A Visualization of Word and Document Embeddings

Joseph Chataignon[1], and Tobias Hodel[1]

[1] Digital Humanities Department, University of Bern, Bern, Switzerland

Abstract

This paper introduces an open-source visualization tool designed to enhance the comprehension of word and document embeddings. Word embeddings, which translate words into high-dimensional numerical vectors, and document embeddings, which encapsulate the meaning of entire documents, are fundamental to the recent advancements in Natural Language Processing (NLP), particularly with the rise of large language models (LLMs). Following the broader movement to understand new NLP models, our tool is tailored for individuals in the Humanities and requires no prior technical knowledge, offering an interactive and user-friendly interface to explore complex relationships between words and documents in a high-dimensional space.

Implemented using a web interface, the tool supports multiple datasets for document embeddings and utilizes pre-trained models like GloVe for word embeddings and Sentence-Transformers for document embeddings. User feedback indicates that the tool is effective in improving the understanding of embeddings. Future work includes enhancing the interface, incorporating more embedding models, and translating the interface into additional languages. This tool represents a step forward in making advanced NLP concepts accessible to a wider audience.

Keywords: Visualization, word embeddings, document embeddings

1 Introduction

Word embeddings have emerged as a crucial component of text processing applications, playing a pivotal role in enabling machines to interpret the semantics of natural language. By translating words into high-dimensional numerical vectors, these embeddings encapsulate the meaning and the relationships of words, thereby forming the bedrock of the most recent advances in natural language processing (NLP). The advent of large language models (LLMs), in particular, has underscored the importance of word embeddings, as these models rely heavily on the nuanced representations of words to generate coherent and contextually relevant text. The rapid advancement and widespread adoption of LLMs in recent years mean that many areas, such as machine translation, sentiment analysis, and text generation, are dependent on the effectiveness of word embeddings.

A similar technique applied to documents produces document embeddings. Document embeddings are vectors of a high-dimensional space that encapsulate the meaning and relationships of whole documents. They too have proven useful in a number of tasks such as document retrieval and document classification.

This paper presents a visualization tool that aims to improve the comprehension of both word embeddings and document embeddings. It contributes to the broader movement toward explainable AI and interpretable NLP models, an endeavor that becomes increasingly important as these techniques become more widely used. The visualization tool is designed for individuals in the Humanities field and requires no prior computer or technical knowledge to operate. For users

Joseph Chataignon, and Tobias Hodel. “A Visualization of Word and Document Embeddings.” In: *Computational Humanities Research 2025*, ed. by Taylor Arnold, Margherita Fantoli, and Ruben Ros. Vol. 3. Anthology of Computers and the Humanities. 2025, 1204–1211. https://doi.org/10.63744/dSuAPBjLPURS.

with more technical expertise, the tool can be easily adapted into an analytical instrument for new datasets and models.

1.1 Related work

While word and document embeddings have taken an important place in modern NLP, there remains a gap in accessible visualization tools that bridge the technical complexity of these representations with the needs of non-technical users. This section reviews existing approaches to embedding visualization.

Numerous visualization approaches for word embeddings have been developed with varying degrees of user accessibility, though relatively few are deployed as publicly accessible web interfaces. Existing online word embedding visualizers, such as W2V Explorer [2] and TensorFlow's Word2Vec Projector [12], provide intuitive interfaces but have limitations in their capacity to reconfigure the projection space based on user-selected subsets of data.

The landscape of document embedding visualization tools presents similar accessibility challenges. While several visualizations of document embeddings have been made available online, these implementations are predominantly distributed within computational notebooks [9] [4] and code repositories, creating accessibility barriers for humanities scholars without programming expertise. RAGxplorer [1] offers a simple interface but strong limitations on the type of data that can be visualized. Critically, similar to word embedding visualization tools, none of these existing solutions natively allow users to reset the dimensionality reduction projection based on a selection of items.

1.2 Objective

The primary objective of the visualization tool is to improve the understanding of word embeddings and document embeddings. It should be designed to offer an interactive and user-friendly interface that allows researchers and practitioners to explore the complex relationships between words in a high-dimensional space. Specifically, the tool should let users explore the positions of arbitrary words in 2-dimensional projections of the vector space, and let users change the 2-dimensional projection itself to highlight the relationships of any given set of words.

The visualization tool is available online. [Link will be published after peer review]

Furthermore, the visualization tool is intended to serve as a foundational platform upon which more advanced analytical tools can be developed, for users with greater technical expertise. This includes functionalities for comparing document embedding models within a Retrieval-Augmented Generation (RAG) framework, as well as analyzing user-provided datasets.

2 Methods

2.1 Requirements

The visualization should enable users to select words and display them within a two-dimensional projection of the vector space. By default, this projection should be determined using Principal Component Analysis (PCA) applied to all vectors in the dataset. Additionally, users should have the ability to reset the projection by performing a PCA specifically on the vectors of the selected words or documents, allowing for a more customized and focused analysis.

The two-dimensional projection space should be fully explorable, with functionalities for panning and zooming. This interactivity will let users navigate the projection space intuitively.

In addition, the application is:

- **open-source.** The tool is released under an open-source license to ensure widespread access. It can be used, modified, and distributed freely by anyone, thus promoting collaborative improvement and innovation.
- **multi-user.** Deployed online, the visualization tool is able to support multiple users concurrently.
- **multi-lingual.** The interface is currently being translated and will be made available in multiple languages to enhance accessibility for a global audience.

2.2 Tools

Models. For word embeddings, we chose to use a pre-trained GloVe [10] model. It was trained on text from the English Wikipedia and the English Gigaword 5 dataset of newswire text data. It has a vocabulary of 400,000 words, and the word vectors have 300 dimensions. For document embeddings we used the Sentence-Transformers fine-tuned model all-MiniLM-L6-v2 available at Hugging Face [13]. This model maps documents to a 384-dimensional space. It was pre-trained with contrastive loss on a total of 1.17 billion sentence pairs from various datasets.

Datasets. For document embeddings, 5 datasets are made available to the user to choose from:

- Reuters-21578 [8], a dataset of short news articles.
- Dbpedia-14 [7] [15], a collection of encyclopedic articles.
- eli5 [3], a dataset of pairs of questions-answers.
- gooaq [6], another dataset of pairs of questions-answers.
- AGnews [5] [14], a second dataset of news articles.

These datasets were chosen because they contain documents of a few sentences, short enough to be understood quickly in the interface.

Software. The visualization uses a web interface and includes the Javascript library Plotly [11] to display vectors. The back-end is a program written in Python with Flask. The Python libraries Numpy, Gensim, Scikit-learn, Sentence-transformers and Huggingface-datasets were used.

3 Implementation

The visualization application can be deployed either in word embedding mode or in document embedding mode. The heaviest processing is done server-side, while the client-side only handles the front-end.

3.1 Architecture

The detail of the architecture is shown in Figure 1. In the user's browser, the interface is made of an HTML template that includes a Plotly canvas. The interactive elements are managed with Javascript code, including the Plotly.js module to display the embeddings.

On the server side, a Flask application handles multiple endpoints to render the main page, load the embeddings, search words or documents, reset the projection, and fetch the interface translations. An EmbeddingVisualizer module handles the data processing. A session management module is used to separate the various settings of concurrent users.

Large resources such as datasets and embedding models are kept on the server for fast access.

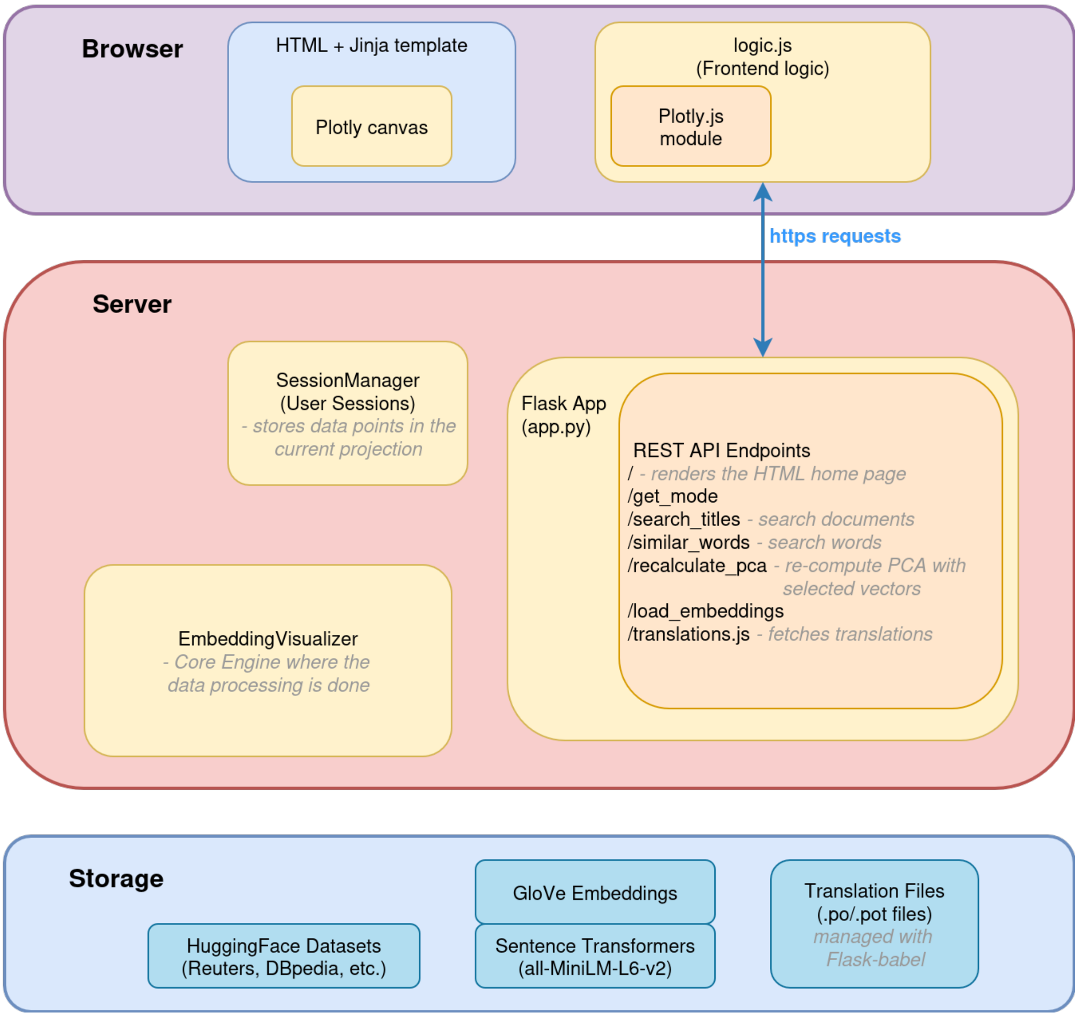


Figure 1: Architecture of the application.

3.2 Interface

The interface of our visualization tool is designed to provide a user-friendly experience, ensuring that users can effortlessly navigate and utilize its features. We have included screenshots that illustrate the interface for both word embedding (Figure 2) and document embedding (Figure 3) visualizations.

In both modes, users can select their preferred interface language using a language selection drop-down menu, as shown in Figure 2.1 and Figure 3.1 .

In document embedding mode, the first component of the interface is the drop-down menu to select a dataset, shown in Figure 3.2 . This feature allows users to choose one of the five datasets mentioned in section 2.2. The button labelled ”Load Embeddings” (Figure 3.3), when clicked, computes and stores the embeddings of all the documents in the selected dataset. These elements are not present in word embedding mode, because only the GloVe model and its vocabulary are available. It is therefore loaded by default when the application starts.

To search and select specific words or documents to visualize, a search bar is available (Figure 2.2 and Figure 3.4). By simply typing in the search bar, users can filter through the vast array of options and find the exact item, word, or document, that they wish to visualize. Once selected, an item remains displayed in the search bar for an intuitive view of the current selection.

The plot area (Figure 2.3 and Figure 3.5) follows the search and selection, and forms the heart

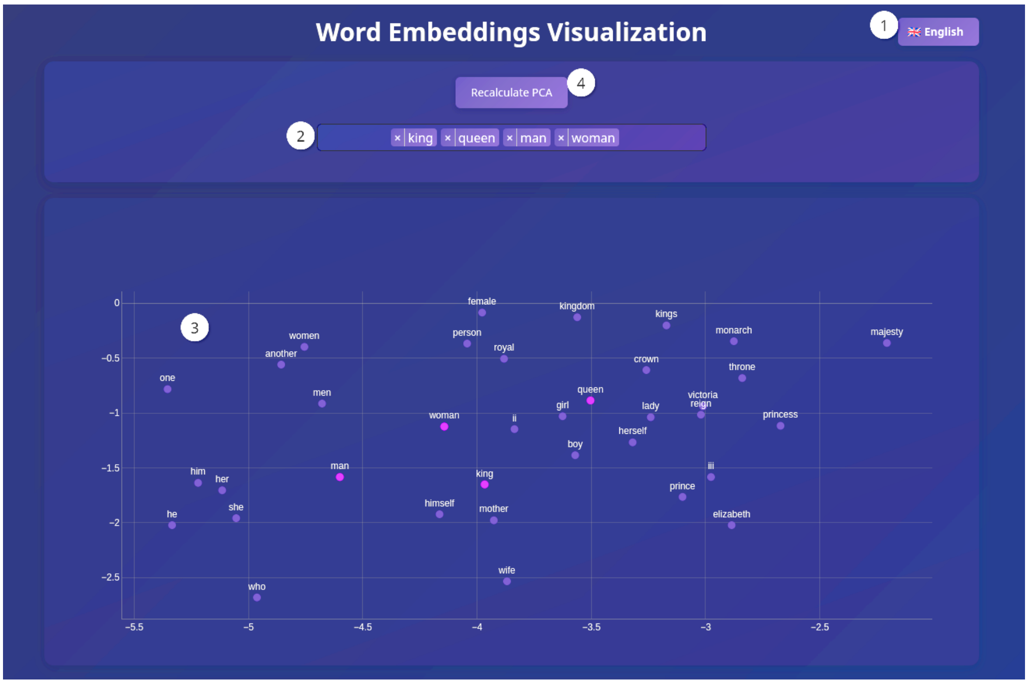


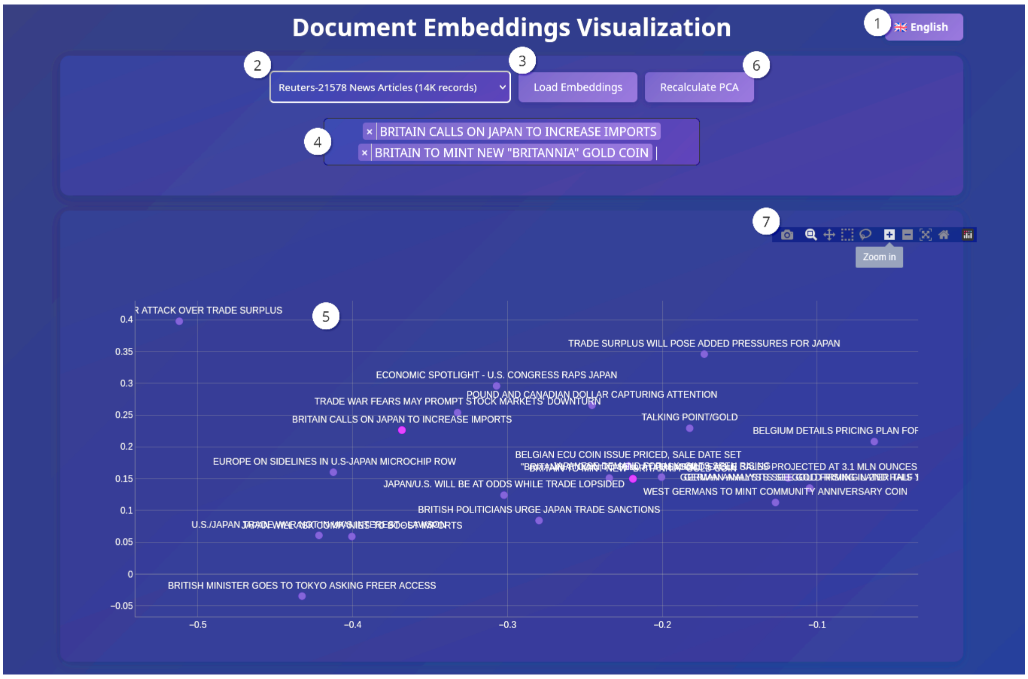
Figure 2: Screenshot of the user interface in word embedding mode. The following elements are labelled with numbers: (1) language selection (2) word search bar (3) plot area (4) button to re-set the projection

of the visualization. Selected items are displayed as bright dots, and the 10 nearest neighbors of each selected item are displayed in a dimmer color. In word embedding mode, the words are displayed next to their corresponding dot. In document embedding mode, the titles of documents are displayed instead.

The plot area has control buttons (Figure 3.7) that appear when hovering the mouse over it, to move through the embedding space or download an image of the plot.

The last notable feature of the interface is the button to reset the projection based on the selected items (Figure 2.4 and Figure 3.6). As explained in section 2.2, the embedding models we use produce embedding vectors of length 300 for words, and 384 for documents. The projection from the resulting 300-dimensional space or 384-dimensional space into a 2-dimensional space suitable for plotting is determined using Principal Component Analysis (PCA). PCA is a dimensionality reduction technique that preserves the most significant variations in the data. At the start, by default, we run the PCA on the whole dataset. But users can use this button to re-run the PCA only on the selected items instead of the whole dataset. When the button is clicked, the PCA is computed again on the selected words only, producing a new projection from high dimensions to 2 dimensions, which is then applied to the whole dataset. In practice, this allows users to highlight the variations that exist between selected items. The new projection sets the selected words apart from each other, and lets users see how other words compare to them.

In addition, the application incorporates a dedicated section at the end that provides explanations about embeddings in general and the specific features of the visualization.



Document Embeddings Visualization

Reuters-21578 News Articles (14K records)

× BRITAIN CALLS ON JAPAN TO INCREASE IMPORTS
× BRITAIN TO MINT NEW "BRITANNIA" GOLD COIN

1
🇬🇧 English
2
3
Load Embeddings
6
Recalculate PCA
4
7
Zoom in
5
R ATTACK OVER TRADE SURPLUS
TRADE SURPLUS WILL POSE ADDED PRESSURES FOR JAPAN
ECONOMIC SPOTLIGHT - U.S. CONGRESS RAPS JAPAN
POUND AND CANADIAN DOLLAR CAPTURING ATTENTION
TRADE WAR FEARS MAY PROMPT STOCK MARKETS' DOWNTURN
TALKING POINT/GOLD
BRITAIN CALLS ON JAPAN TO INCREASE IMPORTS
BELGIUM DETAILS PRICING PLAN FOF
BELGIAN ECU COIN ISSUE PRICED, SALE DATE SET
EUROPE ON SIDELINES IN U.S-JAPAN MICROCHIP ROW
PROJECTED AT 3.1 MLN OUNCES
JAPAN/U.S. WILL BE AT ODDS WHILE TRADE LOPSIDED
WEST GERMANS TO MINT COMMUNITY ANNIVERSARY COIN
BRITISH POLITICIANS URGE JAPAN TRADE SANCTIONS
BRITISH MINISTER GOES TO TOKYO ASKING FREER ACCESS
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
−0.05
−0.5
−0.4
−0.3
−0.2
−0.1

Figure 3: Screenshot of the user interface in document embedding mode. The following elements are labelled with numbers: (1) language selection (2) dataset selection (3) button to load the dataset's embeddings (4) document search bar (5) plot area (6) button to re-set the projection (7) plot area controls

4 Results and discussion

4.1 Deployment

The word embeddings visualization tool has been successfully deployed online at https://redacted-for-anonymity, providing users with an accessible and interactive way to explore word embeddings. This deployment relies on a Linux server environment, coupled with the Apache Server and the mod_wsgi module, which facilitates the integration of Python-based web applications with Apache. This configuration ensures efficient performance for multiple users.

While the word embeddings component of the tool is fully operational, the deployment of document embeddings is pending due to the necessity for a more powerful server infrastructure. Document embeddings, which require substantial computational resources for processing and visualization, will be made available online as soon as the resources are available.

4.2 User feedback

Feedback from users was gathered through both in-person interactions and an online feedback form to ensure a comprehensive understanding of the tool's usability. The visualization was generally appreciated for its user-friendly interface and intuitive design. Almost all users, especially those unfamiliar with the topic, reported that the visualization improved their understanding of embeddings, thereby fulfilling the primary objective of the project.

However, users found the feature designed to reset the projection based on selected items challenging to understand. Despite the inclusion of an explanations section within the interface, this feature remained confusing for many.

Overall, user feedback indicated that while the visualization tool was effective in enhancing the understanding of embeddings and was praised for its user-friendly design, there is a need for improvements in explaining and simplifying the most advanced features.

4.3 Discussion and future work

The visualization tool has shown success in improving the comprehension of word and document embeddings, as substantiated by the positive feedback received from users. Its ability to elucidate these complex concepts through interactive and intuitive visualizations has been well-received, indicating that it effectively meets its primary objective of improving understanding in the field of natural language processing. For instance, the tool has revealed interesting linguistic insights, such as the distinct clustering of French words separate from a main English cluster in GloVe embeddings, and the inclusion of words that exist in both languages firmly inside the English cluster, demonstrating its potential for deeper linguistic analysis.

Looking ahead, several areas have been identified for future work to enhance the tool's functionality and accessibility. Firstly, clarifying the interface to make features more comprehensible, particularly the feature to reset the projection, is essential. This could involve providing more detailed and clearer explanations or tooltips to guide users. Secondly, incorporating a wider range of embedding models, especially more modern word embedding models, will ensure that the tool remains up-to-date with the latest advancements in the field. In addition, we consider facilitating the use of user-uploaded datasets for the document embeddings visualization, which would let users analyze the embedding structure of their own data. Lastly, we will work on translating the interface into additional languages to enhance the tool's accessibility, as it is currently available only in English and French. By addressing these areas, the visualization tool can continue to evolve and serve as a valuable resource for researchers and practitioners in the field of NLP.

References

[1] Chua, Gabriel. "RAGxplorer: Visualizing Document Chunks in the Embedding Space". Online demonstration. Source code available at: https://github.com/gabrielchua/RAGxplorer. 2024. URL: `https://ragxplorer.streamlit.app/`.

[2] Cortext team. "W2V embedding Visualizer". URL: `https://documents.cortext.net/lib/W2Vexplorer/index.html`.

[3] Fan, Angela, Jernite, Yacine, Weston, Jason, and Bordes, Antoine. "ELI5: Long Form Question Answering". In: *Proceedings of ACL*. The dataset was accessed through the sentence-transformers dataset collection on Huggingface. 2019. URL: `https://arxiv.org/abs/1907.09190`.

[4] Grootendorst, Maarten. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022). visualization available in `https://maartengr.github.io/BERTopic/getting_started/visualization/visualization.html`. URL: `https://arxiv.org/abs/2203.05794`.

[5] Gulli, Antonio. "AG's corpus of news articles". 2005. URL: `https://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`.

[6] Khashabi, Daniel, Ng, Amos, Khot, Tushar, Sabharwal, Ashish, Hajishirzi, Hannaneh, and Callison-Burch, Chris. "GooAQ: Open Question Answering with Diverse Answer Types". In: *arXiv preprint* (2021). The dataset was accessed through the sentence-transformers dataset collection on Huggingface.

[7] Lehmann, Jens et al. "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia - 2014 Release". 2014. URL: `https://downloads.dbpedia.org/wiki-archive/data-set-2014.html`.

[8] Lewis, David D. “Reuters-21578 Text Categorization Collection”. The dataset was The dataset was accessed through the sentence-transformers dataset collection on Huggingface. UCI Machine Learning Repository, 1987. DOI: 10.24432/C52G6M.

[9] Marimo team. “Visualizing text embeddings using MotherDuck and marimo”. 2024. URL: https://huggingface.co/spaces/marimo-team/motherduck-embeddings-visualizer.

[10] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher. “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Association for Computational Linguistics, 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162/.

[11] Plotly Technologies Inc. “Collaborative data science”. 2015. URL: https://plot.ly.

[12] Tensorflow. “TensorBoard Embedding Projector”. URL: https://projector.tensorflow.org/.

[13] Wang, Wei, Liu, Li, and Cho, Kyunghyun. “MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers”. In: *arXiv preprint arXiv:2002.10957* (2020). URL: https://arxiv.org/abs/2002.10957.

[14] Zhang, Xiang, Zhao, Junbo, and LeCun, Yann. “AG News Topic Classification Dataset”. AG News dataset constructed for topic classification. 2015. URL: https://huggingface.co/datasets/fancyzhx/ag_news.

[15] Zhang, Xiang, Zhao, Junbo, and LeCun, Yann. “DBpedia 14”. DBpedia-based dataset constructed for ontology classification. 2015. URL: https://huggingface.co/datasets/fancyzhx/dbpedia_14.