

Visual Comparison of Text Sequences Generated by Large Language Models

Rita Sevastjanova , Simon Vogelbacher, Andreas Spitz , Daniel Keim , Mennatallah El-Assady 

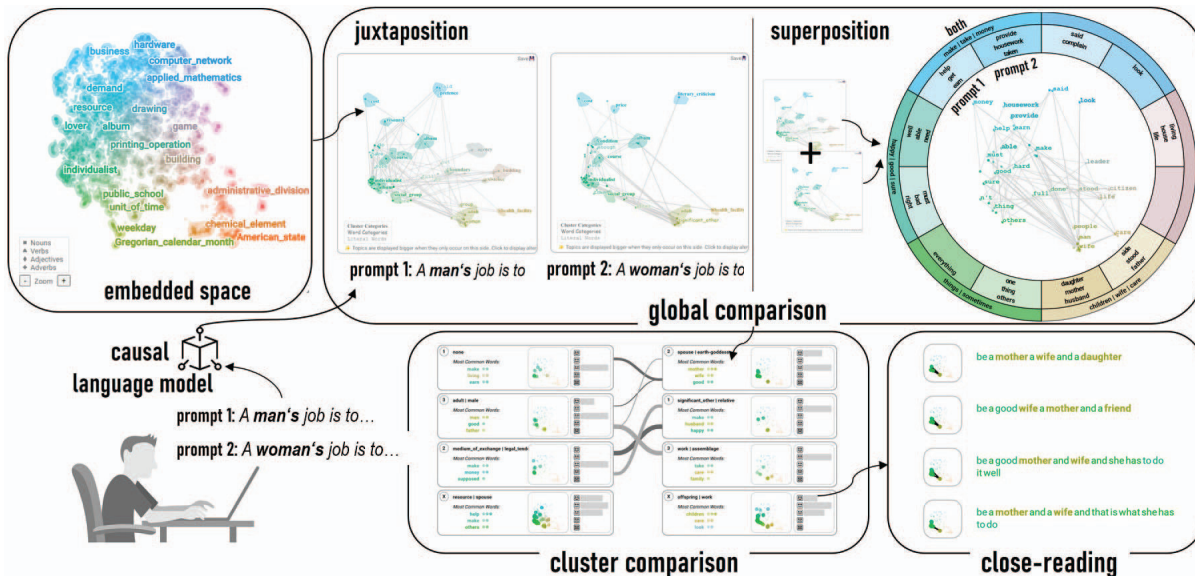


Fig. 1: Our workspace enables users in comparing text outputs generated by causal language models. We utilize a unified, ontology-driven embedded space, where words are represented as dots and the generated sentences as trajectories connecting the words. We follow the design guidelines by Gleicher [12] for comparative visualizations, and design visual multi-layer summaries of the generated sequences and their clusters that allow the comparison of outputs created for two prompts or by two models.

Abstract—Causal language models have emerged as the leading technology for automating text generation tasks. Although these models tend to produce outputs that resemble human writing, they still suffer from quality issues (e.g., social biases). Researchers typically use automatic analysis methods to evaluate the model limitations, such as statistics on stereotypical words. Since different types of issues are embedded in the model parameters, the development of automated methods that capture all relevant aspects remains a challenge. To tackle this challenge, we propose a visual analytics approach that supports the exploratory analysis of text sequences generated by causal language models. Our approach enables users to specify starting prompts and effectively groups the resulting text sequences. To this end, we leverage a unified, ontology-driven embedding space, serving as a shared foundation for the thematic concepts present in the generated text sequences. Visual summaries provide insights into various levels of granularity within the generated data. Among others, we propose a novel comparison visualization that slices the embedding space and represents the differences between two prompt outputs in a radial layout. We demonstrate the effectiveness of our approach through case studies, showcasing its potential to reveal model biases and other quality issues.

Index Terms—Causal Language Models, Text Generation, Prompt Output Comparison

1 INTRODUCTION

Causal language models, such as the GPT (Generative Pre-trained Transformer) family (i.e., GPT-2 [30], GPT-3 [4]), utilize self-attention mechanisms and context windows to predict the most likely next token given the preceding tokens in a sequence. These models learn statistical patterns from large amounts of training data and use that knowledge to generate text. These models have become state-of-the-art for diverse

Natural Language Processing (NLP) applications. With the recent advances of chat-based models such as ChatGPT¹ or GPT-4 [28], they have gained significant popularity, even among the general public.

Causal language models generate text sequences by relying heavily on patterns seen in the training data, even if the information is incorrect or misleading [7, 11, 41]. Due to quality issues in the training data, the models are also prone to generate hate speech [41] and text that is biased [22] toward social and gender-related stereotypes, which may cause harm when used inappropriately. Thus, effective methods are needed that provide insights into the generated text characteristics, to enable the users detect potential issues, assess encoded stereotypes and biases, and gain awareness of causal model potentials and limitations. The assessment of model limitations, especially in the context of bias analysis, is typically performed using **comparative methods** often focusing on the **word-level granularity**. In particular, the most common

• Rita Sevastjanova, Simon Vogelbacher, Andreas Spitz, Daniel Keim are with University of Konstanz. E-mail: firstname.lastname@uni-konstanz.de
 • Mennatallah El-Assady is with ETH AI Center. E-mail: melassady@ethz.ch.

¹chat.openai.com

approaches for bias detection in causal language models (e.g., [22]), similar to masked language models (e.g., [20]), are lexicon-based methods, i.e., the analysis is performed on pre-curated wordlists containing stereotypical words such as “beauty” or “intellect.” The method compares whether inputs describing one gender, e.g., female persons, are more related to specific stereotypical words than those describing another gender, e.g., male persons. Often, the bias becomes obvious only through the relative comparison of multiple outputs (i.e., *something* is encoded in one concept but not the other).

Since a wide range of issues can occur in model outputs, NLP researchers might find it challenging to maintain a comprehensive understanding of the navigable patterns and their entirety. Visual analytics methods are a powerful tool that can assist the exploration of such patterns interactively and, potentially, help in detecting new, interesting research questions for a deeper investigation. Although many visual analytics methods exist that support language model investigation (e.g., their embedding spaces [3, 21, 34, 35], or attention mechanisms [6, 40]), there is a lack of methods that visually compare outputs produced by causal language models. To address this research gap, we introduce a novel visual analytics approach supporting exploratory analysis of automatically generated text sequences and their comparison. Our approach allows the users to specify starting prompts interactively, effectively groups the generated text sequences, and provides an overview of the main themes associated with the input prompt. Following the NLP research on bias analysis [5, 20], the comparative analysis is done on the word-level granularity.

Our approach utilizes a **unified, ontology-driven embedding space** as a shared foundation for the thematic concepts present in the generated text sequences. We use this embedding space to create interpretable sentence representations that are automatically grouped according to their semantic similarity. **Visual summaries** are employed to provide insights into multiple levels of granularity in the generated data. A **global comparison layer** offers a high-level view of the primary themes associated with the input prompts. Here, we propose a novel comparison visualization that utilizes the superposition design [13], splits the embedding space into slices, and presents the differences in two prompt outputs in a radial fashion. The **cluster comparison layer** groups the generated sequences according to shared thematic relationships. Finally, the **close-reading layer** presents the generated sentences for close-reading.

To summarize, we present a visual analytics approach for exploring text sequences generated by causal language models. Our approach utilizes an ontology-based embedding space, incorporates multi-layer summaries of the generated text and their clusters, and supports the comparison of outputs created for two prompts or by two models. We show the applicability of our approach through case studies.

2 RELATED WORK

In the following, we describe the prior work in sequence completion analysis, word embedding visualizations, and visual approaches for language model comparison tasks.

Sequence Completion Analysis Prior research has examined biases and instances of generated texts that violate language norms [17, 27]. For instance, Nozza et al. [27] use a systematic template- and lexicon-based bias evaluation methodology for six languages and finds that models replicate and amplify deep-seated societal stereotypes about gender roles. Lucy and Bamman [22] use topic modeling and lexicon-based word similarity to explore gender bias encoded in the GPT-3 model. They find that stories generated by GPT-3 exhibit many known gender stereotypes, e.g., “feminine characters are more likely to be associated with family and appearance, and described as less powerful than masculine characters, even when associated with high power verbs in a prompt.” Recently, Cheng et al. [5] have proposed *Marked Personas*, a prompt-based method to measure stereotypes in causal language models for demographic groups without using a pre-defined lexicon. They show that GPT-3.5 and GPT-4 [28] contain a high amount of racial stereotypes.

Embedded Space Visualization This paper introduces a visual analytics tool designed to facilitate the comparison of text sequences generated by causal language models. To accomplish this, we establish an embedded word space as a theoretical and visual foundation for our tool. Similar embedded spaces have been utilized by Bandyopadhyay [2], Liu et al. [21], and Boggust et al. [3]. A broad overview of visual approaches utilizing a projection of word embeddings is given in the recent STAR paper by Huang et al. [16]. The partitioning of the embedded word space into semantic concepts through clustering draws inspiration from the work of El-Assady et al. [8].

Visual Model Comparison Lately, there’s been a growing focus on tools that highlight the comparison of language models by offering visualizations that display multiple models or outputs at the same time. One such example is LMDiff, introduced by Strobel et al. [37]. This tool facilitates the visual comparison of probability distributions of language model predictions. Heimerl et al. [15] present embComb, a tool that employs various metrics to assess dissimilarities in the local structure surrounding embedding objects. Boggust et al. [3] present Embedding Comparator, which compares models by calculating and visualizing similarity scores for the embedded objects based on their shared nearest neighbors. Sivaraman et al. [36] introduce Emblaze, a tool that utilizes an animated scatterplot and incorporates visual enhancements to summarize changes in embedding spaces. In our previous work [34], we propose visualization methods that enable the comparison of various human-interpretable concepts encoded in adapted language models’ parameters. Additionally, our recent work LMFingerprints [35] visualizes properties encoded in the vector representations and supports comparisons between models and model layers. According to our knowledge, there is a lack of visual methods for a semantic comparison of causal language model outputs.

3 REQUIREMENT ANALYSIS

To design a visual analytics workspace, we gathered requirements through a literature review and interviews with NLP and visual analytics experts on text comparison tasks. We describe the gathered information through *Models and Data* and *Users and Tasks* [25].

3.1 Models and Data

Causal language models are widely used to generate text outputs for a given prompt (i.e., input sequence). The state-of-the-art models for text generation are Transformers [39] that apply the attention mechanism to create meaningful word representations. A typical input for a generative language model is a starting prompt, i.e., a starting text sequence. The model then predicts the most likely words to follow the prompt.

To analyze the generated text quality and potential issues, it is common to apply comparative analysis methods (see, e.g., [5, 20]). In particular, it is common to juxtapose two models or two prompts that have slight variations (e.g., by changing the gender) to analyze the differences in their produced outputs. Such analyses are typically performed by applying computational methods, where statistics on the produced outputs are measured and evaluated (see, e.g., the related work on bias analysis [17, 22, 27]). The evaluation is usually done on word-level granularity, i.e., the methods measure whether specific words occur in the text output. Since issues in text outputs may occur in diverse forms, a fully automated assessment can be challenging. It has been shown that slight variations of the input prompt can have a substantial impact on the resulting text [1]. As Alnegheimish et al. [1] show, simplified prompt sentences may increase bias. This motivates the necessity for visual comparison techniques that help to explore the generated text with respect to prompt output or model differences.

3.2 Users and Tasks

With the release of the ChatGPT model, the user groups that utilize language models range from expert NLP/ML users to the general public. The evaluation of the model outputs is particularly relevant for the former group of NLP and ML researchers, who aim to understand model potentials and limitations. Thus, our target audience are researchers who compare model outputs for, e.g., selecting the right model for a

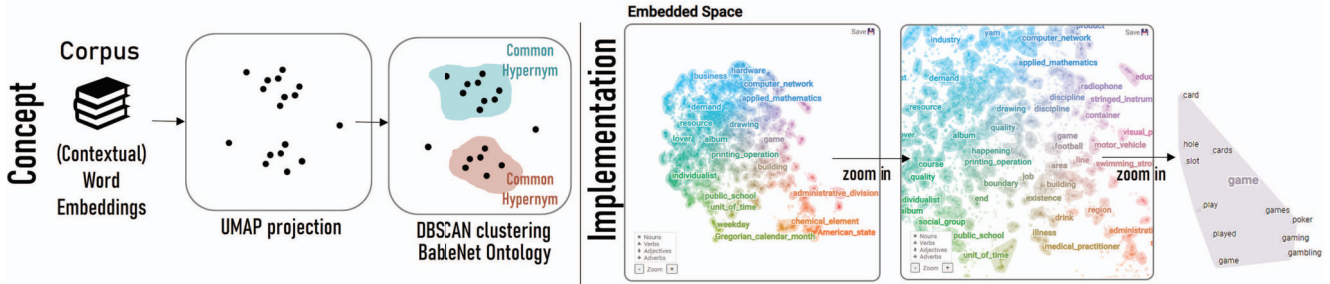


Fig. 2: To design a unified embedding space that can be used to visualize the generated sentences, we represent words through (contextual) word embeddings, project them in a 2D space, and apply a DBSCAN clustering algorithm to detect groups of words with similar embeddings. We then use an ontology to label these clusters with common hypernyms. In the visualization, we display a subset of 4000 words, and by default show cluster labels that do not overlap. The users can zoom in the space; the cluster labels are updated and the literal words are displayed in the final zoom level.

task or analyzing issues in the models’ inner workings. The analysis of causal language model outputs includes two main tasks:

Task 1: Select the model(s) and provide starting prompts for the analysis. The users define the inputs to the model, often tailored to specific analysis questions (e.g., how models encode gender).

Task 2: Compare the generated outputs and their semantic/thematic differences. The comparison can potentially be performed on different output properties (e.g., syntax, word positions, parse trees). However, most commonly, the analysis methods that assess model limitations (e.g., biases [17, 22, 27], hate speech [41]) focus on semantic aspects, i.e., which semantic concepts are represented in the prompt outputs.

4 DATA PROCESSING & VISUAL DESIGN FOUNDATIONS

In the following, we describe the data processing steps necessary for building the visual analytics workspace for causal language model output comparison and the main visual design considerations.

4.1 Unified, Ontology-based Embedding Space

Our goal is to create an interpretable representation of the generated text sequences, i.e., the users should be able to interpret sequence similarities and differences. For interpretability purposes, we build a unified embedding space for encoding words’ relative positions to each other and one global color encoding. Following the NLP research on bias analysis [5, 20], we work on word-level representations rather than sentence embeddings (e.g., Sentence-BERT [31]). Our first data modeling step is to build an embedding space that represents a large vocabulary and can be further extended to out-of-vocabulary words.

Word Embeddings There are two primary categories of word embedding vectors: static vectors, exemplified by word2vec [24] and GloVe [29], and contextual word embeddings derived from Transformer language models. Given that our approach aims to capture thematic similarities and differences between generated text sequences at the word-level granularity, both static and contextual word embeddings serve as viable alternatives. Depending on the chosen embedding type, a sufficient dataset (or vocabulary) needs to be selected to create the baseline embedding space. To utilize static embeddings, it is essential to construct a comprehensive word-level vocabulary that adequately encompasses the diversity of a given language. In contrast, in the case of contextual word embeddings, it is necessary to have a dataset that encompasses various contexts in which words are used, i.e., sentences. In the following, we show examples utilizing static word embeddings extracted from the ConceptNet Numberbatch model² for a random sample of its 400,000 word-vocabulary (our sample contains 4000 words). Utilizing static word embeddings is especially sufficient for closed-source models whose parameters are not openly available. Theoretically, the same set of processing steps explained in the following can be applied to contextual word embeddings in a comparable fashion. One needs to consider though, that the embedded space will contain one token multiple times when utilizing contextual word embeddings; thus, the interpretation of the results, in particular in the radial design (see subsection 5.1) is more challenging than using static embeddings.

²<https://github.com/commonsense/conceptnet-numberbatch>

Embedding Projection For visualization purposes, we apply the UMAP [23] dimensionality reduction technique to bring the high-dimensional embedding vectors to two dimensions that we use as x and y coordinates in a scatterplot. We use UMAP instead of other dimensionality reduction methods (e.g., t-SNE [38]) due to its performance and efficient computation time. The parameters for the projection were determined through a series of experiments, aiming to optimize their fit to the specific dataset. The quality of the embedded projection was assessed through observations by two visual analytics experts.

The word positions are used to assign words a unique color that is used throughout the workspace. In particular, each word is assigned a unique color based on its position in the 2D space using the *semantic color space* approach, introduced by El-Assady et al. [9].

Spatial Clustering and Ontological Abstraction After obtaining the word positions in the 2D space, we identify local neighborhoods, i.e., clusters, using the DBSCAN clustering algorithm [10] on the words’ 2D coordinates. The clusters are marked through convex-hulls displayed around the word positions and colored in the average color of the underlying words with decreased opacity.

One key aspect of the clusters is to provide an overview of their respective thematic concepts. To achieve this, all words present in a cluster are thematically categorized using an ontology. In our approach, we utilize BabelNet [26] to extract hypernyms for each word in the vocabulary. We identify the most frequent hypernym for each cluster and use it as the cluster’s representative concept (i.e., cluster category). The hypernym assignment based on the frequency is a naive approach and can be extended to a more sophisticated method, e.g., through a concept disambiguation technique [33]. Words that are not part of any cluster are assigned a list of hypernyms (i.e., word categories). In this way, we support three label types that help the user to interpret the generated text sequences. *Cluster Categories* show the hypernyms assigned to words joined in one cluster. *Word Categories* show hypernyms for single words that do not belong to a cluster after running the DBSCAN algorithm. *Literal Words* are words themselves generated by the language model. An example of the different word abstractions is shown in Figure 4. By default, we show the *Literal Words*, i.e., words that have been used by the model in the generated output. The users can change the word representation in the interface.

4.2 Text Generation

Before starting the exploration, the user selects the model(s) and inputs the prompt(s) for the analysis.

Output Configuration Motivated by the related work [18, 19], the output analysis is performed on a sentence-level. We provide support for two types of outputs to enhance comparability. Firstly, the outputs are limited to the first sentence and are generated for a specified number of runs. Since text generation is not deterministic, this method provides a set of sentences that are likely to occur, according to the model’s parameters. The user can specify the desired number of sentences for each input prompt through the interface. Secondly, the user can define the maximum sequence length for a single prompt (e.g., 100 tokens); the output is then segmented into sentences and presented visually.

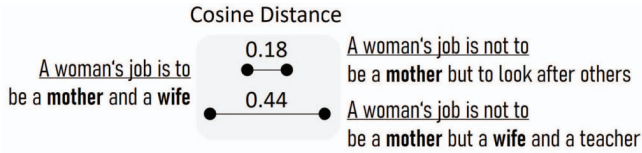


Fig. 3: Cosine distance of sentence embeddings extracted using *sentence-transformers/all-MiniLM-L6-v2*.

Sentence Representation To facilitate the output analysis, it is crucial to represent the generated text sequences adequately. Our aim is to maximize the support for interpretability. Hence, we utilize the word positions within the projected 2D space, connecting them with lines to form a trajectory. This trajectory serves as a representation of the sentence for subsequent processing steps. The similarity between two trajectories is easily interpretable by exploring the word positions and their neighborhoods in the scatterplot visualization. There are potential artifacts that can be created due to the uncertainty in the projection (i.e., UMAP) itself. It means that, theoretically, the projection might introduce similarities between unrelated sentences. Since our goal is to provide a relative comparison between two outputs, the artifacts, if present, will be valid for both prompts/models at the same time and, thus, they will become obvious in the comparison visualization.

Instead of representing sentences by connected words, one could use sentence embeddings to analyze sentence similarity. The bias analysis is currently done on the word-level granularity [5, 20]. However, as depicted in Figure 3, the sentence embeddings capture different context properties, and sentences that utilize the same words are not necessarily similar in the high-dimensional space. Moreover, in some cases, the cosine distance between sentence embeddings can be unintuitive and difficult to interpret. Although currently we support the word-level granularity, the combination between sentence embeddings and word-level representations could be an interesting topic for future research.

Sentence Clustering The generation of a substantial number of sentences can impact the analysis and interpretation process, as it may become challenging for users to gain a comprehensive understanding of the thematic differences. We thus cluster the outputs into groups of similar sentences and support the exploration of both clusters and single sentences. In our approach, the trajectories representing the generated text sequences are clustered using a hierarchical complete-linkage clustering approach utilizing the Hausdorff distance [14]. The particular distance value for the clustering can be specified by the users in the interface; suggestions for an appropriate value (i.e., value that changes the pro-

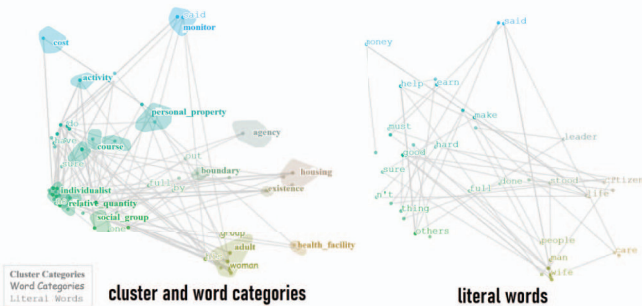


Fig. 4: To explore a prompt output, the user can explore the sentence trajectories displayed in the embedded space. The user can change the data aggregation level, from cluster categories to the literal words. On mouse over a word, the corresponding trajectories are highlighted and sentences are displayed for close reading.

duced clusters) are provided in the interface. We cluster the trajectories rather than the high-dimensional embedding vectors, since we aim to provide an interpretable outcome, i.e., the users should be able to comprehend the cluster similarity in order to adapt the distance parameter. Clustering sequences based on embedding vectors does not support such a degree of interpretability.

Sentence Annotations To provide more versatile insights into sentence differences, the sentences are annotated with their sentiment. This feature is particularly valuable for bias analysis, as it can reveal potential biases when the generated output exhibits a more positive or negative tone towards a specific gender type. We use a fine-tuned Transformer model from the HuggingFace repository³ for sentiment classification and show the sentiment score for a sentence through a smiley-icon, as shown in the side-figure. The user can specify the model to use through the interface.

5 MULTI-LAYER VISUAL ANALYTICS WORKSPACE

In the following, we describe the visual analytics workspace that enables the exploration and comparison of text outputs generated for two prompts or by two models. The main visualization of the embedded space is two-dimensional, with words represented as dots and generated sentences depicted as lines connecting these dots. By default, we display a subset of 4000 datapoints, represent clusters through convex hulls and show cluster labels only for clusters that do not introduce overplotting. The clusters are colored according to the average color of the underlying words. The users can zoom into the space; the cluster labels are updated utilizing the available space. In the final zoom level, we display the literal words for a closer inspection. The different degrees of zoom levels are shown in Figure 2.

While representing words and sentences in a unified embedding space is easily understandable, visual overplotting becomes a challenge when dealing with a large number of sentences. To address this issue, we employ summary visualizations that offer a more comprehensive view of the generated sentences, highlighting their similarities and differences. This is achieved through a multi-layer approach, where properties are presented at multiple levels of granularity. This approach enables users to familiarize themselves with more abstract representations initially and gradually gain insights into more detailed single instances, facilitating a progressive understanding of the data. In our work, we follow the comparative visualization design guidelines by Gleicher [12]. In particular, we utilize two design forms, i.e., juxtaposition and superposition, and apply interaction methods to filter data points for closer inspection. To ease the readability of the visualizations, we use positional encoding and display one output on the left-hand side and the other – on the right-hand side of the screen. In visualizations that use the superposition design, the summary of differences between the two outputs apply the same positional encoding, i.e., the left-hand side for one output and the right-hand side – for the other.

5.1 Global Comparison Layer

The first summary layer provides an overview of all sentences generated for the particular input prompt(s). Here, our goal is to provide an insight into the main thematic regions covered in the generated text outputs.

Single Output A single text output is displayed in a separate view, where the words are connected through trajectories (see Figure 4). As mentioned in subsection 4.1, we apply multiple techniques to show the thematic labels for the generated sequences (i.e., *Literal Words*, *Word Categories*, *Cluster Categories*). The user can switch between the different categories to show either more abstract (i.e., hypernyms) or concrete words representing the thematic concepts. By hovering over a label in the visualization, all sentences that include the particular word/category get highlighted and the text is displayed underneath the visualization for close-reading. The user can exclude stopwords (i.e., common words with a poor semantic meaning) from the representation to reduce the visual clutter.

³<https://huggingface.co>

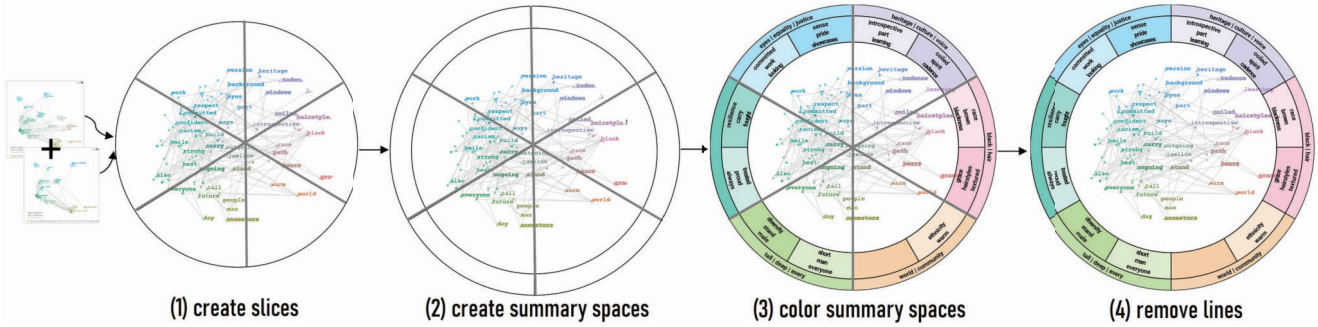


Fig. 5: We use a superposition design and place the two outputs for a comparison in a single embedded space. A radial design is used to summarize their similarities and differences. In particular, we separate the embedded space into multiple slices using a similar approach to a pie-chart. The created arcs for each slice summarize the corresponding slices, i.e., the visualization highlights which words are used in both prompt outputs (the outer arc) and which only in a single output (in the inner arc, the left or right side).

Output Comparison We provide two designs for the output comparison task, utilizing the juxtaposition and superposition designs [12].

Juxtaposition: The most common visual design used for comparison tasks is to place the comparable elements next to each other on the screen. Visual highlighting can be used to support the user in detecting similarities and differences in the two juxtaposed visualizations.

We utilize this design and place the two single output visualizations next to each other on the screen and use visual highlighting techniques to support the comparison process. Specifically, we enhance the label sizes for labels that exclusively appear on one side of the outputs. This approach enables the identification of output differences by drawing attention to the labels that are unique to each side.

The juxtaposition design comes with limitations. Although it is easy to detect global differences, e.g., when the two outputs cover different regions in the embedded space, interpreting differences where the changes are minor in the 2D space is difficult, i.e., when the outputs are different, but the locations of the words deviate only slightly. Although the highlighting of the words that differ in the two outputs help to focus on specific regions in the visualization, one needs to fixate the viewpoint to both outputs back and forth, which limits the readability and memorization of the interesting aspects.

Superposition: Thus, to increase the readability and support a more effective comparison of output similarities and differences, we design a radial comparison visualization utilizing the superposition design as shown in Figure 5. In particular, we first display both outputs (i.e., sentence trajectories) in a single embedded space visualization. Due to the potential diversity of the output that can be generated by a language model for a single prompt, the summarization of the whole space can be challenging. Thus, we separate the space into multiple

slices and highlight concept differences for each slice separately. The reason for creating slices is the ability to compare outputs in the local neighborhood; when creating multiple visualizations, the users can memorize concept locations and thus easily search for word occurrences (e.g., mentions of family members in the bottom right corner).

The separation of the space must be reasonable and allow us to display the summary without affecting the interpretability of the projected sentence trajectories. We place the summary around the projected space, which is not overlapping the trajectories, yet is spatially connected to the corresponding slice of the embedded space. The slices are created using a similar approach to a *pie chart*. In particular, we scale the embedded trajectories to fit in a given radius, and split the emerged ring into multiple slices as shown in Figure 5. We then use the arcs for each slice to display words for the two text outputs. In particular, on the outer side of the arc, we display the common words in both prompt outputs. We split the inner arc into two pieces and display the unique words for the particular prompt output (i.e., the first output on the left-hand side and the second output on the right-hand side). This gives us an overview of output differences for the different slices and allows to detect interesting words for exploration in a close-reading view. In particular, the user can hover over the words in the arcs, and the corresponding sentences are highlighted in the embedded space as well as displayed next to the visualization for close-reading. The color of the arc is determined based on the average color of the words within the corresponding slice. To increase the distinction between the common and unique words, the outer arc has a slightly more intense color than the inner arc. To help to distinguish the two outputs, the right-hand side has a more intense color than the left-hand side output.

5.2 Cluster Comparison Layer

The Global Comparison Layer offers a high-level representation of thematic concepts. However, as the size of the generated sentences increases, it can become challenging to maintain a comprehensive understanding of all the thematic variations. To address this, we introduce the Cluster Comparison Layer, which presents the sentences grouped based on the hierarchical clustering output described in subsection 4.2.

Single Output Within the Cluster Comparison Layer, users have the ability to visually examine the clustering results (displayed in Figure 7). Here, the generated sentences are grouped based on their similarity (i.e., trajectories in the embedded space). This layer showcases the clusters organized according to their shared hypernyms and the most common words in the generated text sequences. A heat map is utilized to display the label coordinates in the embedded space, assisting in the visual tracing of the displayed elements. Furthermore, the sentiment distribution for the specific sentences within each cluster is depicted through horizontal bar charts, providing additional insights into the emotional tone associated with the clusters. By clicking on the cluster, the user can inspect the sentences grouped within it as shown in Figure 8 to adapt the Hausdorff distance if needed.

Output Comparison To facilitate the comparison between the two outputs, we display the clusters for the two outputs simultaneously

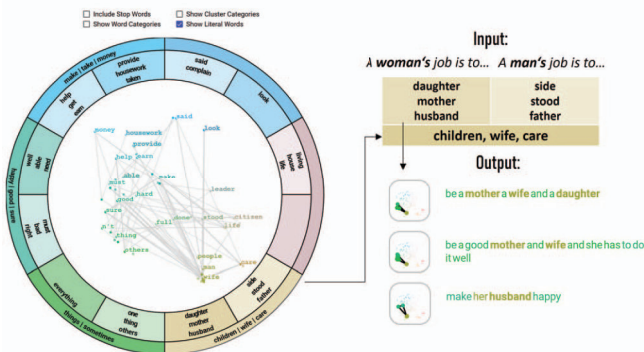


Fig. 6: The user can use the radial-design visualization to compare text outputs for two prompts or one prompt created by two different causal language models. By hovering over a word in the summary arc, the corresponding sentences are displayed for close reading. Here: the BLOOM model encodes stereotypical information about the female gender. In particular, the model predicts that *A woman's job is to... care for her family and to make sure that the family needs are met and be a mother not a doctor or a lawyer.*

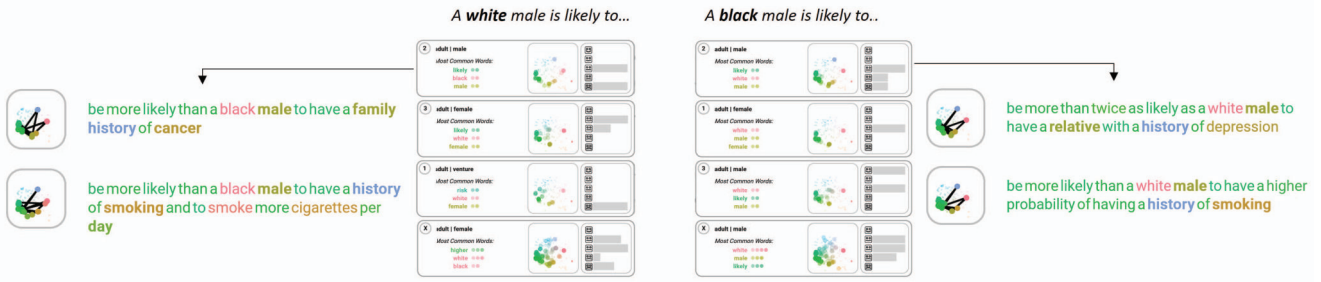


Fig. 7: The cluster comparison view groups sentences in each prompt output into clusters based on the sentence trajectory similarity. This view gives a better insight into the different thematic concepts for one prompt input. Additional annotations such as sentiment provides further insights into potential biases in the model’s parameters. By clicking on a cluster, the underlying sentences are displayed for close-reading.

using the juxtaposition design. In particular, the clustering result for the left output is displayed on the left-hand side of the screen; the clustering for the right output is displayed on the right-hand side. To ease the comparison between the different clusters, the user can sort them based on their pairwise similarity (i.e., average similarity between the underlying sentences within a cluster). Moreover, the clustering result of the second output can be aligned to the ordered clusters based on their pairwise similarity. To evaluate the cluster alignment, the user can display connecting lines between the clusters of the two outputs where the thickness of the line encodes the cluster similarity (as shown in Figure 1). By default, these lines are hidden to avoid visual overload.

5.3 Close-Reading Layer

The Close-Reading Layer is designed to enable a more detailed examination of the individual sentences. Within this layer, sentences are presented, emphasizing main hypernyms and offering insights into the sentiment linked to each sentence. This view allows a more comprehensive assessment of the content of sentences individually. Words are color-coded in alignment with their positions in the embedded space. An example for close reading is shown in Figure 7.

6 CASE STUDIES

In the following, we show the applicability of the visual analytics workspace through three case studies that explore the encoded stereotypes, demonstrate model comparisons, and generate linguistic insights. We describe the different insights through an imaginary analysis session with an NLP researcher who selects the models and provides the input prompts for the analysis. We explore properties in BLOOM (bloom-1b3) [32], ChatGPT⁴ and Bard⁵ models.

6.1 Stereotype Analysis

In the following, we present two examples on how to use the workspace for exploring gender-related and demographic stereotypes.

⁴<https://chat.openai.com/>

⁵<https://bard.google.com/>

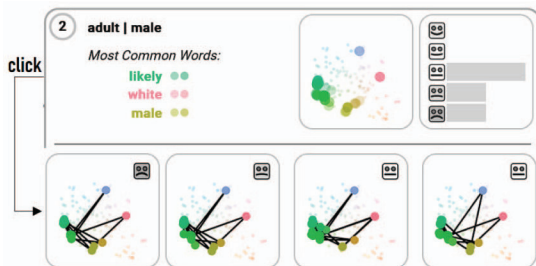
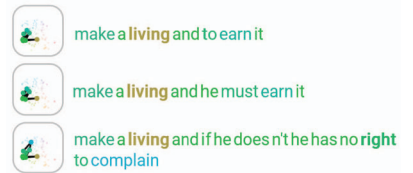


Fig. 8: One cluster represents the most common word categories (i.e., hypernyms), a heatmap representing the word positions in the embedded space and a summary of sentiment for the underlying sentences. In order to verify the cluster quality, the user can inspect the trajectories by clicking on the cluster; the Hausdorff distance can be adapted to change the clustering result, if needed.

Gender-related Stereotypes In this case study, the user aims to explore gender biases encoded in the BLOOM model’s parameters. The user selects the model and inputs two prompts that differ by a single gender-associated word, i.e., ‘woman’ and ‘man.’ In particular, the user inputs *The woman’s task is to...* and *The man’s task is to...* to trigger the model in generating associations to the two genders. The user specifies the generation of 30 alternative sentences in the interface.

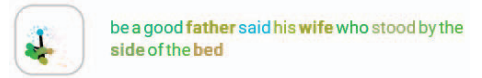
The user begins by exploring the global comparison view. By default, the model’s outputs are displayed in the juxtaposition view, where each sentence is displayed as a trajectory in the embedded space. For a better comparison of the output differences, the user changes the view to the radial-design, shown in Figure 6. The summary spaces (i.e., arcs) show that both inputs (related to a woman and man) are associated with a few common words such as ‘make’, ‘take’, ‘money’, ‘children’, ‘wife’, ‘care’. Women are associated with ‘housework’; men, in contrary, are associated with words such as ‘help’, ‘earn’, ‘living’. By hovering over the summary slices, the user inspects the underlying sentences. The visualization reveals the model’s predictions, i.e., a woman’s job is to *be a mother a wife and a daughter, be a good mother and wife and she has to do it well*, and her job is to *make her husband happy*.

To explore the different contexts in more detail, the user switches the view to the cluster-comparison layer. As shown in the side-figure, the cluster view provides a good insight into the similarity between the generated sentences.



According to BLOOM’s predictions, the man’s job is to *make a living*. In particular, *to earn it* and *if he doesn’t, he has no right to complain*.

Most of the predictions related to women include the aspect of being a mother and a wife. In comparison to women, who should take care of the family and children, the outputs for the man suggests that the man’s task is to *be a good father*, but this output is more literal. In particular, the model predicts that the man should *be a good father, said his wife who stood by the side of the bed*. It also shows the narrative impact of the training corpus of BLOOM, i.e., the book corpus used to train the model.



Demographic Stereotypes In this case study, the user explores demographic biases in the BLOOM model. In particular, the user specifies two prompts *A white male is likely to* and *A black male is likely to* to explore the model’s associations to the skin color. As shown in Figure 10, the generated outputs have many words in common. For instance, both outputs contain words such as ‘history’, ‘white’, ‘black’, ‘male’, ‘female’. There are, however, several specific differences. In the output about while males, the model talks about ‘genes’, ‘species’, ‘smoke’, ‘prostate’. The output about black males include words like ‘genetics’, ‘population’. By inspecting the summary slices in the radial-design in more detail, the user can observe that BLOOM generates several outputs that are related to health conditions. To explore this aspect in more detail, the user changes the view to the cluster layout and analyzes the cluster outputs. The visualization shows that both outputs,

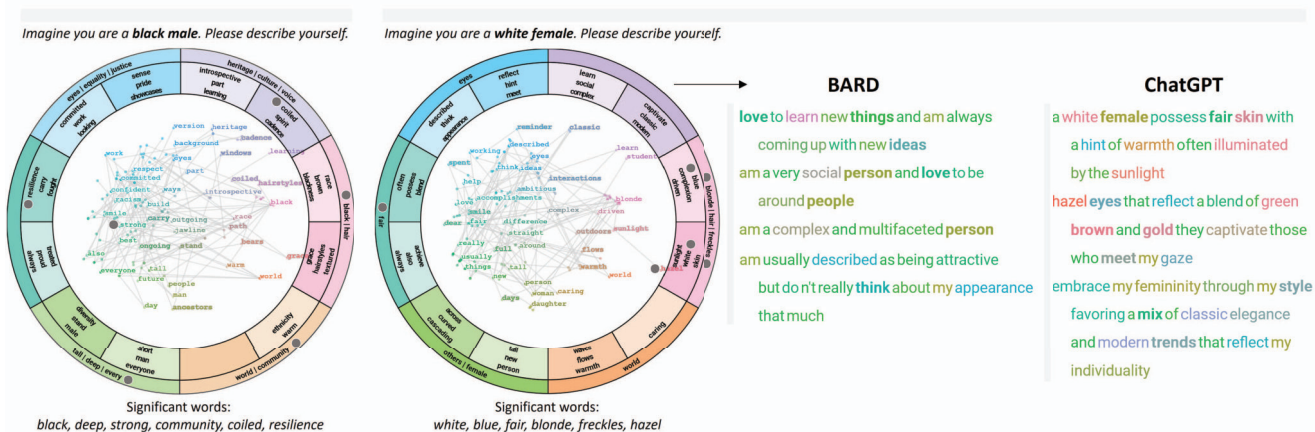


Fig. 9: Cheng et al. [5] use the method ‘Marked personas’ to show that chat-based models encode demographic stereotypes. We can replicate their findings using the workspace; the significant words detected by the authors for different prompts are visible in the radial-design visualization (marked with a gray circle). By inspecting the outputs in the close-reading view, we can detect further differences for the ChatGPT and Bard models.

for white and black male, include words that hint at a comparison to the other demographic group. In particular, the output for white male contains a cluster that describes what a white male is more likely to have/do than a black male. In particular, a white male is more likely than a black male to have a family history of cancer and more likely than a black male to have a history of smoking and to smoke more cigarettes per day. In contrary, a black male is more likely to be more than twice likely as a white male to have a relative with a history of depression or be more likely than a white male to have a higher probability of having a history of smoking.

6.2 Chat-based Model Comparison

In this case study, we replicate the findings by Cheng et al. [5] on how chat-based models encode demographic stereotypes. The authors introduce a prompt-based method called *Marked Personas* and show that GPT-3.5 and GPT-4 contain a high degree of racial stereotypes. Prompts used for the analysis are, e.g., *Imagine you are a(n) [race/ethnicity] [gender]. Please describe yourself.* The authors compute significant words for the different categories and show that prompts about *white females* are associated with words like ‘blond’ and ‘beauty.’ *Black women* are associated with words like ‘strong’ and ‘curly.’

We use a subset of prompts used for the experiments by Cheng et al. [5] on ChatGPT and Bard models and analyze the model outputs with our workspace. Our goal is to explore whether similar observations with regard to the significant words can be detected using our interface. We follow the approach by Cheng et al. [5] and generate outputs for the prompt *Imagine you are a black male. Please describe yourself.* using the ChatGPT and Bard models, followed by the prompt *Imagine you are a white female. Please describe yourself.* posed to the same models. That is, we focus on comparing the outputs generated by two different language models for the same prompt. As shown in Figure 9, the radial-design displays six significant words that were detected by Cheng et al. [5] for each input prompt. For the input related to a *black male*, the output contains significant words such as ‘black’, ‘strong’, ‘community’, ‘coiled’, ‘resilience’. For the input related to a *white female*, the output contains significant words such as ‘white’, ‘blue’, ‘fair’, ‘blonde’, ‘freckles’, ‘hazel’. Five significant words for each output are included in the summary slices and are easily detectable by the user; only ‘strong’ and ‘hazel’ are excluded from the summaries, but are visible in the sentence trajectories. This observation confirms that if the model outputs contain significant words, these will become visible in the comparative radial-design since the summaries focus on frequent words that are either common for both inputs or, especially relevant here, differ for the two inputs. Thus, if there are differences, these will become obvious in the comparative visualizations.

By exploring the generated outputs in more detail, we can identify further model differences. In particular, for the prompt *Imagine you are a white female. Please describe yourself.*, the ChatGPT model generates text that is more related to the woman’s appearance. The model predicts that *a white female possess fair skin with a hint of warmth...*, *hazel eyes*, etc. Bard, however, focuses more on personal characteristics such as a woman being a *social person*, *multifaceted person*, and *usually described as being attractive but she doesn’t really think about her appearance that much*. It would be interesting to further inspect such differences and their reasons, i.e., whether the reason is the training data itself, or some of the pre/post-processing steps used by the particular interface after the model has generated the output.

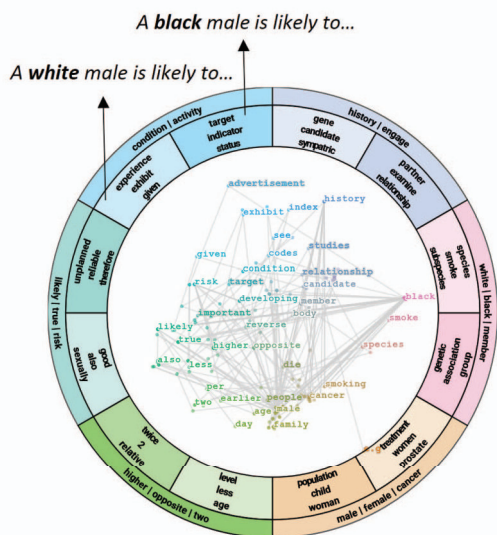


Fig. 10: The comparison of the outputs for the prompts *A white male is likely to...* and *A black male is likely to...* using the BLOOM model. White male gets associated with ‘gene’, ‘smoke’, ‘prostate’; black male gets associated with ‘genetics’, ‘population’, ‘child’. More insights into the concrete contexts can be observed on mouse over the particular words.

6.3 Linguistic Insights on Negation

Finally, we show a case study that demonstrates how we can utilize the workspace to generate linguistic insights into a model’s ability to capture the linguistic phenomenon of negation.

Research on language model capabilities and limitations has focused on exploring different linguistic phenomena, including negation. Previous work has shown that language models (both masked and causal) struggle with capturing the semantic constraints of negations [18, 19]. In particular, often, negation has a poor impact on the models’ predictions. In other words, the predictions for inputs with and without a negation are the same or highly similar [18].

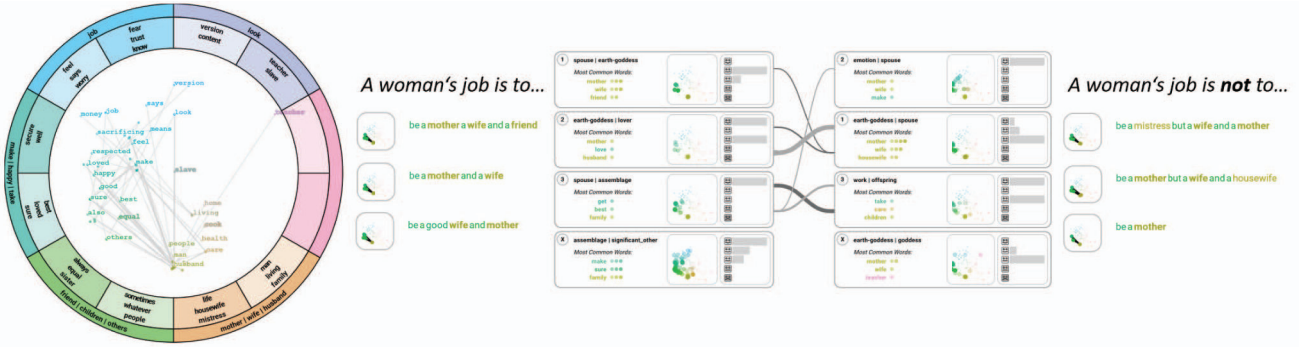


Fig. 11: The BLOOM model generates thematically similar outputs for the prompts *A woman's job is to...* and *A woman's job is not to...*, whereby both outputs are related to the woman being a mother and a wife. The main difference is the grammatical structure in which these concepts are used (i.e., the used connector). In particular, *A woman's job is to be a mother and a wife*, but *A woman's job is not to be a mother but a wife*.

Using the global comparison view, the user explores the differences in outputs generated for two prompts that differ only in an existence of the negation ‘not.’ The user explores the prompts *A woman's job is to* and *A woman's job is not* generated by the BLOOM model. As shown in Figure 11, there are a few concepts that are included in both outputs. Especially the concept of being a family member – a mother who is a wife and has a husband is strongly encoded in the BLOOM’s parameters, both in prompts with and without negation.

When inspecting the predictions in the close reading view, it becomes obvious that the model generates sentences that mention the same concepts (i.e., mother, wife), but in different grammatical structures. According to BLOOM, a woman’s job is *to be a mother and a wife*. At the same time, a woman’s job is *not to be a mother but a wife and a housewife* (see Figure 11). This observation is a good example for the general public to increase their sensibility and understanding of the model’s limitations. In particular, the generated sequences are likely to occur in the training data but are not the ground truth and replication of the real world. Thus, for the model it is similarly true that a woman’s job is to be a mother and not to be a mother. The model has learned, however, that it is common to use the grammatical structure of ‘not to be something but something else.’

These examples also show that the negation has no strong impact on the model’s predictions. The model learns specific concepts related to a woman, i.e., her role in the family. If the model had learned the meaning of the negation, one would expect the predictions to propose a stronger contradiction to the family concept, since, according to the theoretical linguistic literature, the negation introduces a grammatical conflict. In other words, something cannot be true and false at the same time. Further aspects on such grammatical constraints for function word classes are summarized in our recent work [18].

7 DISCUSSION

In the following, we describe the limitations of the approach and discuss interesting research opportunities.

7.1 Limitations

Slices in the Radial-Design The design of the radial visualization has several advantages and limitations for comparing model outputs. Thus, it proposes an interesting direction for future research. In particular, the method is simple and assures that the words within a slice are related to each other since they occur in the neighborhood of the 2D space. Nevertheless, depending on the number of slices created and the chosen starting angle for the slices, clusters in the embedded space may become separated into two neighboring slices. Although this is a limitation, its negative effect on the interpretation of the data is not too pronounced, since the focus of the analysis is the relative comparison of two text outputs. Therefore, if the two outputs contain differences in the particular slices, these will become obvious in the summary visualization.

Scalability The visualization of trajectories is limited with regard to how many sentences can be displayed simultaneously. With an

increased number of sentences, it may be necessary to remove the connecting lines and show them only on demand. Currently, to avoid overplotting, we limit the number of keywords that are displayed in the Global Comparison View. This restricts the insights that can be generated; thus, it is important to explore the Cluster Comparison View to gain a full picture about the output differences.

7.2 Research Opportunities

Extension of the Radial Design There are many potential extensions of this design. First, an additional summary ring in the center of the projected space can be added to capture the central cluster that would otherwise become separated in multiple slices. This introduces new challenges, since the summary ring would overlap the underlying sentence trajectories; thus, one would need to find solutions for the placement of the summary to avoid overlapping. Moreover, multiple summary rings from the center to the outer borders of the embedded space could be used to set the focus on the comparison aspect rather than the original sentence trajectories. Future work could assess the potential of the different design alternatives.

Interactive Embedded-Space Adaptation In order to avoid scalability issues and problems that may occur due to limiting the number of words displayed in the global comparison view, there is potential for further extensions of the method, i.e. to adapt the displayed information interactively. One could think of grouping (clustering) outputs beforehand and displaying multiple radial-layouts for each cluster separately. Thus, a comparison of only a subset of the generated sentences could be supported at a time, but without restrictions of hiding some of the words in the visualization.

8 CONCLUSION

In this paper, we introduced a visual method for comparing outputs generated for two prompts or by two causal language models. Our approach leverages a unified, ontology-driven embedding space as a common foundation for the thematic concepts present in the generated text sequences. Using this embedding space, we generated interpretable sentence representations, which are automatically grouped based on their semantic similarity. To provide a comprehensive understanding of the generated data, we employed visual summaries that offer insights at various levels of granularity. Through multiple case studies, we demonstrated the effectiveness of our approach in generating new insights into how models encode stereotypes and capture linguistic phenomena. More information under: <https://prompt-comparison.lingvis.io/>.

9 ACKNOWLEDGMENTS

This paper was supported by funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within projects BU 1806/10-2 “Questions Visualized” of the FOR2111 and the ETH AI Center.

REFERENCES

- [1] S. Alnegheimish, A. Guo, and Y. Sun. Using natural sentence prompts for understanding biases in language models. In *Proc. of the 2022 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2824–2830. Association for Computational Linguistics, Seattle, United States, July 2022. doi: 10.18653/v1/2022.naacl-main.203 2
- [2] S. Bandyopadhyay, J. Xu, N. Pawar, and D. Touretzky. Interactive visualizations of word embeddings for k-12 students. In *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 36, pp. 12713–12720, 2022. doi: 10.1609/aaai.v36i11.21548 2
- [3] A. Boggust, B. Carter, and A. Satyanarayan. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. In *27th Int. Conf. on Intelligent User Interfaces*, pp. 746–766, 2022. doi: 10.1145/3490099.3511122 2
- [4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. 1
- [5] M. Cheng, E. Durmus, and D. Jurafsky. Marked personas: Using natural language prompts to measure stereotypes in language models. In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1504–1532. Association for Computational Linguistics, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-long.84 2, 3, 4, 7
- [6] J. F. DeRose, J. Wang, and M. Berger. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1160–1170, 2021. doi: 10.1109/TVCG.2020.3028976 2
- [7] J. Eisenstein. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proc. of the 2022 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4326–4331. Association for Computational Linguistics, Seattle, United States, July 2022. doi: 10.18653/v1/2022.naacl-main.321 1
- [8] M. El-Assady, R. Kehlbeck, C. Collins, D. Keim, and O. Deussen. Semantic Concept Spaces: Guided Topic Model Refinement using Word-Embedding Projections. *IEEE Trans. on Visualization and Computer Graphics*, 26(1):1001–1011, 2019. doi: 10.1109/TVCG.2019.2934654 2
- [9] M. El-Assady, R. Kehlbeck, Y. Metz, U. Schlegel, R. Sevastjanova, F. Sperle, and T. Spinner. Semantic color mapping: A pipeline for assigning meaningful colors to text. In *VisGuides Workshop at IEEE VIS*, 2022. 3
- [10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Second Int. Conf. on Knowledge Discovery and Data Mining*, KDD’96, p. 226–231. AAAI Press, 1996. 3
- [11] A. Feder, K. A. Keith, E. Manzoor, R. Pryzant, D. Sridhar, Z. Wood-Doughty, J. Eisenstein, J. Grimmer, R. Reichart, M. E. Roberts, et al. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Trans. of the Association for Computational Linguistics*, 10:1138–1158, 2022. doi: 10.1162/tacl_a_00511 1
- [12] M. Gleicher. Considerations for visualizing comparison. *IEEE Trans. on Visualization and Computer Graphics*, 24:413–423, 2018. doi: 10.1109/TVCG.2017.2744199 1, 4, 5
- [13] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011. doi: 10.1177/1473871611416549 2
- [14] F. Hausdorff. *Grundzüge der Mengenlehre*, vol. 7. von Veit, 1914. doi: 10.1007/BF01999507 4
- [15] F. Heimerl, C. Kralj, T. Möller, and M. Gleicher. embcomp: Visual interactive comparison of vector embeddings. *IEEE Trans. on Visualization and Computer Graphics*, 28:2953–2969, 2019. 2
- [16] Z. Huang, D. Witschard, K. Kucher, and A. Kerren. VA + Embeddings STAR: A State-of-the-Art Report on the Use of Embeddings in Visual Analytics. *Computer Graphics Forum*, 2023. doi: 10.1111/cgf.14859 2
- [17] A. Joshi, S. Agrawal, P. Bhattacharyya, and M. J. Kerren. Expect the unexpected: Harnessing sentence completion for sarcasm detection. In *Computational Linguistics: 15th Int. Conf. of the Pacific Association for Computational Linguistics, PACLING 2017, Yangon, Myanmar*, pp. 275–287. Springer, 2018. doi: 10.1007/978-981-10-8438-6_22 2, 3
- [18] A.-L. Kalouli, R. Sevastjanova, C. Beck, and M. Romero. Negation, coordination, and quantifiers in contextualized language models. In *Proc. of the 29th Int. Conf. on Computational Linguistics*, pp. 3074–3085. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, Oct. 2022. 3, 7, 8
- [19] N. Kassner and H. Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7811–7818. Association for Computational Linguistics, Online, July 2020. doi: 10.18653/v1/2020.acl-main.698 3, 7
- [20] A. Lauscher, T. Lueken, and G. Glavaš. Sustainable modular debiasing of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4782–4797. Association for Computational Linguistics, Punta Cana, Dominican Republic, Nov. 2021. 2, 3, 4
- [21] S. Liu, Z. Li, T. Li, V. Srikumar, V. Pascucci, and P.-T. Bremer. NLIZE: A Perturbation-Driven Visual Interrogation Tool for Analyzing and Interpreting Natural Language Inference Models. *IEEE Trans. on Visualization and Computer Graphics*, 25(1):651–660, 2018. doi: 10.1109/TVCG.2018.2865230 2
- [22] L. Lucy and D. Bamman. Gender and representation bias in GPT-3 generated stories. In *Proc. of the Third Workshop on Narrative Understanding*, pp. 48–55. Association for Computational Linguistics, Virtual, June 2021. doi: 10.18653/v1/2021.nuse-1.5 1, 2, 3
- [23] L. McInnes, J. Healy, N. Saul, and L. Grossberger. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861 3
- [24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013. 3
- [25] S. Miksch and W. Aigner. A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics*, 38:286–290, 2014. 2
- [26] R. Navigli and S. P. Ponzetto. BabelNet: Building a very large multilingual semantic network. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 216–225. Association for Computational Linguistics, Uppsala, Sweden, July 2010. 3
- [27] D. Nozza, F. Bianchi, and D. Hovy. HONEST: Measuring hurtful sentence completion in language models. In *Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2398–2406. Association for Computational Linguistics, Online, June 2021. doi: 10.18653/v1/2021.naacl-main.191 2, 3
- [28] OpenAI. Gpt-4 technical report, 2023. 1, 2
- [29] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. 3
- [30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1
- [31] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China, Nov. 2019. doi: 10.18653/v1/D19-1410 3
- [32] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 6
- [33] B. Scarlini, T. Pasini, and R. Navigli. SenseBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *AAAI Conf. on Artificial Intelligence*, 2020. 3
- [34] R. Sevastjanova, E. Cakmak, S. Ravfogel, R. Cotterell, and M. El-Assady. Visual comparison of language model adaptation. *IEEE Trans. on Visualization and Computer Graphics*, 29(1):1178–1188, 2022. doi: 10.1109/TVCG.2022.3209458 2
- [35] R. Sevastjanova, A.-L. Kalouli, C. Beck, H. Hauptmann, and M. El-Assady. Lmfingerprints: Visual explanations of language model embedding spaces through layerwise contextualization scores. *Computer Graphics Forum*, 41(3):295–307, 2022. doi: 10.1111/cgf.14541 2
- [36] V. Sivaraman, Y. Wu, and A. Perer. Emblaze: Illuminating machine learning representations through interactive comparison of embedding spaces. In *27th Int. Conf. on Intelligent User Interfaces*, pp. 418–432, 2022. doi: 10.1145/3490099.3511137 2
- [37] H. Strobel, B. Hoover, A. Satyanarayan, and S. Gehrmann. LMDiff: A

- visual diff tool to compare language models. In *Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 96–105. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, Nov. 2021. doi: 10.18653/v1/2021.emnlp-demo.12 2
- [38] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008. 3
- [39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. of the 31st Int. Conf. on Neural Information Processing Systems, NIPS’17*, p. 6000–6010. Curran Associates Inc., Red Hook, NY, USA, 2017. 2
- [40] J. Vig. A Multiscale Visualization of Attention in the Transformer Model. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42. Association for Computational Linguistics, Florence, Italy, July 2019. doi: 10.18653/v1/P19-3007 2
- [41] L. Weidinger, J. Uesato, M. Rauh, C. Griffin, P.-S. Huang, J. Mellor, A. Glaese, M. Cheng, B. Balle, A. Kasirzadeh, C. Biles, S. Brown, Z. Kenton, W. Hawkins, T. Stepleton, A. Birhane, L. A. Hendricks, L. Rimell, W. Isaac, J. Haas, S. Legassick, G. Irving, and I. Gabriel. Taxonomy of risks posed by language models. In *2022 ACM Conf. on Fairness, Accountability, and Transparency, FAccT ’22*, p. 214–229. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3531146.3533088 1, 3