WEC-Explainer: A Descriptive Framework for Exploring Word Embedding Contextualization

Rita Sevastjanova[*]
University of Konstanz

Mennatallah El-Assady[†]
ETH AI Center

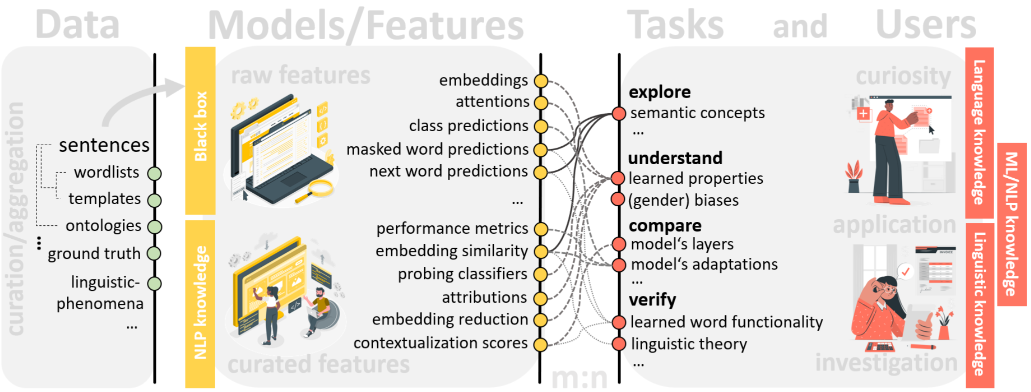

Figure 1: We present a descriptive framework for building applications for embedding contextualization that connects **data**, **models**, **tasks**, and **users**. In the interactive version of the framework (`https://embedding-framework.lingvis.io`), we give examples of various case studies from the related work and support researchers in specifying new analysis setups.

Abstract

Contextual word embeddings – high-dimensional vectors that encode the semantic similarity between words – are proven to be effective for diverse natural language processing applications. These embeddings originate in large language models and are updated throughout the model's architecture (i.e., the model's layers). Given their intricacy, the explanation of embedding characteristics and limitations – their contextualization – has emerged as a widely investigated research subject. To provide an overview of the existing explanation methods and motivate researchers to design new approaches, we present a descriptive framework that connects data, features, tasks, and users involved in the word embedding explanation process. We use the framework as theoretical groundwork and implement a data processing pipeline that we use to solve three different tasks related to word embedding contextualization. These tasks enable answering questions about the encoded context properties in the embedding vectors, captured semantic concepts and their similarity, and masked-prediction meaningfulness and their relation to embedding characteristics. We show that divergent research questions can be analyzed by combining different data curation methods with a similar set of features.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Information visualization

1 Introduction

Masked language models (LMs) such as BERT [8] or generative models like GPT-3 [5] are state-of-the-art for natural language processing (NLP) and understanding tasks. These models generate contextualized word embeddings – high-dimensional (HD) vectors that encode a word's context information, e.g., surface, syntax, and semantics [29]. Contextualized word embedding interpretability is relevant for various tasks, e.g., not only to understand the model's strength [29] and limitations (e.g., [10, 22]) but also encoded biases [23] and information relevant for making decisions on, e.g., which layer's embeddings to use for analysis [39] or how to adapt the model for the end user and task [17]. Furthermore, embeddings are commonly used as features for diverse visual analytics application scenarios (see a broad overview of applications in [13]). It is important to understand embedding characteristics to apply them to a specific use case effectively.

Continuously new computational and visual approaches are created to gain insights into embedding properties. The most common method is probing classifiers through which, e.g., Tenney et al. [37] have shown that BERT follows the typical NLP pipeline. The high-dimensional vectors [9, 38] or low-dimensional coordinates (2D) are commonly used to cluster words according to embedding similarity and analyze properties in their local neighborhoods [2, 11]. In addition to features, also the data used for explanations play a crucial role in insight generation. Most of the tasks require a variety of contexts, i.e., multiple contexts per token, to see the context's influence on embedding vectors [9]. Some tasks require more careful curation, e.g., specific contexts for bias detection [19] or syntactic/semantic structures for linguistic analysis [16]. Although most of the explainability tasks are tailored toward expert users (e.g., with machine learning (ML), NLP, or linguistic background), some tasks are relevant for laymen to gain a basic understanding of popularity-gaining language models. An overview of potential methods for data curation, features, potential tasks, and users and their interplay is currently missing.

In this paper, we present WEC-Explainer, a descriptive framework[1] that brings the different aspects involved in the explanation process, i.e., data, models/features, tasks, and users into relation. Furthermore, we use the descriptive framework as a theoretical groundwork to implement a data processing pipeline and use it for three case studies for visually explaining contextualized word em-

[*] e-mail: rita.sevastjanova@uni-konstanz.de
[†] e-mail: melassady@ethz.ch

[1] `https://embedding-framework.lingvis.io/`

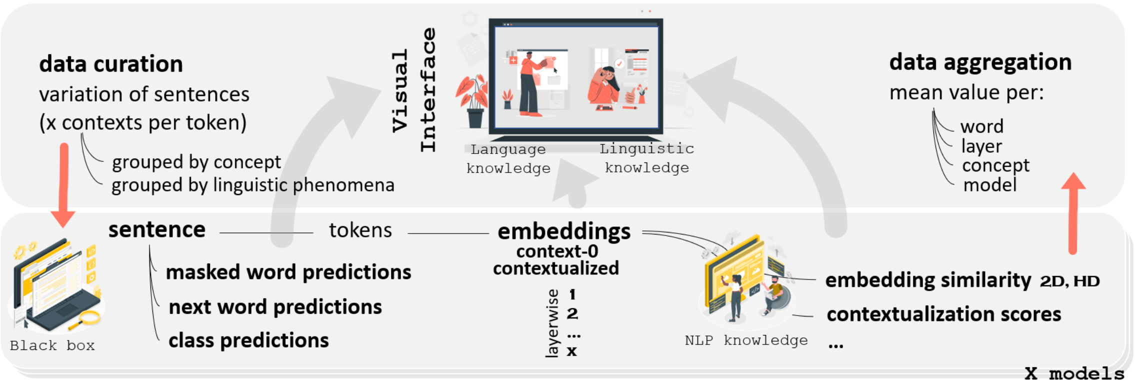

Figure 2: We use the descriptive framework as a groundwork to build a data processing pipeline that solves multiple embedding contextualization tasks. We extract both raw features coming directly from the model's architecture (e.g., contextualized embeddings) as well as representations curated by NLP researchers (e.g., contextualization scores).

beddings. These instances show how a variety of data input on a similar set of features can support different users and tasks.

2 Related Work

Various methods have been used to explore embedding contextualization (see a survey by Zini et al. [40] on LM explainability and a recent survey paper on utilizing embedding vectors in visual analytics applications [13]). It is shown that embeddings generated by deep-learning LMs are contextualized, i.e., words have different vector representations across different contexts [9]. Diverse computational and visual methods are used to explore embedding properties, i.e., characteristics encoded in vectors. One strand of research uses probing classifiers [14, 20] and adversarial testing [10, 22].

Another widely used method is embedding similarity. The similarity task is prevalent for visual exploration approaches. Early approaches analyze static embeddings, such as word2vec [25] and Glove [27], facilitating word analogies [21]. Reif et al. [28] and Wiedemann et al. [38] show that contextualized embeddings cluster with respect to word senses. Berger [1] explores correlations between embedding clusters in BERT. For instance, related work visualizes word embeddings in a scatterplot and applies metrics to measure local neighborhood changes [2, 11] or use animations and visual augmentations to show changes in the embedding spaces [34], just to name a few.

3 Descriptive Framework

In this paper, we focus on contextualized word embeddings and emphasize the potential of their combination with additional features and data curation methods for interpretability purposes. By following the design by Miksch and Aigner [26], we bring **data** and **models** into relation in a descriptive framework and highlight the role of **users** and their knowledge for diverse analysis **tasks**.

Data: A set of sentences is a typical input for word embedding exploration tasks. This set can be randomly generated or purposely curated to answer specific analysis questions. The questions related to contexts' impact on embedding change require a sufficient number of context variations per token [9]. For semantic similarity tasks, we can sample data representing concepts through predefined wordlists to answer questions related to gender-related stereotypes for analyzing biases [19]. We can use templates to prepare data systematically [18] to test embedding changes on minimal context alternation. By aggregating data based on ontologies, we can explore how the model encodes different categories [24]. We can create annotated ground-truth data containing labels for diverse tasks (e.g., sentiment) and analyze embedding specificities for different classes [35]. For linguistically motivated analysis, we can prepare samples that represent diverse linguistic phenomena, e.g., function words such as negation [16], or different aspects of linguistic theory [33], and analyze whether the model behaves as the theory expects. As shown in Fig. 1, methods can be combined to create the desired data format.

Models/Features: Features that can be used in combination with contextualized word embeddings can either be extracted from the **black box model** or produced when some **NLP knowledge** is available. Depending on the model's type (e.g., masked or generative) and analysis task, we can combine embeddings with classification outcomes, masked word-, or next-word predictions to analyze aspects like the main (semantic) concepts inherited in the model. We can also apply external knowledge and produce new features by, e.g., measuring embedding similarity [9] or applying further scoring techniques to explain linguistic properties encoded in embedding vectors [32]. We can train probing classifiers on embedding vectors [14, 20] to analyze encoded linguistic phenomena, as well as explore attributions [7] that get computed on embedding vectors to gain insight into word importance for a classification task. We can also compute static embeddings from contextualized ones for their better interpretation [3]. Theoretically, an endless number of features can be placed on this spectrum that requires some NLP knowledge for their curation, whereby some of them are more interpretable than others. One must be careful, though, because even in a sense interpretable features may require LM understanding for a correct interpretation (see, e.g., Sevastjanova and El-Assady [31]).

Tasks and Users: User groups for embedding contextualization tasks are similar to those of language model explainability (see, e.g., Brath et al. [4]). We describe them as users with general **language**-, **ML/NLP**-, and **linguistic knowledge**. There is not a clear cut between the groups, though. Depending on the users' knowledge and expertise, they have different interests and goals. Laymen typically explore the embedding space out of **curiosity**. There is no further objective to use the knowledge in a particular way. ML/NLP experts want to understand embedding properties and compare them among layers and models to **apply** the knowledge in future tasks, e.g., to select an embedding layer for an NLP task or improve the models' performance. Linguistic experts try to verify linguistic theories, e.g., whether LMs learn linguistic structures (e.g., syntax) or word functionality. Typically, they **investigate** a concrete hypothesis on a specific dataset. Theoretically, one can use the same features to test both linguistic theories and produce interpretable insights for a layman. Hence, this **m:n** relationship between features and tasks generates a huge space for exploration.

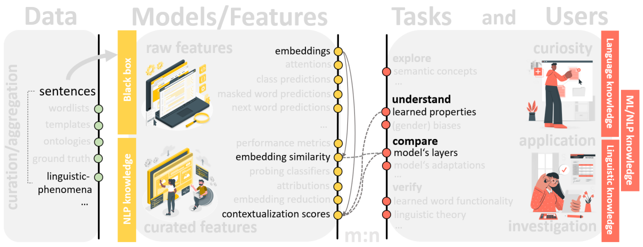

Figure 3: Data and features for analyzing context properties. To analyze linguistic phenomena, we can extract embedding vectors, compute their similarity as well as assess diverse contextualization scores and use them to compare the model's layers and gain insights into the learned properties in the different layers.

4 Data Processing Pipeline

We use the descriptive framework as a theoretical groundwork for building a data processing pipeline, as depicted in Fig. 2.

Data curation: A text corpus for the analysis is determined based on the use case at hand. The corpus contains a list of sentences; however, the specific use case controls whether sentences are grouped based on semantic concepts/wordlists (e.g., for bias detection tasks) or linguistic properties (e.g., for linguistic analysis).

Black box features: We first extract the raw features from the black box model for a given text corpus. If the use case includes model comparison tasks, the features are extracted from a set of models, respectively. The main feature for all use cases is contextualized word embeddings. We extract these embeddings layerwise, allowing us to investigate how information gets propagated through the model's architecture. We additionally extract context-0 (decontextualized [3]) embeddings by using the model's special tokens and the word itself as the input to the model (e.g., [CLS] word [SEP] in BERT). These can be used as a baseline, i.e., to learn what gets encoded in embeddings when no context is available. They are commonly used for the Word Embedding Association Test (WEAT) [6]. If required, the model computes masked word or next-word predictions. We combine these predictions with embedding vectors to describe the strongest word associations and potentially ignored context aspects (e.g., negations).

NLP features: After extracting the raw features, we build representations on top of embeddings in means of similarities (in 2D and HD) and scoring techniques. We compute several scores that describe the degree of contextualization using cosine similarity between layerwise embeddings of different reference tokens (e.g., tokens within the same context, nearest neighbors), initially introduced in [32]. We also compute scores that capture common characteristics of tokens and their nearest neighbors. These are measured on token string representations and context properties. For instance, we compute the inverse edit distance between a token and its nearest neighbor tokens to see whether the model captures a token's lexical representation. To see whether the model learns the token's word class, we measure the occurrence of nearest neighbors having the same POS tag. This information aggregated for a word group, or the whole corpus shows properties encoded in embeddings in different layers. For more details, see Sevastjanova et al. [32].

5 Application Scenarios

We present an instantiation of the descriptive framework through three application scenarios. In particular, we use the data processing pipeline in combination with visualizations to solve three different tasks related to word embedding contextualization, recently published as separate works (see [16, 30, 32]). We show that divergent research questions can be analyzed by combining different data curation methods with a similar set of features.

5.1 Encoded Context Properties

First, we analyze different linguistic phenomena (i.e., semantic, syntactic, and surface features) learned by LMs [32].

Task and users: The task is to gain an overview of properties encoded in embedding vectors in different model layers. This task is especially relevant for users with NLP expertise (see Fig. 3).

Data curation: Since the goal is to generate insights into various context properties, the only requirement for the text corpus is a sufficient number of sentences with different (semantic and syntactic) context variations.

Feature curation: We begin by extracting the layerwise contextualized word embeddings for all words in the corpus and use these to obtain words' nearest neighbors in the HD space using cosine similarity. Finally, we compute contextualization scores described in Sevastjanova et al. [32].

Data aggregation: We aggregate the scores in two levels to support two degrees of exploration. First, a corpus-level score aggregation highlights the layerwise changes, i.e., layers in which specific properties (e.g., lexical information) are more expressed in embedding vectors. Second, a token-level score aggregation computed as the average score of the same word used in multiple contexts enables

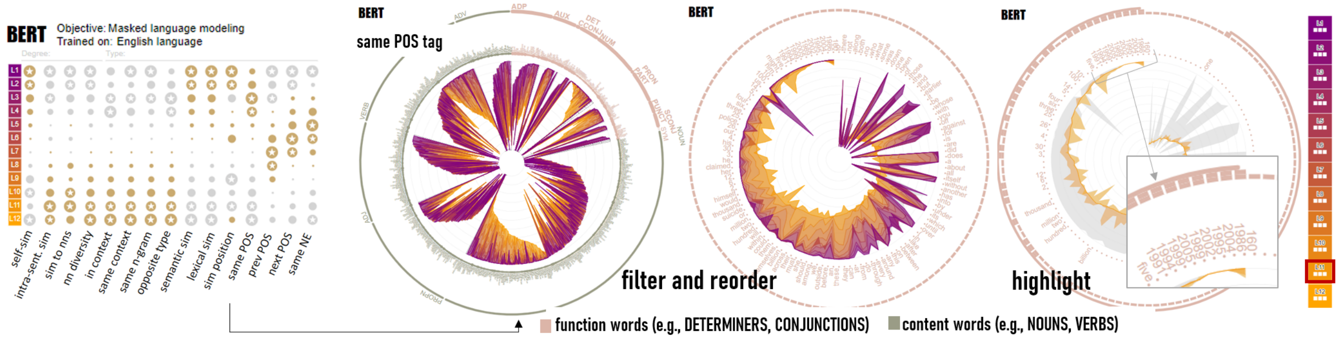

Figure 4: The embedding similarity in the form of contextualization scores shows context properties that get encoded in embedding vectors, which are relevant insights for linguistic experts. In BERT, surface features such as the word lexical similarity (i.e., edit distance) are highest in the early layers. The model learns syntactic context properties, such as the POS tag information in the middle layers. In the upper layers, word embeddings for words within the same sentence become more similar. Further explorations reveal that numbers stay similar to other numbers within all layers. More examples under `https://lmfingerprints.lingvis.io/`.

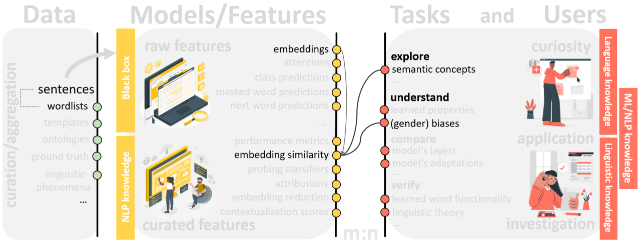

Figure 5: Data and features for concept similarity task. We can use wordlists describing different human-interpretable concepts, such as gender, and positive and negative human qualities, to define the corpus for analysis. For words and sentences in this corpus, we can extract embedding vectors, compute their inter and intra-concept similarity, and use it to explore concept similarity as well as gain insights into models' biases.

comparing words with different roles in the sentence (e.g., function words and content words).

Visualization: We use visualizations to gain insights into the encoded properties. A matrix shows contextualization specificities and highlights characteristics captured in different models' layers (see the golden circles with stars in the matrix in Fig. 4). The columns in the matrix represent scores; the rows depict layers of the particular model. Additionally, we use a radial layout to provide an overview of contextualization specificities for different token groups, e.g., proper nouns, function words, etc. We visually group tokens based on their POS tag to ease the word comparison task.

A single layer is displayed as a line that connects the score values for all tokens in the corpus (i.e., 12 lines for a model with 12 layers). We color the lines according to a sequential color scale (i.e., from purple representing layer 1 to orange representing layer 12). To facilitate readability, we additionally color the area between two succeeding layers and decrease their opacity to see overlapping layers. The design is similar to a braided graph visualization, whereby each braid has transparency, and thus, the overlapping layers are visible. We support several interaction techniques to ease the interpretation of the shown patterns. First, to analyze token groups in more detail, the users can select POS tag(s), and the corresponding tokens are filtered. Second, the users can sort tokens according to different properties: alphabet, maximum score value among all layers, and score value in a specific layer.

Findings: Our approach allows reproducing insights obtained using probing classifiers concerning encoded properties in embedding vectors [29]. Among others, the scoring functions, similar to probing classifiers, show that the syntactic information is most prominent in the middle layers, and surface features are captured best in the early layers of BERT. In addition to these known insights, we observe that some named entity categories (e.g., geographical locations) and numbers have relatively poor contextualization. The self-similarity and the POS tag similarity score of year numbers stay high throughout all layers (see Fig. 4). It means that embeddings of these tokens do not change within the model's architecture independently in which context they are used.

5.2 Semantic Concept Similarity

In addition to the analysis of encoded properties, our framework enables the comparison of multiple model instances to determine which one creates embeddings that fit user expectations regarding word semantic similarity [30].

Task and users: The task is to understand how well semantic concepts are separated in the (HD or 2D) embedding space and it is relevant for users with general language knowledge (see Fig. 5).

Data curation: We use pre-defined wordlists to sample sentences that contain keywords representing different semantic concepts (e.g., human characteristics, person names). To ease the analysis of bias-related tasks, we represent one concept by two subconcepts, each having a specific polarity (e.g., positive and negative human characteristics, female and male person names, respectively).

Feature curation: To understand whether the model learns to separate the semantic concepts in the embedding space, we first represent words by their context-0 and layerwise contextualized word embeddings. The contextualized embeddings get aggregated for each unique word (e.g., one average embedding from all occurrences of a word per layer). Next, we obtain embedding similarity in both HD and 2D space. The latter is done using a dimensionality reduction method (e.g., PCA [15]).

Visualization: Finally, we use visualization techniques to show the similarity between concepts in HD and 2D space. Two example visualizations are shown in Fig. 6. In both, we use color encoding and area (contour lines) to group words belonging to the same sub-concept. The *Concept Embedding Similarity* visualization (see the left side of Fig. 6) displays the cosine similarity between two concepts. Here, one concept is used as an anchor for explanation purposes, whereby its two sub-concepts (here: female- and male

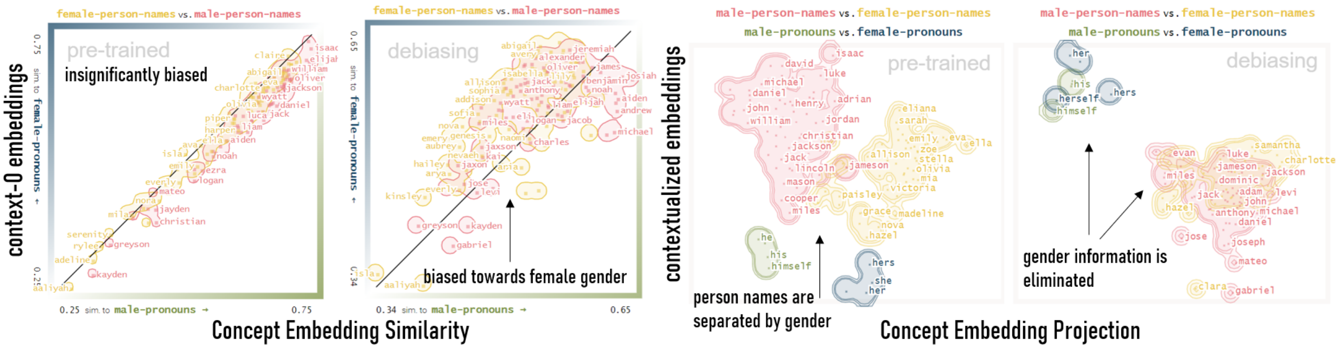

Figure 6: The embedding similarity between semantic concepts such as person names and pronouns shows whether models encode gender information. This is particularly relevant for bias detection tasks. Here, the context-0 embeddings in the pre-trained BERT and the debiasing adapter by Lauscher et al. [19] do not encode the gender information in person name vectors. For instance, context-0 embeddings of person names (both male and female) are slightly more similar to male pronouns. However, the contextualized embeddings in pre-trained BERT encode person names and pronouns, both separated by gender; however, the debiasing adapter removes the gender information. There, embeddings are separated for the different POS tags. More examples under `https://adapters.demo.lingvis.io/`.

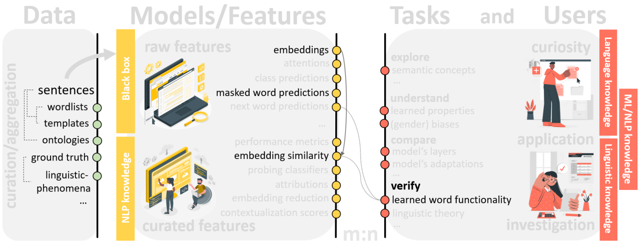

Figure 7: Data and features for the masked prediction task. We prepare the ground truth data on diverse linguistic phenomena (i.e., function word classes) by using several wordlists and ontology-based approaches.

pronouns) define the two axes in the visualization. The similarity to these sub-concepts defines the second concept's word positions in the visualization. In the *Concept Embedding Projection* (see the right side of Fig. 6), on the contrary, the 2D positions are obtained using a dimensionality reduction technique. The two visualizations can be used standalone or combined to detect artifacts produced by a single method.

Findings: Using our framework, we reproduce findings by Lauscher et al. [19] who have introduced an adapter model [12] for gender bias elimination. In particular, the authors show that the model removes gender biases according to diverse evaluation methods except for the WEAT. WEAT results show that the model does not reduce the bias but instead inverts it. The same insights provide our *Concept Embedding Similarity* visualization on context-0 embeddings (Fig. 6 left side) where person names (both male and female) are more similar to the female gender than in the pre-trained BERT. This bias inversion is visible for context-0 embeddings, though. In contextualized embeddings, there is no separation between the person-name and pronoun concepts (Fig. 6 right side). Exploring further models, we observe that the gender information is typically obtained from the word's context, and it generally is not encoded in the word (e.g., person name) itself. Such a visual exploration also helps to understand the properties and limitations of context-0 embeddings used for semantic analysis tasks since these vectors often do not match the properties of their contextualized versions.

5.3 Masked Prediction Quality

Extending the data and feature curation steps, we can gain insights into linguistic phenomena (e.g., function words) that have a smaller impact on the models' performance [16].

Task and users: The task is to gain insight into masked prediction meaningfulness when it comes to contexts involving various function word classes, i.e., the task is to detect how an LM captures a word's functionality. This task is especially relevant for users with linguistic expertise (Fig. 7).

Data curation: For this approach, the data curation requires more linguistic input than the previous use cases. In particular, since the goal is to analyze the quality of masked predictions, the data needs to carry ground truth – a descriptor that specifies which predictions are more or less likely to occur in the real world. We thus combine several approaches. We use templates, pre-defined wordlists, and ontologies (e.g., ConceptNet [36]) to curate sentences for the analysis. We prepare sentence pairs that vary only by a function word from a specific type (e.g., quantifiers *all* and *no* in Fig. 8). The last word of each sentence gets masked. Using linguistic knowledge, some of the sentences get annotated with forbidden words according to the theoretical linguistic literature. For others, our goal is to measure prediction overlap between the sentence pairs that vary by a single function word (e.g., *All insects have [MASK].* and *No insect has [MASK].*).

Feature curation: We extract the ten most likely predictions for the masked words and obtain their embedding similarity to all other words in the particular sentence. We aim to explore similarity patterns concerning meaningful and meaningless predictions. Using this approach, we aim to gain the first insights into the potential of using similarity scores for prediction quality analysis.

Visualization: Masked predictions are displayed as rows in a matrix visualization (i.e., ten rows for ten predictions as shown in Fig. 8). We display the prediction's probability on the left through a horizontal bar. Next to the probability are placed tokens visualized as colored rectangles. The rectangle's color represents its cosine similarity to the predicted word of the particular sentence. The darker the color, the higher the similarity. On the right is displayed the predicted word. To support the analysis of prediction overlaps and prediction of forbidden words, we color the words that overlap or are forbidden, respectively, in red color.

Findings: Using this approach, we can not only reproduce findings by Kassner and Schütze [18] regarding the high number of prediction overlaps that suggest that models *potentially ignore* some of the function words when predicting masked tokens, but we can also see some evidence in the correlation between prediction meaningfulness and embedding similarity. In particular, as shown in Fig. 8, for a sentence pair *All insects have [MASK].* and *No insect has [MASK].* the three overlapping (and meaningless) predictions for the second

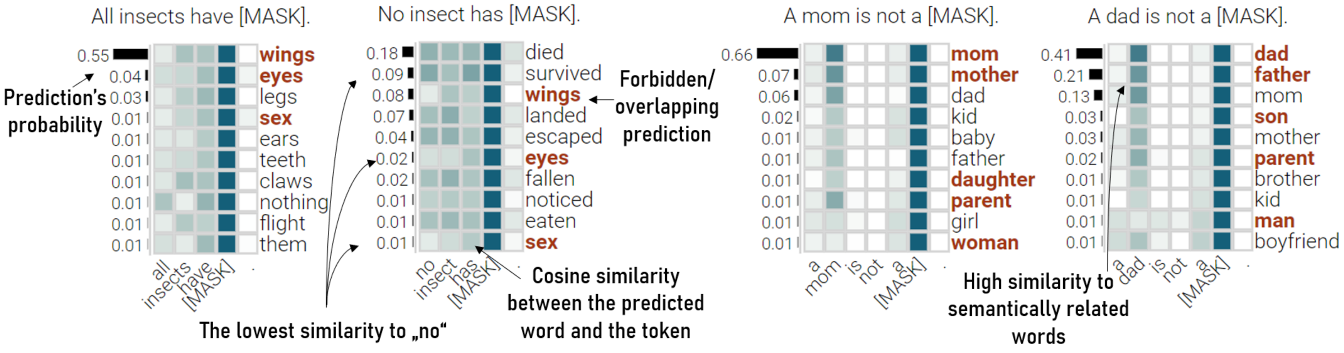

Figure 8: Embedding similarity can indicate situations where parts of the context (e.g., function words) impact the masked predictions less than semantically rich content words. In the examples *No insect has [MASK].* using the BERT model, the lowest similarity to the function word *no* is in semantically wrong predictions. The example *A mom is not a [MASK].* shows that predictions are influenced by semantically rich words without considering the functional role of function words. More examples under `https://function-words.lingvis.io/`.

.

sentence are those where the predicted work has the lowest similarity to the word *no*. This might suggest that the negation in these cases is ignored. This is an interesting future research direction to learn how embedding similarity can be used as an indicator of masked prediction quality.

6 Discussion

Although many approaches for embedding explanations have been developed in both computational linguistics and visualization communities, there are many open research questions related to embedding properties, capabilities, and limitations. With this work, we aim to motivate researchers to keep working on further explanation approaches for embedding contextualization tasks. Our descriptive framework should help to structure new research questions by giving an overview of potential feature- and data combinations.

The exploration of embedding properties, and the design of new approaches, is still an open research challenge. Not only do some current explanations generate contradicting results [29], but many questions related to embedding contextualization are still not fully answered. For instance, it is not fully clear how the model captures the context, which properties are more prominent in embeddings than others, whether all the learned properties are kept till the last model's layer, or whether some information gets overwritten due to the characteristics of the model's architecture (i.e., transformers). One also could extend the embedding contextualization analysis to single neurons. This is particularly challenging since their interpretation is more restricted than, e.g., semantic concept analysis presented in this work. Since more and more capable models are produced every day (e.g., ChatGPT[2]), the question of how to assess the generated output quality gains an increased relevance also outside the computational linguistics and visualization research fields.

7 Conclusion

We present a descriptive framework for word embedding explanation tasks that relates data and features to tasks and users. We use the framework as theoretical groundwork for a data processing pipeline that we use to solve three different tasks related to word embedding contextualization. Our work highlights the huge exploration space for future research and should help researchers in designing their embedding explanation interfaces.

8 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] M. Berger. Visually Analyzing Contextualized Embeddings. In *IEEE Visualization Conf. (VIS)*, pp. 276–280. IEEE Computer Society, Los Alamitos, CA, USA, oct 2020. doi: 10.1109/VIS47514.2020.00062

[2] 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.

[3] R. Bommasani, K. Davis, and C. Cardie. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4758–4781. Association for Computational Linguistics, Online, July 2020. doi: 10.18653/v1/2020.acl-main.431

[4] R. Brath, D. Keim, J. Knittel, S. Pan, P. Sommerauer, and H. Strobelt. The Role of Interactive Visualization in Explaining (Large) NLP Models: from Data to Inference. *arXiv*, 2023. doi: 10.48550/ARXIV.2301.04528

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[6] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[7] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen. A survey of the state of explainable AI for natural language processing. In *Proc. of the 1st Conf. of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th Int. Joint Conf. on Natural Language Processing*, pp. 447–459. Association for Computational Linguistics, Suzhou, China, Dec. 2020.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. ACL, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-1423

[9] K. Ethayarajh. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proc. of the Conf. on Empirical Methods in Natural Language Proc. and the Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65. ACL, Hong Kong, China, Nov. 2019. doi: 10.18653/v1/D19-1006

[10] M. Glockner, V. Shwartz, and Y. Goldberg. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 650–655. Association for Computational Linguistics, 2018.

[11] F. Heimerl, C. Kralj, T. Moller, and M. Gleicher. embcomp: Visual interactive comparison of vector embeddings. *IEEE Trans. on Visualization and Computer Graphics*, 2020.

[12] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *Int. Conf. on Machine Learning*, pp. 2790–2799. PMLR, 2019.

[13] 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

[14] G. Jawahar, B. Sagot, and D. Seddah. What Does BERT Learn about the Structure of Language? In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657. Association for Computational Linguistics, Florence, Italy, July 2019. doi: 10.18653/v1/P19-1356

[15] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[16] 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. Int. Committee on Computational Linguistics, Gyeongju, Republic of Korea, Oct. 2022.

[17] M. Kang, J. Baek, and S. J. Hwang. KALA: knowledge-augmented language model adaptation. In *Proc. of the 2022 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5144–5167. Association for Computational Linguistics, Seattle, United States, July 2022. doi: 10.18653/v1/2022.naacl-main.379

[18] 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

[2] `https://openai.com/blog/chatgpt`

[19] 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.

[20] Y. Lin, Y. C. Tan, and R. Frank. Open Sesame: Getting inside BERT's Linguistic Knowledge. In *Proc. of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 241–253. Association for Computational Linguistics, Florence, Italy, Aug. 2019. doi: 10.18653/v1/W19-4825

[21] 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 Trans. on Visualization and Computer Graphics*, 24(1):553–562, 2017. doi: 10.1109/TVCG.2017.2745141

[22] R. Marvin and T. Linzen. Targeted Syntactic Evaluation of Language Models. In *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, pp. 1192–1202. Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018. doi: 10.18653/v1/D18-1151

[23] N. Meade, E. Poole-Dayan, and S. Reddy. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, May 2022.

[24] J. Michael, J. A. Botha, and I. Tenney. Asking without telling: Exploring latent ontologies in contextual representations. In *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6792–6812. Association for Computational Linguistics, Online, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.552

[25] 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.

[26] 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.

[27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[28] E. Reif, A. Yuan, M. Wattenberg, F. B. Viegas, A. Coenen, A. Pearce, and B. Kim. Visualizing and Measuring the Geometry of BERT. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, eds., *Advances in Neural Information Processing Systems 32*, pp. 8594–8603. Curran Associates, Inc., 2019.

[29] A. Rogers, O. Kovaleva, and A. Rumshisky. A Primer in BERTology: What We Know About How BERT Works. *Trans. of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349

[30] R. Sevastjanova, E. Cakmak, S. Ravfogel, R. Cotterell, and M. El-Assady. Visual Comparison of Language Model Adaptation. In *IEEE Trans. on Visualization and Computer Graphics*, 2022 (accepted).

[31] R. Sevastjanova and M. El-Assady. Beware the Rationalization Trap! When Language Model Explainability Diverges from our Mental Models of Language. *Conf.: Communication in Human-AI Interaction Workshop at IJCAI-ECAI'22*, abs/2207.06897, 2022.

[32] 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.

[33] R. Sevastjanova, A.-L. Kalouli, C. Beck, H. Schäfer, and M. El-Assady. Explaining contextualization in language models using visual analytics. In *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. on Natural Language Processing (Volume 1: Long Papers)*, pp. 464–476. Association for Computational Linguistics, Online, Aug. 2021. doi: 10.18653/v1/2021.acl-long.39

[34] 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.

[35] F. D. Souza and J. a. B. d. O. e. S. Filho. Bert for sentiment analysis: Pre-trained and fine-tuned alternatives. In *Comp. Proc. of the Portuguese Language: 15th Int. Conf., PROPOR 2022*, p. 209–218. Springer-Verlag, Berlin, Heidelberg, 2022. doi: 10.1007/978-3-030-98305-5_20

[36] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conf. on Artificial Intelligence*, 2017.

[37] I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601. ACL, Florence, Italy, July 2019. doi: 10.18653/v1/P19-1452

[38] G. Wiedemann, S. Remus, A. Chawla, and C. Biemann. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. In *Proc. of KONVENS 2019*. Erlangen, Germany, 2019.

[39] H. Xu, B. Van Durme, and K. Murray. BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation. In *Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing*, pp. 6663–6675. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, Nov. 2021.

[40] J. E. Zini and M. Awad. On the explainability of natural language processing deep models. *ACM Comput. Surv.*, 55(5), dec 2022. doi: 10.1145/3529755