# CogniVal in Action: An Interface for Customizable Cognitive Word Embedding Evaluation

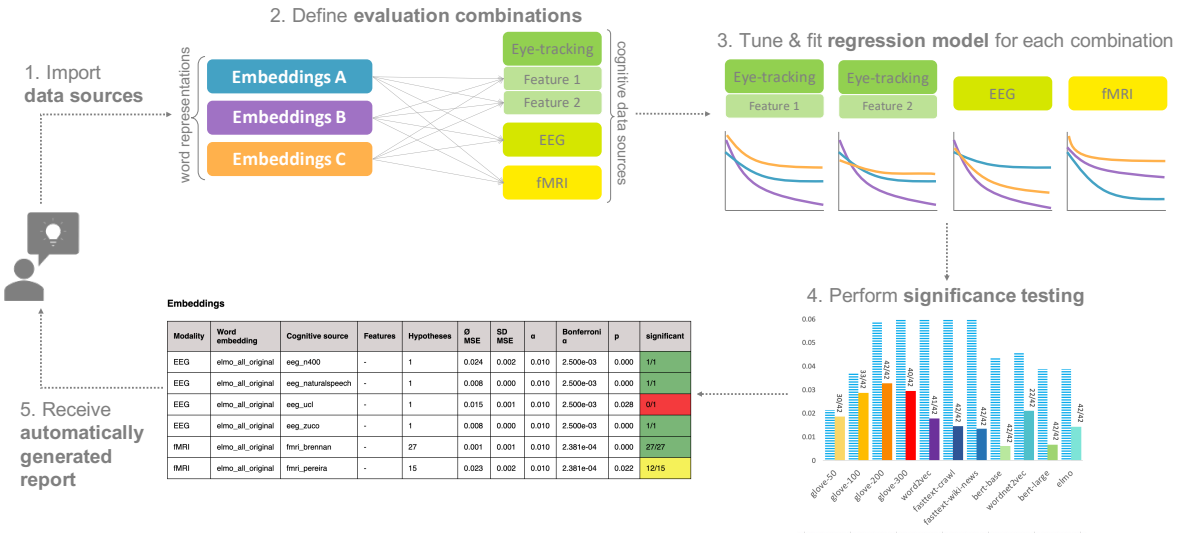
**Anonymous COLING submission**

## Abstract

We demonstrate the functionalities of the new user interface for CogniVal. CogniVal is a framework for the cognitive evaluation of English word embeddings, which evaluates the quality of the embeddings based on their performance to predict human lexical representations from cognitive language processing signals from various sources. In this paper, we present an easy-to-use command line interface for CogniVal with multiple improvements over the original work, including the possibility to evaluate custom embeddings against custom cognitive data sources.

## 1 Introduction & Background

The system presented in this work is based on the CogniVal framework presented by Hollenstein et al. (2019). We present the first encompassing framework for cognitive word embedding evaluation. We improve and extend the original features of CogniVal and provide a simple command line interface for scalable and customized experiments. CogniVal is openly available at `https://github.com/DS3Lab/cognival-cli` and can be easily installed with `pip`.




| Modality | Word embedding | Cognitive source | Features | Hypotheses | Ø MSE | SD MSE | α | Bonferroni α | p | significant |
|---|---|---|---|---|---|---|---|---|---|---|
| EEG | elmo_all_original | eeg_n400 | - | 1 | 0.024 | 0.002 | 0.010 | 2.500e-03 | 0.000 | 1/1 |
| EEG | elmo_all_original | eeg_naturalspeech | - | 1 | 0.008 | 0.000 | 0.010 | 2.500e-03 | 0.000 | 1/1 |
| EEG | elmo_all_original | eeg_ucl | - | 1 | 0.015 | 0.001 | 0.010 | 2.500e-03 | 0.028 | 0/1 |
| EEG | elmo_all_original | eeg_zuco | - | 1 | 0.008 | 0.000 | 0.010 | 2.500e-03 | 0.000 | 1/1 |
| fMRI | elmo_all_original | fmri_brennan | - | 27 | 0.001 | 0.001 | 0.010 | 2.381e-04 | 0.000 | 27/27 |
| fMRI | elmo_all_original | fmri_pereira | - | 15 | 0.023 | 0.002 | 0.010 | 2.381e-04 | 0.022 | 12/15 |

Figure 1: User interaction with the CogniVal interface.

Language models and word representations are the corner stones of state-of-the-art NLP models. Evaluating and comparing the quality of different word representations is a well-known, largely open challenge. While word representations and language models have proven very useful for NLP applications, their interpretability is inherently challenging. Interpretability is key for many NLP applications to be able to understand the algorithms' decisions. For a truly intrinsic evaluation of word embeddings more research about the cognitive plausibility of current language models is required (Rogers et al., 2018).





*Proceedings of the 27th International Conference on Computational Linguistics*, pages 34–40
Barcelona, Spain (Online), December 12, 2020.

Moreover, one of the challenges for computational linguistics is to build cognitively plausible models of language processing, i.e., models that integrate multiple aspects of human language processing at the syntactic and semantic level (Keller, 2010).

Currently, word embeddings are evaluated with extrinsic or intrinsic methods. Extrinsic evaluation is the process of assessing the quality of the embeddings based on their performance on downstream NLP tasks, e.g., sentiment analysis. However, embeddings can be trained and fine-tuned for specific tasks, but this does not mean that they accurately reflect the meaning of words. One the other hand, intrinsic evaluation methods, such as word similarity and word analogy tasks, merely test single linguistic aspects. These tasks are based on conscious human judgements, which can be biased by subjective factors (Nissim et al., 2019). It has been noted that current intrinsic evaluation methods do not capture the cognitive plausibility of the word embeddings and language models (Manning et al., 2020). Both intrinsic and extrinsic evaluation types often lack statistical significance testing and do not provide a global quality score. CogniVal addresses these issues and proposes an evaluation method based on cognitive lexical semantics.

Cognitive lexical semantics proposes that words are defined by how they are organized in the brain (Miller and Fellbaum, 1992). Recordings of brain activity play a central role in furthering our understanding of how human language works. Huth et al. (2016) showed in a neuroscientific study how words are represented in semantic maps across the brain. Moreover, language representations tuned on brain activity show improved performance on NLP tasks (Schwartz et al., 2019), and word representations trained on brain activity are more generalizable to unseen words (Fyshe et al., 2014). Hence, it seems natural to evaluate language models against human language processing data, as we have proposed with CogniVal (Hollenstein et al., 2019).

To accurately encode the semantics of words, we believe that embeddings, and full language models, should reflect this mental lexical representation. This allows us to evaluate word embeddings by quantifying their cognitive plausibility. There was a need for an extensive approach showing the utility of human cognitive data for language model evaluation and its correlation in predicting downstream task performance (Gladkova and Drozd, 2016). Evaluating word embeddings with cognitive language processing data has been proposed previously. For instance, Abnar et al. (2018) and Rodrigues et al. (2018) evaluated different embeddings by predicting the neuronal activity of nouns. Søgaard (2016) showed preliminary results in evaluating embeddings against continuous text stimuli in eye-tracking and functional magnetic resonance imaging (fMRI) data. Moreover, Beinborn et al. (2019) recently presented an extensive set of language–brain encoding experiments. Electroencephalography (EEG) data has been used for similar purposes. Schwartz and Mitchell (2019) and Ettinger et al. (2016) show that components of event-related potentials can successfully be predicted with neural network models and word embeddings. However, these approaches mostly focus on one modality of brain activity data from small individual cognitive datasets. The presence of only few and small data sources has been one reason why this type of evaluation has not been too popular until now (Bakarov, 2018).

Hence, for CogniVal we collected a wide range of cognitive data sources ranging from eye-tracking to EEG and fMRI to ensure coverage of a large vocabulary and of different features of the cognitive processes during language comprehension. The CogniVal command line interface (CLI) is the first tool to unify a diverse range of cognitive data sources of multiple recording modalities of cognitive processing signals and to provide a generic user interface. In this paper, we provide a user interface for CogniVal, which evaluates English word embeddings against the lexical representations of words in the human brain, recorded when passively understanding language. The CogniVal command line interface (CLI) makes large-scale cognitive word embedding evaluation accessible to NLP practitioners. It offers pre-processed cognitive data sources, readily provided for evaluation in a user-friendly interaction. It supports and complements other intrinsic and extrinsic evaluation methods for word embeddings. The CogniVal CLI is a unified framework, allowing the evaluation of a large set of existing pre-trained embeddings but also of custom word representations and language models on a large range of cognitive sources.

| Interaction step | Example command(s) |
|---|---|
| 1. Import data sources | `$ import cognitive-sources source=yourCustomSource`<br>`$ import embeddings youCustomEmbeddings.zip`<br>`$ import embeddings glove.6B.50`<br>`$ import random-baselines glove.6B.50 num-baselines=10` |
| 2. Define evaluation combinations | `$ config experiment cognitive-sources=[eeg_zuco] embeddings=[glove.6B.50]`<br>`$ config experiment cognitive-sources=[eye-tracking_geco] embeddings=[fasttext]` |
| 3. Fit regression models | `$ run embeddings=[glove.6B.50,glove.6B.100] cognitive-sources=[eye-tracking_geco] cognitive-features=[WORD_FIXATION_COUNT]` |
| 4. Perform significance testing | `$ significance run_id=0 modalities=[eye-tracking, eeg, fmri] alpha=0.01 test=Wilcoxon` |
| 5. Generate report | `$ report open-html=True` |

Table 1: Main steps and example commands for using the CogniVal CLI for cognitive word embedding evaluation.

## 2 System Overview

The Cognival CLI is implemented in Python (version 3.7.4) and provides an interactive shell using `python-nubia`[1]. For the purpose of cognitive embedding evaluation, Hollenstein et al. (2019) collected and prepared 15 cognitive data sources and evaluated 6 pre-trained embedding types, including GloVe, word2vec, WordNet2Vec, FastText, ELMo and BERT. The command line interface provides these preprocessed data types. For details about the format of the cognitive data sources please refer to Hollenstein et al. (2019).

The evaluation process is automatized in the CogniVal CLI and works as as depicted in Figure 1. First, the user defines the general evaluation configuration, including path specifications and training parameters (command: `config`). If required, the user can then import custom word representations as well as custom cognitive data sources, using the `import` function. Second, the user specifies the embedding/cognitive-data combinations to be evaluated, as well as the hyper-parameter ranges for the neural regression models. Moreover, if requested, CogniVal generates random vectors of the same dimension of the embeddings to be evaluated. The embeddings can also be evaluated against this random baseline. As an improvement from Hollenstein et al. (2019), the CogniVal CLI automatically generates 10 sets of different random embeddings and averages over the results for a fairer comparison to a more robust baseline. The tuning (implemented through a grid search) and training of all models (*n* embeddings x *m* cognitive data sources) is fully automatized within the command `run`.

Thereafter, the user can either use the saved results as they are (i.e., mean squared errors for each word in the vocabulary), or they can run the significance testing (command: `significance`), which consists of a Wilcoxon signed-rank test for each hypothesis (i.e., for each embedding/cognitive-data evaluation combination), applying the Bonferroni correction for the multiple hypotheses problem, as described by Dror et al. (2018). Finally, the automatized generation of the report also includes significance testing by default and compares the results to the baseline of random embeddings before aggregating them. The dynamic HTML or PDF reports include all detailed results for the individual combinations, as well as aggregated over the modalities (see Figure 2). Table 1 presents the most important commands provided in the CogniVal CLI. Additionally, please refer to the GitHub repository for a full tutorial[2].

## 3 Use Cases

The target audience for the CogniVal CLI are NLP and machine learning practitioners and researchers developing word embeddings and in need of an evaluation benchmark. In this CogniVal demonstration paper, we describe the following two possible use case scenarios.

[1] `https://github.com/facebookincubator/python-nubia`

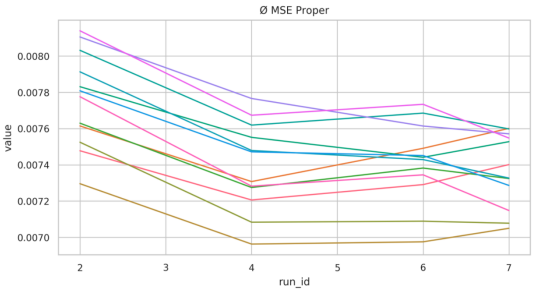
[2] `https://github.com/DS3Lab/cognival-cli/blob/master/cognival_tutorial.pdf`

**Results (Aggregated per Embedding and Modality)**

**Eye-Tracking**

| Word embedding | Ø MSE Baseline | Ø MSE Proper | Significance |
|---|---|---|---|
| bert1 | 0.13905 | 0.00740 | 42/42 |
| bert2 | 0.13974 | 0.00760 | 42/42 |
| bert3 | 0.14057 | 0.00705 | 42/42 |
| bert4 | 0.14013 | 0.00708 | 42/42 |
| bert5 | 0.13949 | 0.00732 | 42/42 |
| bert6 | 0.14030 | 0.00753 | 42/42 |
| bert7 | 0.13835 | 0.00760 | 42/42 |
| bert8 | 0.13772 | 0.00733 | 42/42 |
| bert9 | 0.14116 | 0.00729 | 42/42 |
| bert10 | 0.13918 | 0.00757 | 42/42 |
| bert11 | 0.13794 | 0.00755 | 42/42 |
| bert12 | 0.13936 | 0.00715 | 42/42 |

(a) Automatically generated result table.




(b) Plot over time: When adding more eye-tracking features in each run, the aggregated results become more precise. Each colored line represents a different BERT layer.

Figure 2: Snippets from an automatically generated result report in the CogniVal CLI.

### Scenario 1: Custom Word Embeddings & Cognitive Data Sources

One of the most relevant features of the CogniVal CLI is the possibility to upload custom word representations from any language model. Any type of word embedding can be imported into the system as text or binary files, and can then be evaluated against the available cognitive data sources and compared to the other embeddings included in CogniVal by default. Moreover, the automated versioning and reporting supports the development process of new embeddings by readily generating plots over the course of time to show whether the performance of the embeddings in development is improving or deteriorating across multiple runs.

In addition, custom cognitive data sources can also be imported into the CogniVal interface. This feature allows the user to add more cognitive language processing data as more of these datasets become available (Alday, 2019). Through these features CogniVal becomes a generic framework for cognitive word embedding evaluation and drastically increases the number of possible applications in any language.

### Scenario 2: Complementary Benchmark & Evaluation Over Time

A second use case scenario for the CogniVal CLI is to use the cognitive word embedding evaluation as a complementary evaluation and a benchmark for embedding selection. If the user wants to compare the results achieved by their embeddings on CogniVal to other intrinsic or extrinsic results achieved with the same embeddings, the CLI allows to upload external results into the result directory and run the significance testing and aggregation report on all available results. Hence, CogniVal can easily be extended to include external results and can be leveraged as a tool for embedding selection. Furthermore, CogniVal can be used during the development of language models. If one is comparing a certain checkpoint of a language model to extrinsic results on a downstream task, the cognitive evaluation can help to ensure that the word representations are not overfitting on the downstream task and that they still maintain cognitive plausibility. To this end, the automatic report generated in the CogniVal CLI also includes plots to show changes over various runs, which can be very useful during the development of new language models or fine-tuning of existing pre-trained language models (see Figure 2b).

## 4 Example Application: Comparison of BERT Layers

As an exemplary use case scenario for the CogniVal CLI, we analyze the performance of different layers of BERT pre-trained contextual word representations on all available cognitive data sources of eye-

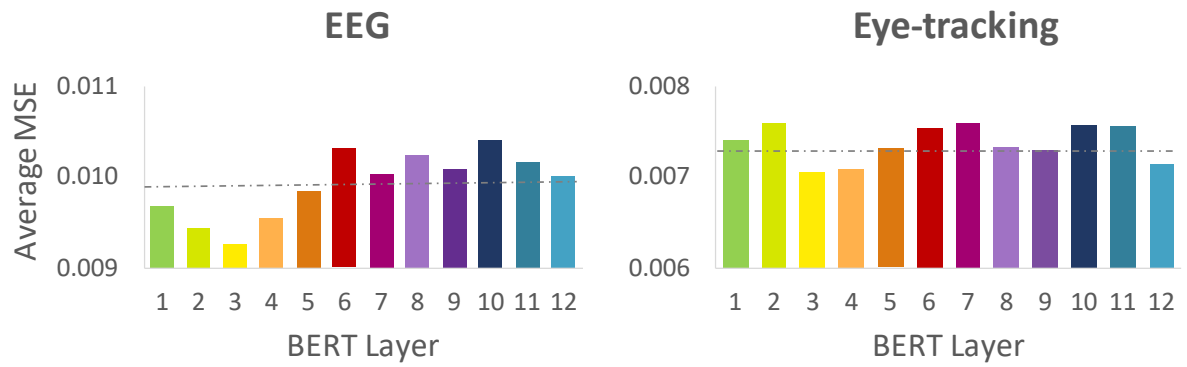



Figure 3: CogniVal results on evaluating the 12 layers of a BERT model against all available EEG and eye-tracking data sources.

tracking and EEG. Transformer-based language models such as BERT are widely used in state-of-the-art NLP, but their inner workings are still largely unknown (Rogers et al., 2020). We extract word-level BERT embeddings (Devlin et al., 2019) for all words where there is cognitive data available. Using the bert-as-service package[3] we extract the hidden states of all 12 layers of the BERT base uncased model with 768 dimensions. Subsequently, using the new CogniVal functionality to load custom embeddings, we import the BERT states of each layer easily into the CogniVal interface. We set the configuration to run all experiments against the 10-fold random baseline, with the following parameters for training: 3 fold cross-validation, a hidden layer of 200 dimensions, 20% validation split, batch size of 128, and ReLu activation functions.

The results of this example application are presented in Figure 3. In addition, Figure 2a shows the numerical summary of the results of BERT embeddings predicting eye-tracking features as it is presented in the automatically generated report, including the results of the random baselines and the results of the significance testing. While all hypotheses tested on the BERT layers proved to be statistically significant against the random baseline (4 EEG hypotheses – one for each dataset, and 42 eye-tracking hypotheses – one for each feature), there are visible differences in performance between the layers. Surprisingly, for both EEG and eye-tracking, layer 3 performs best. The results also show, how the last layer performs very closely to the average of all layers (dashed line). This finding reflects the original performance of the layers of the BERT base model on downstream NLP tasks (Devlin et al., 2019). It is also in line with Toneva and Wehbe (2019), who find that the lower layers perform best at predicting neural activation for short context ranges. Lin et al. (2019) show that the lower layers have the most linear word order information, which is likewise reflected in our results. This application scenario show how Cognival can be used to explore and support findings concerning the interpretability of language models.

## 5 Conclusion & Future Work

In this demonstration paper, we presented the new command line interface for CogniVal. The CogniVal CLI builds upon the work by Hollenstein et al. (2019) and extends it with various new features, especially the ability to evaluate custom embeddings against custom cognitive data sources. We described the functionalities of the tool as well as various use cases and an application scenario. The Cognivsl CLI aims at improving the accessibility and usability of cognitive embedding evaluation for NLP practitioners. CogniVal is still under active research and will be extended to additionally support the evaluation of sentence embeddings and further languages.

## Acknowledgements

We thank Antonio de la Torre and Leonard von Kleist for their contributions to the command line interface.

[3] https://github.com/hanxiao/bert-as-service

## References


Samira Abnar, Rasyan Ahmed, Max Mijnheer, and Willem Zuidema. 2018. Experiential, distributional and dependency-based word embeddings have complementary roles in decoding brain activity. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 57–66.

Phillip M Alday. 2019. M/EEG analysis of naturalistic stories: A review from speech to language processing. *Language, Cognition and Neuroscience*, 34(4):457–473.

Amir Bakarov. 2018. Can eye movement data be used as ground truth for word embeddings evaluation? In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lisa Beinborn, Samira Abnar, and Rochelle Choenni. 2019. Robust evaluation of language-brain encoding experiments. *International Journal of Computational Linguistics and Applications*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Allyson Ettinger, Naomi Feldman, Philip Resnik, and Colin Phillips. 2016. Modeling N400 amplitude using vector space models of word representation. In *CogSci*.

Alona Fyshe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2014. Interpretable semantic vectors from a joint model of brain-and text-based meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 489–499.

Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 36–42.

Nora Hollenstein, Antonio de la Torre, Nicolas Langer, and Ce Zhang. 2019. CogniVal: A framework for cognitive word embedding evaluation. In *Proceedings of the 23nd Conference on Computational Natural Language Learning*.

Alexander G Huth, Wendy A de Heer, Thomas L Griffiths, Frédéric E Theunissen, and Jack L Gallant. 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458.

Frank Keller. 2010. Cognitively plausible models of human language processing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: Short Papers*, pages 60–67.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*.

George A Miller and Christiane Fellbaum. 1992. WordNet and the organization of lexical memory. In *Intelligent tutoring systems for foreign language learning*, pages 89–102. Springer.

Malvina Nissim, Rik van Noord, and Rob van der Goot. 2019. Fair is better than sensational: Man is to doctor as woman is to doctor. *arXiv preprint arXiv:1905.09866*.

Joao António Rodrigues, Ruben Branco, João Silva, Chakaveh Saedi, and António Branco. 2018. Predicting brain activation with WordNet embeddings. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 1–5.

Anna Rogers, Shashwath Hosur Ananthakrishna, and Anna Rumshisky. 2018. What's in your embedding, and how it predicts task performance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2690–2703.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*.

Dan Schwartz and Tom Mitchell. 2019. Understanding language-elicited EEG data by predicting it from a fine-tuned language model. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 43–57.

Dan Schwartz, Mariya Toneva, and Leila Wehbe. 2019. Inducing brain-relevant bias in natural language processing models. In *Advances in Neural Information Processing Systems*, pages 14100–14110.

Anders Søgaard. 2016. Evaluating word embeddings with fMRI and eye-tracking. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 116–121.

Mariya Toneva and Leila Wehbe. 2019. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain). In *Advances in Neural Information Processing Systems*, pages 14928–14938.