LLM Assisted Analysis of Text-Embedding Visualizations

Allen Detmer*
detmeran@mail.uc.edu

Raj Bhatnagar†
bhatnark@ucmail.uc.edu

Jillian Aurisano‡
aurisajm@ucmail.uc.edu

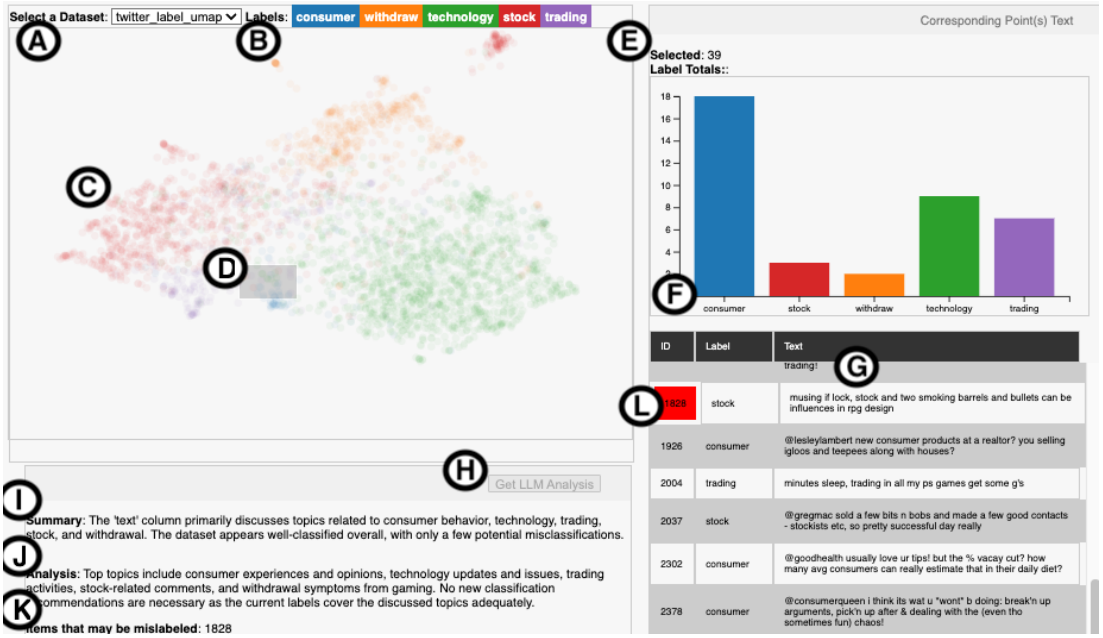

Figure 1: Text-Embedding Selection Sidekick (TESS) (A) Dataset selection (B) Text labels in the dataset (C) Two-dimensional text embedding points (D) Brush selection of points (E) Number of selected points (F) Bar chart displaying the count of each label in the brushed selection (G) Table showing identifier, label and text of brushed selection, with sortable columns (H) Button for prompting the LLM driven system to analyze the underlying text and labels of brushed selection data (I) Summary generated by the LLM for the selected data points (J) Analysis generated by the LLM for the selected data points (K) Items that the LLM identifies as possibly misclassified (L) Corresponding items in the table that the LLM identified as possibly misclassified

Abstract

Dimensionality reduction is a widely adopted tool in Natural Language Processing (NLP). Techniques such as Uniform Manifold Approximation and Projection (UMAP) transform high-dimensional embeddings of text data into a lower-dimensional space for visualization. Two-dimensional plots of these embeddings aid in developing insights into model performance. To make sense of these plots, users need to inspect the underlying text represented by the points which can be time-consuming and cognitively intensive. To address this challenge, we developed a novel approach for summarizing and analyzing data behind user selections in text embedding plots. Our interactive approach involves allowing the user to make selections on the text embedding and then utilizing a large-language model (LLM) for: getting a quick overview of the selection, identifying instances of miss-classification, understanding text data within a mixed-class selection, and suggesting additional labels that better fit the underlying text. We implemented our approach in a prototype application, Text-Embedding Selection Sidekick (TESS), and present our initial results.

*e-mail: detmeran@mail.uc.edu
†e-mail: bhatnark@ucmail.uc.edu
‡e-mail: aurisajm@ucmail.uc.edu

Index Terms: LLM, Text-Embedding, Visualization.

1 Introduction

Natural Language Processing (NLP) is used in everyday life from smart devices, social media, and numerous other use cases across various domains. It has become standard in NLP to reduce high-dimensional text data to low-dimensional embedding spaces. These embedding spaces can be reduced to two-dimensions for visualization with attraction/repulsion dimensionality reduction (ARDR) methods, such as Uniform Manifold Approximation and Projection (UMAP) [5]. Text-embedding visualizations can be used for comparing class labels and sentence similarity. Exploring embedded text data assists in gaining insights, viewing semantic relationships, providing an overview of the structure and separation[4].

However, it can be difficult to understand text-embeddings and use them to explain or diagnose NLP models. Interaction techniques such as details-on-demand or brushing-and-linking can be used to allow users to inspect specific points or clusters. But synthesizing and analyzing selected texts can be time-consuming and cognitively intensive. Text summarization methods like word clouds or tag clouds rely on frequencies and do not show word relationships.

To address this challenge, we developed a novel approach for summarizing and analyzing data behind user selections in text-embedding plots. Our approach leverages large-language models (LLMs) (e.g., GPT [6]) which have demonstrated remarkable possibilities for summarizing and analyzing textual data [9]. We integrated our approach in a prototype tool, Text-Embedding Selec-

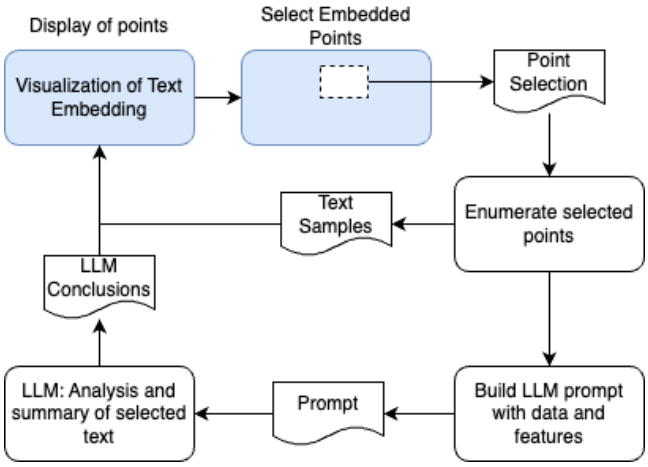

Figure 2: An illustration of TESS process flow. Brush interactions select points that are then enumerated and used to build the prompt to the LLM. The resulting LLM response is displayed in the visualization as summary, analysis, and possible misclassifications.

tion Sidekick (TESS). Our goal is to reduce the cognitive burden in understanding text embedding plots and using them for model evaluation and refinement.

In this abstract, we present a brief overview of the design and implementation of TESS, and describe our plans for future work.

2 Application

2.1 Overview, Goals, Tasks

We have developed a set of preliminary tasks that TESS supports to reduce cognitively burdensome and time-consuming manual tasks of reading and analyzing the text behind the points in a plot of text-embeddings. These include: (1) Reviewing the underlying text data and assessing general model performance, (2) Explaining small regions of embedded space having overlapping labels, (3) Identifying and explaining mislabeled or misclassified text, and (4) Identifying classifications that would better partition the data. These tasks are drawn from the first author's several years of experience working in NLP in academic and industrial settings.

2.2 Implementation

Data and Pre-processing: We utilized two multi-class labeled datasets for testing. These datasets were selected because they show different distributions of text similarity and accuracy in labeling, allowing us to explore the use of TESS for model debugging and refinement. The first dataset is emotion-labeled data [2]. For demonstration we selected the first 3000 entries. The second dataset is from the paper 'Twitter sentiment classification using distant supervision' [3]. We filtered this dataset to include texts potentially related to financial content, by selecting tweets with words "stock", "trading", "withdraw", "consumer", and "technology". We obtained text-embeddings using Sentence Bert [7].

System overview: Fig. 2 shows the interactive process flow for TESS. TESS uses D3 [1] for visualization. User selections are enumerated and inserted with markdown into the LLM prompt. The LLM prompt includes instructions to provide a summary, analysis, and possible misclassified labels, and the output is then displayed in the TESS application interface. TESS is currently designed for datasets consisting in 3,000-5,000 short text instances (e.g., Tweets), which covers many real-world usage scenarios.

Prompt construction: Prompt engineering is critical for sending effective instructions and data to the LLM. We used an iterative approach of prompt testing and refinement [8] a zero-shot prompt.

2.3 Visualization and Interface

Fig. 1 shows the interface. The main view is the text-embedding, which allows the user to quickly visualize the text classifications and distances (Fig. 1 C). The interface includes a dataset selection drop-down menu at the top and a legend displaying class labels, at the top (Fig. 1 A). All the points from the dataset are plotted using categorical color scheme for class labels (Fig. 1 B). When the user identifies a region of interest, they may use a brush interaction to select those points (Fig. 1. D) The count of selected points is displayed (Fig. 1 E) along with a bar chart showing the number of points for each label (Fig. 1 F). A table displays the text and class label for each selected text (Fig. 1 G). User can click the button "Generate LLM Summary" (Fig. 1 H) to trigger the LLM-driven system to generate a summary and analysis of the underlying text data. The corresponding results are showed in three categories: Summary (Fig. 1 I), Analysis (Fig. 1 J), and "Items that may be Misclassified" (Fig. 1 K). The potentially misclassified items are highlighted in red in the table, so the user can assess the LLM's response (Fig. 1 L). The LLM generated response times range from 2 to 4 seconds with self testing. A graphical spinner is displayed as an indicator that the generation is in progress.

3 Conclusion and Future Work

TESS is a prototype tool for exploring interactive visualizations with LLM assisted explanations and analysis. Initial internal tests with TESS suggest such LLM assisted tools could significantly reduce manual and time-consuming text analysis tasks. Future work will involve conducting user studies and evaluations for exploring system effectiveness and future tool enhancements. We also intend to address how our approach might scale to larger datasets and longer text inputs. In internal testing we noted a few instances where the LLM did not consistently follow prompt instructions. We addressed this issue through prompt refinement and verification functions in our system. We will research other techniques eg. one-shot and chain-of-thought (CoT) for improved quality of the LLM response. We will expand techniques for verification of LLM-generated responses in future work.

References

[1] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011. 2

[2] N. Elgiriyewithana. Emotions, 2024. doi: 10.34740/KAGGLE/DSV/7563141 2

[3] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009. 2

[4] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci. Visual exploration of semantic relationships in neural word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):553–562, 2018. doi: 10.1109/TVCG.2017.2745141 1

[5] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. 1

[6] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1

[7] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. 2

[8] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023. 2

[9] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024. 1