The D-WISE Tool Suite: Multi-Modal Machine-Learning-Powered Tools Supporting and Enhancing Digital Discourse Analysis

Florian Schneider[*†], **Tim Fischer**[*†], **Fynn Petersen-Frey**[†]
Isabel Eiser[‡], **Gertraud Koch**[‡], **Chris Biemann**[†]
[*] equal contributions
[†] Language Technology Group, Department of Informatics, Universität Hamburg, Germany
[‡] Institute of Anthropological Studies in Culture and History, Universität Hamburg, Germany
{florian.schneider-1, firstname.lastname}@uni-hamburg.de

Abstract

This work introduces the D-WISE Tool Suite (DWTS), a novel working environment for digital qualitative discourse analysis in the Digital Humanities (DH). The DWTS addresses limitations of current DH tools induced by the ever-increasing amount of heterogeneous, unstructured, and multi-modal data in which the discourses of contemporary societies are encoded. To provide meaningful insights from such data, our system leverages and combines state-of-the-art machine learning technologies from Natural Language Processing and Computer Vision. Further, the DWTS is conceived and developed by an interdisciplinary team of cultural anthropologists and computer scientists to ensure the tool's usability for modern DH research. Central features of the DWTS are: a) import of multi-modal data like text, image, audio, and video b) preprocessing pipelines for automatic annotations c) lexical and semantic search of documents d) manual span, bounding box, time-span, and frame annotations e) documentation of the research process.

1 Introduction

In today's digital era, ever-increasing amounts of heterogeneous, unstructured, and multi-modal data are ubiquitous. Within this data, discourses of contemporary societies are included in various forms, such as news articles or videos, social media postings, forum threads, memes, podcasts, or TV shows. This induces an issue for Digital Humanities (DH) researchers when conducting digital qualitative discourse analysis (Keller, 2011) with such data to examine complex sociological patterns and discussions: It becomes infeasible for a researcher or an average research group to investigate the data manually so they rely on computer assisted qualitative data analysis software (CAQDAS). Although there are many such tools (Eckart de Castilho et al., 2016; Gius et al., 2022; Shnarch et al., 2022; Schneider et al., 2023), they often lack support for such (amounts of) data or are proprietary software.

With the D-WISE Tool Suite (DWTS) introduced in this work, we provide a novel working environment to support and enhance digital qualitative discourse analysis. The tool is conceived and developed within the D-WISE[1] project in close co-creation by an interdisciplinary team of cultural anthropologists and computer scientists to ensure the tool's usability for modern DH research.

Our system relies on recent advances in Natural Language Processing and Computer Vision and their combination to address the challenges of large amounts of heterogeneous and multi-modal data. Specifically, we employ state-of-the-art text (Devlin et al., 2019; Reimers and Gurevych, 2019), vision (Zhu et al., 2021; Li et al., 2023), speech (Radford et al., 2022), video (Ni et al., 2022; Tong et al., 2022), and visio-linguistic models (Radford et al., 2021) in our multi-modal preprocessing pipeline for tasks like named entity recognition, multi-modal similarity search, object detection, image captioning, automatic speech recognition (transcription), and video understanding tasks. Other essential functionalities for digital discourse analysis like lexical search or manual annotations, are also supported by the DWTS.

In this paper, we describe the system architecture and central features of the DWTS: (a) import of text, image, audio, and video documents; (b) preprocessing pipelines for automatic annotations and indexing; (c) lexical and semantic similarity search, (c) manual annotations; (d) automatic and manual documentation of the research process. The DWTS is designed as an extensible and scaleable open-source software system and is publicly available on GitHub[2]. Links to a demonstration instance[3] and a video[4] are provided.

[1] https://www.dwise.uni-hamburg.de

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

[3] acl-dwts.ltdemos.informatik.uni-hamburg.de

[4] https://youtu.be/NEmq4AMXVss

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 328–335
July 10-12, 2023 ©2023 Association for Computational Linguistics

2 Related Work

In the following, we discuss the features and limitations of popular open-source CAQDAS and general purpose annotation tools concerning qualitative discourse analysis on large heterogenous, multi-modal, and multi-lingual document collections.

WebAnno (Eckart de Castilho et al., 2016), CodeAnno(Schneider et al., 2023), INCePTION (Klie et al., 2018) are a family of web-based collaborative platforms for linguistic annotations based on the UIMA framework. While they support a wide range of annotation-related features for text documents, they do not support image, audio, or video material. Although there exists a WebAnno extension (Remus et al., 2019) for multi-modal document annotations, the tool's objective is annotating relatively small corpora. Further, the platforms do not include any deep learning technology or preprocessing steps for automatic annotations. Moreover, all platforms lack an overview of the document collection and general search functionality, making it challenging to apply for discourse analysis projects. Another popular tool is CATMA (Gius et al., 2022). Although the web-based platform is designed for qualitative and quantitative research, including annotation, analysis, and visualization functionalities, it only supports text documents. Other CAQDAS systems involving deep learning techniques for automatic annotations are LabelSleuth (Shnarch et al., 2022) and TextAnnotator (Abrami et al., 2020). However, both tools only support textual documents. Recogito2 (Simon et al., 2017) is an open-source tool that can be used for collaborative research projects, is applicable for discourse analysis, and supports text and image annotations. However, the tool does not implement preprocessing pipelines and is designed for small document collections. The widely used CAQDAS tools MAXQDA and Atlas.ti fulfill almost all considered criteria but are proprietary closed-source software, and require expensive licenses, what often poses a hurdle of academic research groups. Further, the tools do not support advanced deep learning features like similarity search, object detection, or automatic audio and video transcriptions.

3 System Demonstration

The D-WISE Tool Suite is a web-based working environment for digital qualitative discourse analysis. It is designed to handle multi-lingual, multi-modal material of large corpora, achieved by a scaleable architecture built upon a unified data model and state-of-the-art machine learning technology. The application is developed for research groups featuring collaborative projects with multiple users and role management. Further, the system is easily deployable using Docker, not requiring the installation of additional third-party software. Generated data can be exported in common formats for further analysis or processing with other tools.

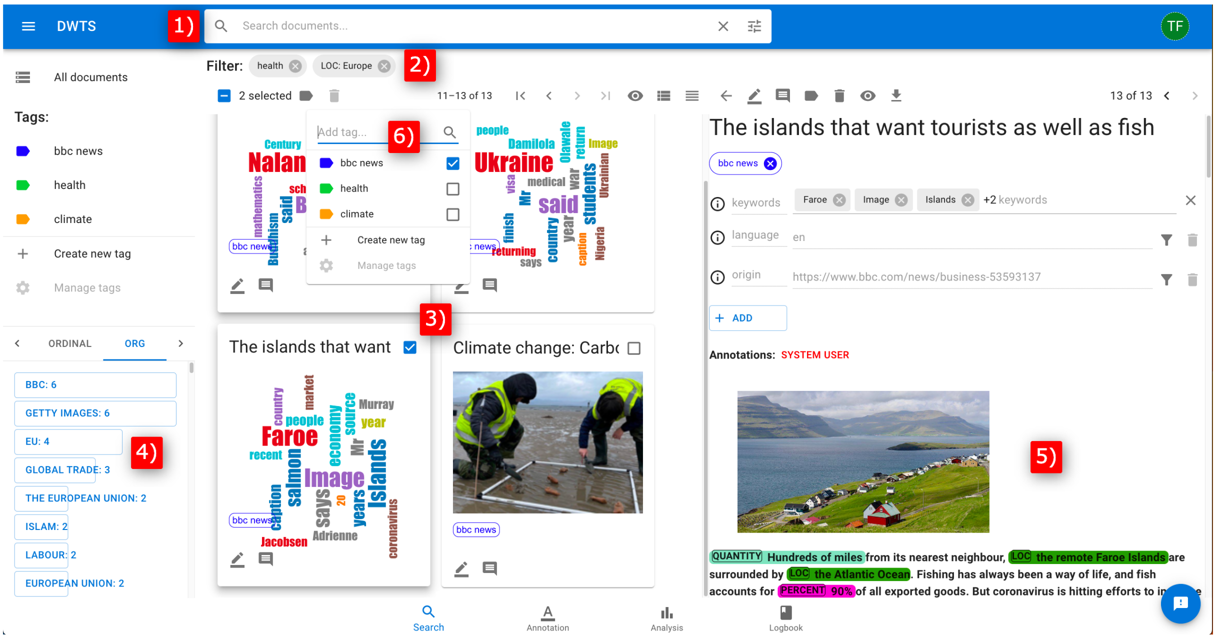
3.1 Typical Workflow

The D-WISE Tool Suite has a variety of functions that can be used in many diverse ways. For illustrative purposes, we demonstrate the key features with a typical workflow. Imagine Alice, a researcher who examines the discourse on electronic health apps. She accesses DWTS using her web browser and is greeted by a login screen. After registering and logging in, she creates a new project for herself and her project partners.

Data Import To kick off the project, she uploads her material as single files and in a ZIP archive. The data comes in various formats and languages: She saved relevant websites as HTML comprising text and images, related PDF and DOCX documents, and articles in raw text files. Further, she found relevant videos and podcasts on the matter. The DWTS can handle most data formats for text, image, audio, and video documents. Additionally, DWTS offers an external crawler implemented with Scrapy and Beautifulsoup to scrape websites in case additional material is required.

Data Preprocessing When a document is uploaded, it is automatically pre-processed by the DWTS pipeline. It comprises ML-powered steps to extract metadata and enrich the material with additional information like annotations for named entities or objects in images and videos. It handles vast quantities of multi-modal, multi-lingual data as explained in detail in Section 4.1. While the import operation runs in the background, Alice enjoys a coffee until all data has been processed.

Data Exploration Once the process has finished, Alice starts her research by exploring the data (see Figure 1). The DWTS offers traditional search methods such as full text and keyword search 1). Documents can be further filtered by manually and automatically created annotations, codes and tags 2). During the exploration, Alice can overview the



The islands that want tourists as well as fish

keywords: Faroe, Image, Islands, +2 keywords

language: en

origin: https://www.bbc.com/news/business-53593137

Annotations: SYSTEM USER

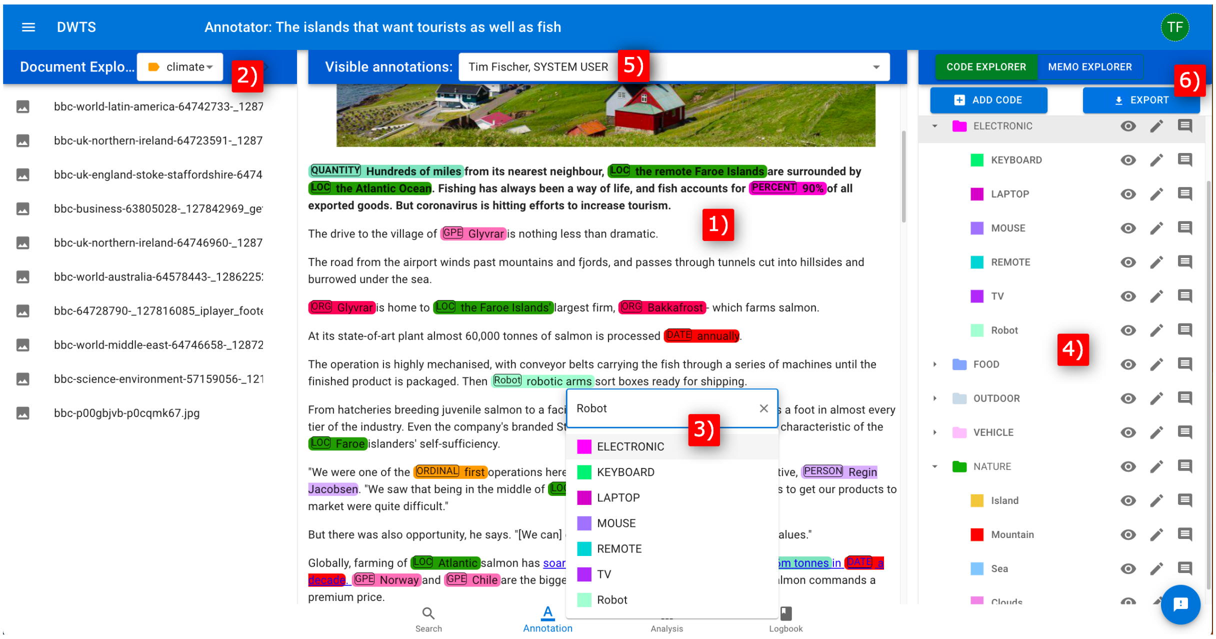
QUANTITY Hundreds of miles from its nearest neighbour, LOC the remote Faroe Islands are surrounded by LOC the Atlantic Ocean. Fishing has always been a way of life, and fish accounts for PERCENT 90% of all exported goods. But coronavirus is hitting efforts to in e

Figure 1: Components of the DWTS search interface: 1) Search bar for lexical search and filter options; 2) Currently applied filters; 3) Multi-modal search results; 4) Search results statistics; 5) Document viewer with tags, metadata, and annotations; 6) Tag editor popup. *The screenshot has been optimized for demonstration purposes.*

current search results 3) by referring to the statistics panel 4) that lists the most frequent keywords, tags, and entities in these documents. Clicking on one of the documents opens a reader view on the right panel 5), which can display text, image, audio, video, or mixed documents. Here, both manual and automatic annotations, e.g., text passages or objects in images or videos, are highlighted. Metadata associated with the document can be viewed and edited as well. This view also allows for multimodal semantic search by right-clicking on images, videos, or sentences. As documents of all modalities are represented in the same embedding space, the semantic search can be utilized to retrieve similar documents of the same or different modality. Using these functionalities, Alice found several relevant documents related to her research questions. She creates and applies a tag 6) to mark the documents and save them in a collection.

Annotation / Coding Having found relevant documents, Alice decides to read them in detail 1) and opens the annotator (see Figure 2). Using the previously applied tag, she can easily access and jump between those relevant documents with the document explorer 2). While reading, Alice annotates interesting passages in text, audio, and video documents or regions of interest in image documents on the fly 3). During this process, she constructs a taxonomy by introducing various codes. The code explorer 4) visualizes her team's collaborative hierarchical code tree and allows to rename, delete, and merge codes if required. Codes and annotations from other users and the system can be enabled and disabled 5), which fosters collaboration and discussion. This way, the D-WISE Tool Suite realizes the three coding phases of Grounded Theory (Strauss and Corbin, 1990; Strauss et al., 1996): Open coding by creating new codes on-the-fly and axial and selective coding by providing means to update codes. At any time, Alice can export the automatically and manually created annotations, the created taxonomy, and the raw documents in common formats for further analysis.

Documentation and Reflection While searching and going through documents, Alice learned about new concepts, identified several issues, and developed new ideas and insights on her research topic. The memo feature of the DWTS allows her to attach notes to all objects of interest: documents, annotations, codes, and tags. This reflection process is essential for qualitative discourse analysis to elaborate patterns and phenomena effectively. Finishing her work for today, Alice wants to recap the session. She opens the documentation view, a filterable and searchable overview of her memos. Further, the system automatically logs her interaction



DWTS
Annotator: The islands that want tourists as well as fish
TF
Document Explo...
climate
2)
Visible annotations:
Tim Fischer, SYSTEM USER
5)
CODE EXPLORER
MEMO EXPLORER
6)
ADD CODE
EXPORT
bbc-world-latin-america-64742733-_1287
bbc-uk-northern-ireland-64723591-_1287
bbc-uk-england-stoke-staffordshire-6474
bbc-business-63805028-_127842969_ge
bbc-uk-northern-ireland-64746960-_1287
bbc-world-australia-64578443-_1286225
bbc-64728790-_127816085_iplayer_foote
bbc-world-middle-east-64746658-_12872
bbc-science-environment-57159056-_121
bbc-p00gbjvb-p0cqmk67.jpg
1)
Robot
3)
ELECTRONIC
KEYBOARD
LAPTOP
MOUSE
REMOTE
TV
Robot
ELECTRONIC
KEYBOARD
LAPTOP
MOUSE
REMOTE
TV
Robot
4)
FOOD
OUTDOOR
VEHICLE
NATURE
Island
Mountain
Sea
Clouds
Search
Annotation
Analysis
Logbook

QUANTITY Hundreds of miles from its nearest neighbour, LOC the remote Faroe Islands are surrounded by LOC the Atlantic Ocean. Fishing has always been a way of life, and fish accounts for PERCENT 90% of all exported goods. But coronavirus is hitting efforts to increase tourism.

The drive to the village of GPE Glyvrar is nothing less than dramatic.

The road from the airport winds past mountains and fjords, and passes through tunnels cut into hillsides and burrowed under the sea.

ORG Glyvrar is home to LOC the Faroe Islands' largest firm, ORG Bakkafrost - which farms salmon.

At its state-of-art plant almost 60,000 tonnes of salmon is processed DATE annually.

The operation is highly mechanised, with conveyor belts carrying the fish through a series of machines until the finished product is packaged. Then Robot robotic arms sort boxes ready for shipping.

From hatcheries breeding juvenile salmon to a faci s a foot in almost every tier of the industry. Even the company's branded St characteristic of the LOC Faroe islanders' self-sufficiency.

"We were one of the ORDINAL first operations here tive, PERSON Regin Jacobsen. "We saw that being in the middle of LOC s to get our products to market were quite difficult."

But there was also opportunity, he says. "[We can] alues."

Globally, farming of LOC Atlantic salmon has soar m tonnes in DATE a decade. GPE Norway and GPE Chile are the bigge lmon commands a premium price.

Figure 2: Components of the DWTS annotation interface: 1) The opened document with annotations of the current user and automatic annotations from the SYSTEM; 2) Document explorer with tag selection; 3) Annotation editor popup for code selection; 4) Code explorer with hierarchical codes; 5) Annotation selection; 6) Annotation export button. *The screenshot has been optimized for demonstration purposes.*

with the tool showing exactly when and where she created codes, tags, or annotations. Alice uses the integrated logbook to summarise today's findings. To share her findings with external researchers, she downloads the logbook and sends it via email.

4 System Architecture

The DWTS is a client-server application with a Python backend and a React frontend. A REST API encapsulates core functionalities to enable communication between frontend and backend.

4.1 Backend Architecture

An essential requirement for the DWTS is scalability, i.e., the system's ability to operate with large and growing amounts of data and many simultaneous users. This was considered and implemented from the beginning by utilizing only scalable software frameworks and libraries and deploying and orchestrating the system using modern platform-independent technologies. Further, the backend was designed and implemented following common software patterns and idioms to ensure high-quality software fulfilling essential criteria such as extensibility, availability, scalability, and maintainability. Moreover, the system is required to be open-source, i.e., we require third-party software to be open-source licensed, too.

The backend is divided into several components responsible for different functionalities grouped into data storage and retrieval, data preprocessing, communication, and deployment components.

4.1.1 Data Storage and Retrieval

Arguably the most crucial component of the DWTS is its underlying data model. This data model connects the elements of the business logic, e.g., projects, users, memos, or annotations, with the heterogeneous and multi-modal documents in different representations. Business logic data is modeled as SQL tables using the Python ORM framework SQLAlchemy and stored in a PostgreSQL database.

Text-, image-, audio-, video-, or mixed-modality documents are represented and stored in several ways. The raw files are stored on disk and can be downloaded by users as described in Section 4.1.3. Semantic vector embeddings of documents or segments are stored in FAISS (Johnson et al., 2019) indices. To retrieve the best matching documents for a given query, we apply common information retrieval techniques described in more detail in Section 4.1.2. Textual information is stored in inverted indices using ElasticSearch (Gormley and Tong, 2015)[5] (ES) and is retrieved via ES Query DSL executed utilizing the Python ElasticSearch client

[5] We use v7.16.1 which is open-source licensed

library. A Redis database caches intermediate results from costly operations and stores bug reports and feedback.

4.1.2 Document Preprocessing Pipeline

The document preprocessing pipeline is a central part of the DWTS and responsible for many of its unique selling point features. A schematic overview of the pipeline is shown in Figure 3. In the following, we briefly describe the workflows and essential steps of the pipeline, which is realized as a distributed system using the Celery framework.

Whenever a document is uploaded to the DWTS, a series of actions, referred to as flow, is applied to it depending on its modality. Each flow is executed in an isolated celery worker process that can be run on one or different machines. This design allows for easy scaling of the system if more computing resources are required. Currently, our system supports text, image, audio, and video documents resulting in four different flows, described in more detail in the following.

Text Document Flow Text documents such as HTML, PDF, Word, or TXT files are processed in the text flow of the pipeline. After the document is stored on disk and registered in the database, the textual content gets extracted. For PDF and Word documents, we use Apache Tika. For HTML files, we use Readability.JS, to retain only a website's content. Image, video, or audio files in an HTML page are extracted beforehand and run separately through the respective flows. Next, metadata like the language of the text is detected using langdetect to load the language-specific pretrained language model (PLM). Currently, we support English, German, and Italian using the respective transformer models (Vaswani et al., 2017; Devlin et al., 2019) within the spaCy framework (Honnibal et al., 2020), which we also use in the subsequent steps to do tokenization, sentence segmentation, and Named Entity Recognition. Sentence embeddings are computed using a pretrained multilingual CLIP (Radford et al., 2021; Reimers and Gurevych, 2020) model from the SentenceTransformers framework (Reimers and Gurevych, 2019). CLIP is a state-of-the-art visiolinguistic model with strong zero-shot performance in many downstream tasks, including text-to-image and image-to-text retrieval. The embeddings are stored in FAISS (Johnson et al., 2019) indices. Finally, the textual content is stored in an ElasticSearch index.

Image Document Flow First, the image is stored on disk and in the database with its extracted metadata like the image dimensions. Then, we detect objects in the image using a pretrained DETR (Zhu et al., 2021) model, an efficient and effective object detection model available within the huggingface transformers framework (Wolf et al., 2020). Next, a global semantic image embedding is computed using the same CLIP model as in the text flow and stored in a FAISS index. This enables multi-modal similarity search of images and texts. Finally an image caption is generated utilizing a pretrained BLIP-2 (Li et al., 2023) model. This caption is passed through the text flow to enable lexical and semantic search.

Audio Document Flow In this flow, audio documents in standard formats, such as MP3 or WAV, are processed. The file is stored on disk and in the database along with its extracted metadata like the length of the recording. Then the audio file gets chunked in about 5s segments, which result in about 16.5 words with an average words-per-minute rate of 198 (Wang, 2021), which is in the range of the average English sentence lengths (Moore, 2011). These chunks are then forwarded through a Whisper (Radford et al., 2022) model to compute semantic embeddings stored in a FAISS index to enable audio-to-audio similarity search.[6] In the final step, we generate a textual transcription of the audio using a Whisper model. We treat the audio transcription as a text document and run it through the text flow to support lexical and semantic textual search.

Video Document Flow Documents in standard video formats, such as MP4 or MOV, are processed in the video flow. Again, first, the file is stored on disk and in the database along with its extracted metadata, like the duration or dimensions. Afterwards, following the motivation for audio files, the video gets chunked into about 5s clips. These chunks are then forwarded through a VideoMAE (Tong et al., 2022) or X-CLIP (Ni et al., 2022) model to compute rich semantic embeddings, which are stored in a FAISS index to enable video-to-video similarity search.[7]

Finally, we extract the audio stream from the video and process it in the audio flow.

[6] The similarity search for raw audio and video documents or chunks is work-in-progress and has yet to be released in our system demonstration DWTS instance.

[7] This is still undergoing research and is not yet available in the DWTS demonstration instance.

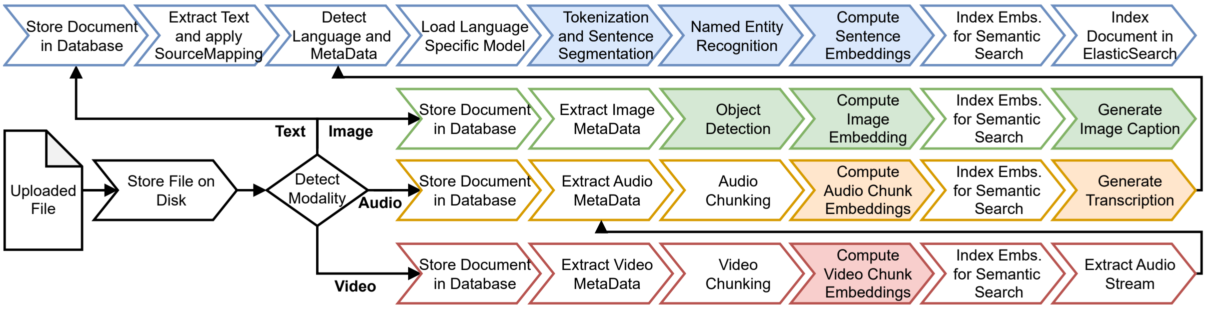

Store Document in Database
Extract Text and apply SourceMapping
Detect Language and MetaData
Load Language Specific Model
Tokenization and Sentence Segmentation
Named Entity Recognition
Compute Sentence Embeddings
Index Embs. for Semantic Search
Index Document in ElasticSearch
Uploaded File
Store File on Disk
Detect Modality
Text
Image
Audio
Video
Store Document in Database
Extract Image MetaData
Object Detection
Compute Image Embedding
Index Embs. for Semantic Search
Generate Image Caption
Store Document in Database
Extract Audio MetaData
Audio Chunking
Compute Audio Chunk Embeddings
Index Embs. for Semantic Search
Generate Transcription
Store Document in Database
Extract Video MetaData
Video Chunking
Compute Video Chunk Embeddings
Index Embs. for Semantic Search
Extract Audio Stream

Figure 3: Illustration of the multi-modal preprocessing pipeline in the DWTS. Each flow for a specific modality is executed in a separate Celery worker process. Steps involving deep learning models are highlighted.

4.1.3 Client-Server Communication

A REST API implemented using FastAPI encapsulates all functionality of the DWTS. Using FastAPI, an OpenAPIv3 schema and a SwaggerUI are automatically generated, drastically easing the development of UIs or other external applications consuming DWTS functionality. To simplify communication between clients and the server, the parameters and return types of all the API endpoints are defined as data-transfer-objects (DTO) using Pydantic models. These DTOs are also widely used within the backend to decouple entities from database sessions, for communication between different system components, and to transmit data between the Celery workers of the preprocessing pipeline.

Further, we employ lighttpd as a file server for raw text, image, audio, and video files.

4.2 Frontend Architecture

The frontend is an interactive web application build with React and TypeScript. It communicates with the backend by consuming its RESTful API to realize most features. Since the backend exposes its functionality with an API defined by an OpenAPIv3 schema, we use a code generator to automatically generate a client that fully supports all functionality, including TypeScript interfaces or classes for all parameter and return types. Data fetching, caching, and synchronization are handled with React-Query making the app faster and more responsive while saving bandwidth and increasing performance. The client state is managed by Redux, which enables powerful functionalities like undo/redo and state persistence. The interface and interaction follow the recognized Material Design System, which utilizes material-like surfaces for the components. We use ready-made components from the MUI library, as well as custom-tailored components to implement the user interface.

4.3 System Deployment

The DWTS is deployed using modern containerization technology. Every system component, i.e., the databases and indices, API, Celery workers, and the frontend, is deployed in a separate Docker container. All containers are orchestrated in a Docker Compose file that makes up the DWTS system. This approach has several advantages: First, the software is platform-independent and can be conveniently deployed on any modern system or infrastructure with minimal adaptation and configuration efforts. Thus, the system can be deployed on local servers for projects with confidential or privacy constrained data. Second, the components can be deployed on different machines, e.g., computationally intensive components like the Celery workers can run on a GPU server, while memory-intensive components like indices and databases can run on a storage server. Third, the system can be easily scaled using Docker Swarm or similar technology.

5 Conclusion

This paper presented the D-WISE Tool Suite (DWTS), a web-based open-source application to support and enhance digital discourse analysis for Digital Humanities by being able to operate on large amounts of heterogeneous and multi-modal data. We discussed the motivation and need for our system by pointing out the limitations of existing DH tools for extensive multi-modal document collections. Further, we demonstrated central functionalities of the DWTS by describing a typical workflow illustrated by screenshots, and provide technical details about the system architecture in a separate section. Currently work-in-process but released in future work are video and audio annotations and 4-way multi-modal similarity search between text, image, audio, and video documents.

Limitations & Ethics Statement

The DWTS makes state-of-the-art machine learning (ML) models accessible to researchers that could previously not benefit from these advances. Our tool targets Digital Humanities researchers and is intended to assist with qualitative discourse analysis. As with most digital tools, though, it could be misused for other work. We strongly believe that including and enabling more researchers to benefit from modern ML technology outweighs the potential for misuse.

When using ML models, it is important to understand their limitations and critically reflect on their predictions. ML models often include certain biases that can manifest in various types and forms and are not without error. We try to mitigate this by visualizing confidence scores where applicable and additionally provide traditional methods as ML-free alternatives. In particular, we offer a multi-modal semantic similarity search and highlight the confidences, but also provide a standard lexical search to cross-check the results. Still, errors in preprocessing steps, like removing relevant content during data cleanup or the entity recognizer model missing a certain entity of interest, may lead to biased search and analysis results.

Naturally, we also introduce a bias with our system design and envisioned workflow. While we tried our best to model the process of digital discourse analysis as closely as possible, we might still restrict a user in their workflow by our design decisions.

Regarding privacy and security, we identified and mitigated two limitations. The DWTS requires users to upload their potentially sensitive data to the system. To alleviate this, we ensured the tool is easily deployable and can be self-hosted even by non-experts allowing users to stay in full control of their data. In particular, it is even possible to install DWTS locally on private devices. Further, the DWTS includes a feature that automatically logs user actions and populates a logbook in order to improve the documentation and reflection of research processes. By making this an opt-in feature, we guarantee that users are in control of their usage data.

We are aware of the limitations of our system and the technology therein and are committed to actively participating in discussions with domain experts regarding ML, privacy, and bias to identify and iron out further constraints.

Acknowledgement

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

References

Giuseppe Abrami, Manuel Stoeckel, and Alexander Mehler. 2020. TextAnnotator: A UIMA Based Tool for the Simultaneous and Collaborative Annotation of Texts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 891–900, Marseille, France. European Language Resources Association.

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*, pages 4171–4186, Minneapolis, MN, USA.

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan.

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.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide*, 1st edition. O'Reilly Media, Inc.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. https://spacy.io/.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.

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

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pretraining with Frozen Image Encoders and Large Language Models. *ArXiv*, abs/2301.12597.

Andrew Moore. 2011. The long sentence: a disservice to science in the internet age. *BioEssays*, 33(12):193–193.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. 2022. Expanding Language-Image Pretrained Models for General Video Recognition. In *Proceedings of the 17th European Conference on Computer Vision (ECCV)*, pages 1–18, Tel Aviv, Israel.

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.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *ArXiv*, abs/2212.04356.

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.

Steffen Remus, Hanna Hedeland, Anne Ferger, Kristin Bührig, and Chris Biemann. 2019. WebAnno-MM: EXMARaLDA meets WebAnno. In *In Selected papers from the CLARIN Annual Conference, 2018*, Pisa, Italy.

Florian Schneider, Seid Muhie Yiman, Fynn Petersen-Frey, Gerret von Nordheim, Katharina Kleinen-von Königslöw, and Chris Biemann. 2023. CodeAnno: Extending WebAnno with Hierarchical Document Level Annotation and Automation. In *The 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023), System Demonstrations*, Dubrovnik, Croatia.

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*.

Rainer Simon, Elton Barker, Leif Isaksen, and Pau de Soto Cañamares. 2017. Linked data annotation without the pointy brackets: Introducing Recogito 2. *Journal of Map & Geography Libraries*, 13(1):111–132.

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.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In *Advances in Neural Information Processing Systems (NIPS)*, New Orleans, LA, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, pages 5998–6008, Long Beach, CA, USA.

Li Wang. 2021. British English-Speaking Speed 2020. *Academic Journal of Humanities & Social Sciences*, 4(5):93–100.

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 (EMNLP)*, pages 38–45, Online.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations (ICLR)*, Vienna, Austria (Virtual Event).