggle visibility of concepts. (2) + (4) Sentence Annotator: A ranked list sentences similar to the selected concept. (3) 2D visualization of the search space, annotated sentences are colored & highlighted. (5) Controls to manually start, reset & configure model training. Training status is updated live.

typical workflow. Consider Alice, a researcher exploring discourse on the COVID pandemic from 2020 to 2024. Using her web browser, she accesses DATS, registers, logs in, and creates a new project. Uploading material from various news websites, she ensures the presence of metadata indicating the publication date, which is required for *Concept Over Time Analysis*. To start her analysis, Alice locates the feature in the "Analysis" tab and creates an empty *Concept Over Time Analysis*.

Defining concepts Alice is presented with the annotation view of the *Concept over time analysis* view (see Figure 1). To initiate the analysis, she defines three concepts "Covid 19 as a risk", "Covid 19 Outbreaks", "Vaccinations", and provides a short description for each (1). To identify interesting concepts, DATS offers various dataset exploration features ranging from the distant, quantitative search, filter and statistics features to the close, qualitative reading and annotation of documents.

Concept annotation After inputting concepts and descriptions, Alice starts the first iteration of the analysis. The Sentence Annotator (2) displays ranked sentences based on similarity to the concept, accompanied by a scatter plot (3) visualizing all search space sentences. Clicking on a dot navigates to and highlights the corresponding sentence in the annotator. While exploring the search space, Alice realizes that the scatter plot lacks effective clustering for her concepts, primarily because, at this stage, a concept's representation relies solely on the initial description.

Alice enhances her defined concepts with more information by providing feedback through sentence annotations. While scrolling through lists of similar sentences, she annotates sentences with the respective concept (4), identifying fitting mentions, ideas, or paraphrases. This initial annotation is recognized as the most challenging part of the analysis, and we aim to assist by providing an initial similarity ranking.

Iterative concept refinement After annotating the minimum required sentences per concept (default is five), she proceeds to the next step by clicking the refine button (5). Through her feedback, the system improves its ability to distinguish between concepts and provides more relevant similar sentences for each. The scatter plot visualization is updated, showing clusters of sentences centered around the annotated ones. Annotated sentences are in Alice's defined color, while others are visualized in purple (3). With the improved scatter plot and sentence rankings, it becomes notably easier for Alice to identify additional relevant sentences for her concepts.

After annotating a few more relevant sentences, Alice is prompted to refine the Concept Over Time

Analysis again. Alice can continue this process until she's certain in the system's ability to distinguish between concepts. Typically, this is achieved if the scatter plot exhibits clear clustering, allowing for effective separation, and if most top similar sentences for a concept are deemed relevant.

Timeline analysis Satisfied with the clustering and the suggested similar sentences, Alice switches to the Timeline Analysis visualization (see Figure 2). This visualization (1) depicts the date on the x-axis and the aggregated count of similar sentences per concept on the y-axis, illustrating the development of her defined concepts over time. Alice customizes the aggregation (year, month, day) and adjusts the similarity threshold to filter out irrelevant sentences (2). Alice explores the results by clicking on several dots, which reveals all sentences of the selected concept on the chosen date in the Sentence Annotator (3).

4 System Implementation

The visualization is implemented using React and the Recharts[2] library. The asynchronous model training is implemented with Celery[3] and SetFit[4] (Tunstall et al., 2022).

The main challenges involve classifying concepts and adapting to limited user feedback while ensuring high performance and interactivity. Additionally, we noted that concepts are often nuanced and exist within the same domain, resulting in high inter-concept similarity. Therefore, key requirements considered during the implementation are fast training, the capacity to learn from a few labeled samples, and the ability to distinguish between topically similar instances.

The *Concept Over Time Analysis* workflow is internally structured as a three-step pipeline, encompassing: establishing the initial search space, fine-tuning the Sentence Transformer model, and computing the results. In the subsequent sections, we provide a detailed explanation of each step.

The initial search space To initiate the *Concept Over Time Analysis*, users provide concepts along with proper descriptions. The descriptions are embedded with a Sentence Transformer model to generate an initial representation of each concept. The model is configurable during the tool setup and defaults to a pretrained multilingual CLIP (Radford et al., 2021; Reimers and Gurevych, 2020) model.

Following this, the concept representations are employed to identify semantically similar sentences, leveraging the existing semantic similarity search feature within DATS. Each uploaded document runs through a multi-step pre-processing pipeline, including sentence splitting and sentence embedding, with the results stored in Weaviate[5], an open-source vector database. Consequently, the vector database is utilized to retrieve the top K similar sentences for each concept representation, along with their respective similarity scores. K is configurable in the UI and defaults to 1000.

The set of returned sentences is considered as the search space, which remains constant in the subsequent steps. However, the UI allows resetting the search space and restarting the analysis process.

The initial concept descriptions play a pivotal role in the entire analysis, determining the sentences considered during the analysis. This step is only executed once at the beginning, but crucial to the process as it limits the sentences to the concepts' domain and fastens the following steps.

Model fine-tuning Users provide feedback in terms of labeling sentences with concepts. This step is skipped, if there are insufficient annotations for any given concept. The minimum labeled examples per concept can be configured in the UI, however, our experiments suggest that 4, 8, and 16 are good thresholds to start the fine-tuning.

A contrastive training dataset is generated based on the users' annotations, which is then used to fine-tune a pre-trained Sentence Transformer model (more details in Section 5). The model trains asynchronously in the background for 1 epoch, the frontend is informed about status changes and displays them accordingly.

Results In the final step, we compute a 2D representation of the search space, update concept representations, re-rank sentences, and compute the timeline analysis.

To generate the 2D representation, displayed in the annotation view, the fine-tuned model computes sentence embeddings, which are then passed to UMAP (McInnes et al., 2020) or t-SNE (Van der Maaten and Hinton, 2008) for dimensionality reduction. Initially, the embeddings stored in the vector database represent the search space sentences. Throughout iterative refinement, models with im-

[2] https://recharts.org/

[3] https://docs.celeryq.dev/

[4] https://huggingface.co/docs/setfit

[5] https://weaviate.io/

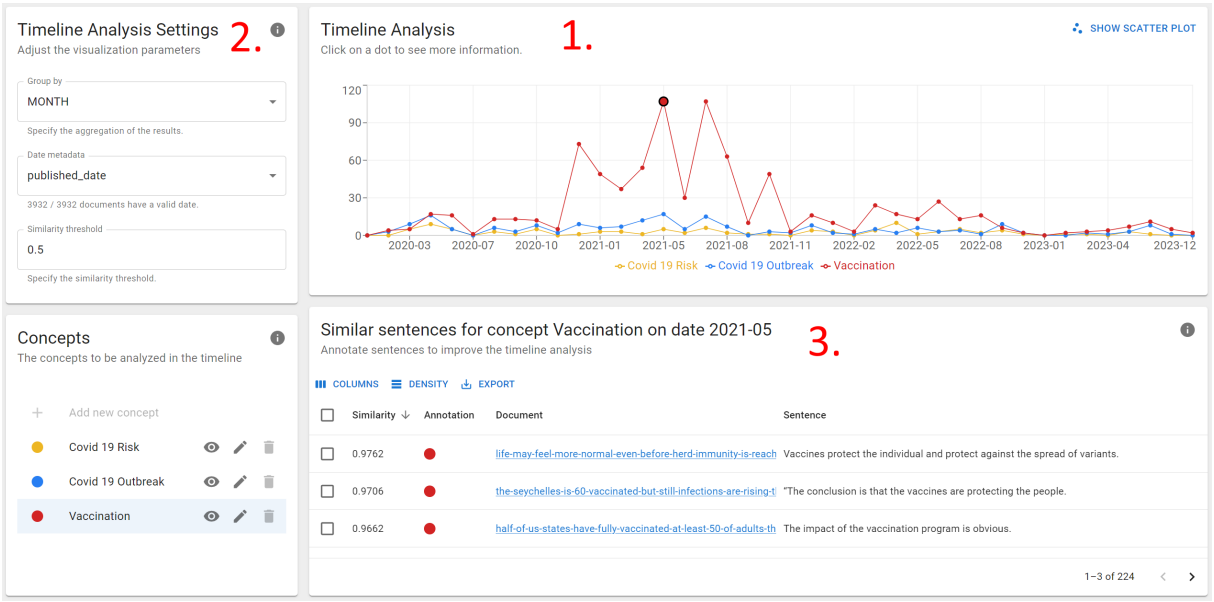


Figure 2: The timeline view of the *Concept Over Time Analysis* extension: (1) Aggregated occurrences of sentences similar to concepts over time. (2) Settings. (3) Sentence Annotator showing similar sentences of selected concept at selected date.

proved performance enhance the sentence representations, causing sentences to "move" toward their most similar concept and form clusters. Appendix B illustrates this process.

Initially, a concept is represented by its embedded description. An enhanced representation is derived by averaging the embeddings of labeled sentences for the concept. Next, we calculate the cosine similarity between concept representations and all embedded search space sentences. The sentences in the Sentence Annotator are then ranked based on the resulting similarity scores.

Finally, the timeline analysis is computed. For each concept, sentences below the similarity threshold are filtered out, and the remaining ones are aggregated and counted based on the specified time intervals (year, month, or day).

5 Experiments

Few-shot fine-tuning of Sentence Transformers has been shown to achieve high accuracy with few labeled data by Tunstall et al. (2022). Our following experiments assess the suitability of this approach for the *Concept Over Time Analysis* extension, while also gaining insights into its limitations.

We formulate the following requirements that our system should meet: Require few labeled samples, enabling users to quickly achieve satisfactory results. Ensure short training time to facilitate an interactive user experience with fast iterations. Improve performance with more training samples rather than more training time, allowing iterative refinement of the model.

5.1 Datasets

The 20 newsgroup dataset[6] (NG20), containing around 20,000 newsgroup posts across 20 topics, aligns with our users' domain. Given that concepts are commonly closely related and within the same domain, we align this experiment with the expected real-world setting by considering the classes "misc," "guns," and "mideast" in the politics subject. Similarly, we consider the related classes "Wellness", "Style & Beauty" and "Healthy Living" of the News Category dataset (Misra, 2022) (NCT). It contains around 210.000 news headlines from 2012 - 2022.

The Stanford Sentiment Treebank dataset (SST-5) (Socher et al., 2013), containing approximately 12,000 sentences from movie reviews, involves fine-grained sentiment classification with five labels ranging from "very negative" to "very positive." We explore this dataset to assess the applicability of our proposed system for Sentiment Analysis, a common need in DH research.

5.2 Setup

Following Tunstall et al. (2022), we fine-tune a Sentence Transformer model with few-shot training

[6] http://qwone.com/ jason/20Newsgroups/

data and train a Logistic Regression classifier on top. We use `paraphrase-mpnet-base-v2`[7] for all experiments.

The contrastive training data comprises sentence pairs, where a pair is deemed positive (1.0) if both sentences belong to the same class and negative (0.0) otherwise. Considering C classes with N sentences per class, all combinations $(N \times |C|)^2$ are evaluated, but duplicate $(A, B) = (B, A)$ and identical (A, A) sentence pairs are removed. To balance positive and negative pairs, positive examples are oversampled, resulting in a substantial training set even with limited labeled samples.

We mimic the iterative concept refinement step, by incrementally increasing the labeled training data. However, our experiments maintain a uniform label distribution in the labeled data, a condition not assured in real-world settings.

The model is fine-tuned for 1 epoch with cosine similarity loss, utilizing the AdamW optimizer with default parameters and a batch size of 16. Experiments are conducted on a single A100 GPU.

5.3 Evaluation

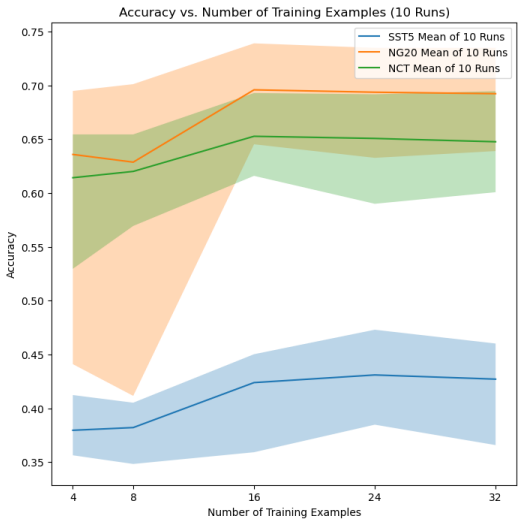


Figure 3: Accuracies over 10 runs of the experiments described in Section 5.

Figure 3 shows averaged accuracy from 10 runs of the SST-5, NG20 and NCT experiments. For all three experiments, we observe a steady improvement in accuracy up to 16 training examples per class. Labeling more examples does not improve the results. Additionally, we note that the quality of annotation significantly impacts our setup. In the 20 newsgroup experiment, there is a difference of over 30% accuracy between the best-performing and worst-performing run.

We acknowledge that the performance of our approach does not match state-of-the-art results on the datasets due to the few-shot nature of these experiments. To mitigate this, in the timeline analysis, users can define a similarity threshold to filter out irrelevant results. In addition, we provide an interface to assess the considered sentences for a given concept and time.

In summary, our implemented system achieves most of our requirements: It requires few labeled training data, the training times are fast (see Appendix A), and it scales with more training samples, but only up to 16 examples per class.

6 Conclusion

In this paper, we presented the *Concept Over Time Analysis* extension, a machine-learning-based tool facilitating the identification and analysis of user-defined concepts within large datasets over time. The proposed tool empowers users to define concepts, identify relevant sentences, and visualize results in a timeline through a unified and interactive interface that is fully integrated within the Discourse Analysis Tool Suite. Embracing the Human-in-the-loop paradigm, the system iteratively refines the underlying model and enhances the timeline analysis with the help of user feedback.

It employs few-shot sentence classification utilizing Sentence Transformer models for sentence and concept representations. With minimal human feedback, the model is fine-tuned on a contrastive dataset generated from the provided annotations.

This results in a powerful tool for qualitative data analysis in large datasets. The chronological overview of custom concepts allows researchers to identify patterns, trends, and shifts in discourse over time. Beyond discourse analysis, the extension supports various applications, including visualizing topics or sentiments over time.

This work centered on analyzing and visualizing concepts in texts over time. In the next iteration, we plan to support image data, i.e., using cross-modal Sentence Transformers powered by CLIP, and adapting the interface for image processing. We also aim to fine-tune Sentence Transformers with adapters to further reduce training time while maintaining comparable performance.

[7] https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2

7 Acknowledgement

This work is supported by the D-WISE project, Universität Hamburg, funded by BMBF (grant no.01UG2124).

8 Ethics Statement

The proposed *Concept Over Time Analysis* extension of the DATS heavily relies on Machine Learning (ML) models and technology. We acknowledge that while ML models can provide valuable insights, they are not perfect and may produce errors. It is crucial to understand and accept the limitations of these models to avoid drawing false impressions or conclusions based on their outputs. We urge users to exercise caution and critical thinking when interpreting results generated by our tool. The accuracy and reliability of any findings depend heavily on the quality and appropriateness of the input data as well as the model's assumptions and parameters. Users should also consider potential biases that may exist within the training data used to develop the models. In addition, we emphasize the importance of validating all results through alternative methods such as manual coding or expert review. This step will help ensure the accuracy and robustness of the findings and prevent overreliance on automated tools.

References

Evelyn Gius, Jan Christoph Meister, Malte Meister, Marco Petris, Christian Bruck, Janina Jacke, Mareike Schumacher, Dominik Gerstorfer, Marie Flüh, and Jan Horstmann. 2022. CATMA: Computer Assisted Text Markup and Analysis.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Christian Haase, Saba Anwar, Seid Muhie Yimam, Alexander Friedrich, and Chris Biemann. 2021. SCoT: Sense Clustering over Time: a tool for the analysis of lexical change. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 198–204, Online. Association for Computational Linguistics.

Andreas Holzinger. 2016. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131.

Reiner Keller. 2011. The sociology of knowledge approach to discourse (SKAD). *Human Studies*, 34:43–65.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*.

Rishabh Misra. 2022. News Category Dataset. *arXiv preprint arXiv:2209.11429*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, Online.

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)*, pages 3973–3983, Hong Kong, China.

Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online.

Florian Schneider, Tim Fischer, Fynn Petersen-Frey, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2023. The D-WISE Tool Suite: Multi-Modal Machine-Learning-Powered Tools Supporting and Enhancing Digital Discourse Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 328–335.

Eyal Shnarch, Alon Halfon, Ariel Gera, Marina Danilevsky, Yannis Katsis, Leshem Choshen, Martin Santillan Cooper, Dina Epelboim, Zheng Zhang, Dakuo Wang, Lucy Yip, Liat Ein-Dor, Lena Dankin, Ilya Shnayderman, Ranit Aharonov, Yunyao Li, Naftali Liberman, Philip Levin Slesarev, Gwilym Newton, Shila Ofek-Koifman, Noam Slonim, and Yoav Katz. 2022. Label Sleuth: From Unlabeled Text to a Classifier in a Few Hours. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Anselm Strauss and Juliet Corbin. 1990. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. SAGE Publications, Inc.

Anselm Strauss, Juliet Corbin, Solveigh Niewiarra, and Heiner Legewie. 1996. *Grounded Theory: Grundlagen Qualitativer Sozialforschung*. Beltz, Psychologie-Verlag-Union Weinheim.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts. *arXiv preprint arXiv:2209.11055*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Changbo Wang, Zhao Xiao, Yuhua Liu, Yanru Xu, Aoying Zhou, and Kang Zhang. 2013. SentiView: Sentiment Analysis and Visualization for Internet Popular Topics. *IEEE Transactions on Human-Machine Systems*, 43(6):620–630.

A Few-Shot Training Runtime

An important requirement for implementing usable tools is the time or latency. We measured the training times for the experiments described in Section 5 to get an estimate of the time users have to wait until they receive the results of the COTA Refinement in the UI. The results are shown in Figure 4. Here we can see that the runtime heavily depends on the lengths of the sentences (see Table 1) and the number of samples used for training. Since the sentences within our tool are created using spaCy[8], we can assume typical lengths. Hence, we expect the users of our tool to wait a maximum of two minutes when refining a COTA with 32 annotations per concept.

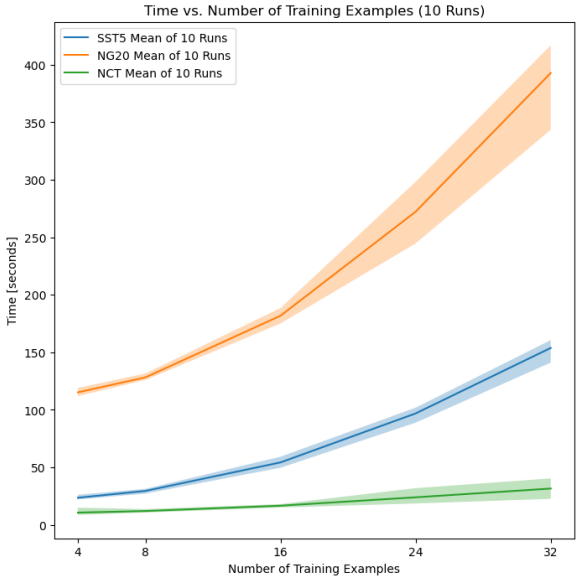


Figure 4: Average runtimes over 10 runs of the experiments described in Section 5.

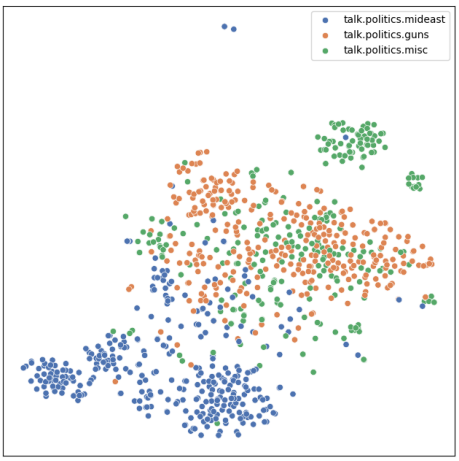
Table 1: Number of white space separated words per dataset.

Dataset	mean	std	min	max
NG20	205.48	652.55	1.00	20082.00
SST-5	19.14	9.31	2.00	52.00
NCT	9.15	3.25	1.00	38.00

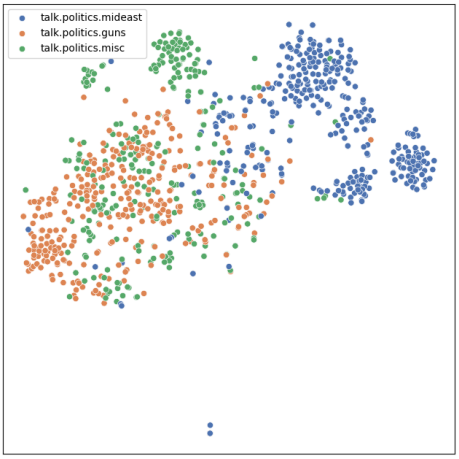
B Sentence Embedding Evolution

In this appendix section, we show the evolution of sentence embeddings produced during our experiment with the NG20 dataset. We use t-SNE (Van der Maaten and Hinton, 2008) to reduce the embeddings to 2D for better visualization.

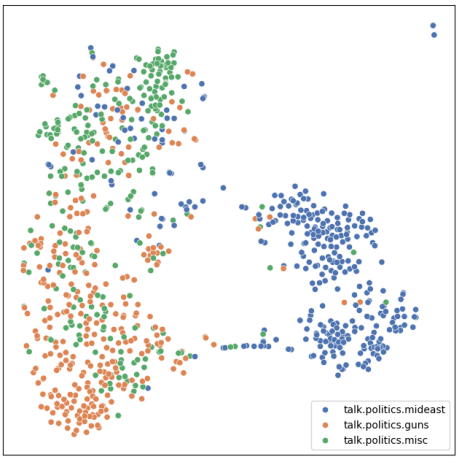
[8] https://spacy.io/



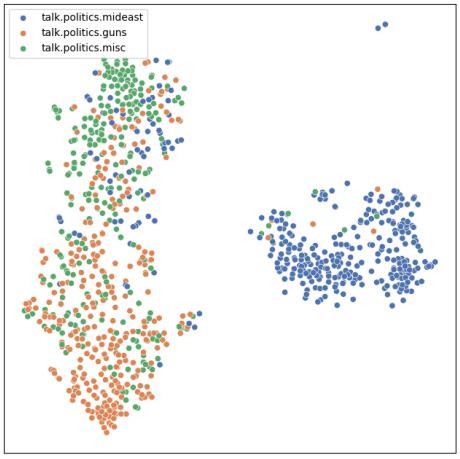
(a) Number of training samples per class: 4



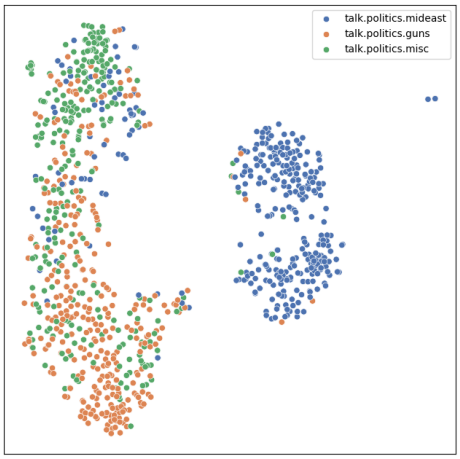
(b) Number of training samples per class: 8



(c) Number of training samples per class: 16



(d) Number of training samples per class: 24



(e) Number of training samples per class: 32

Figure 5: The evolution of sentence embeddings produced during our experiment with the NG20 dataset.