Personality as Search: Automated Big Five Assessment with BERT (ColB5ERT)

Ricardo Primi

University of São Francisco & Edulab21, Ayrton Senna Institute, Brazil

R2D2
Powered by GPT8

Research question

Since digital footprints provide immense sample of people’s behavior (Matz, 2025)

Can we measure a person’s big five traits (but also values, moral foundations, interests) from their written texts?

- Spontaneous texts: Posts, written messages on social media.
- Elicited texts: Autobiographies, written responses to structured questions about personality traits.

Can be a continuous monitor for assessment

A new general method for assessment.

Method Overview

Framing self report assessment as Computer Science Information Search

Training a BERT model using Sentence Transformers / Colbert = ColB5ert

Using the model to score texts

Studying the statistical distributions automated measures of B5 and validity

Personality assessment as search

Basic Elements of Information Search in Computer Science

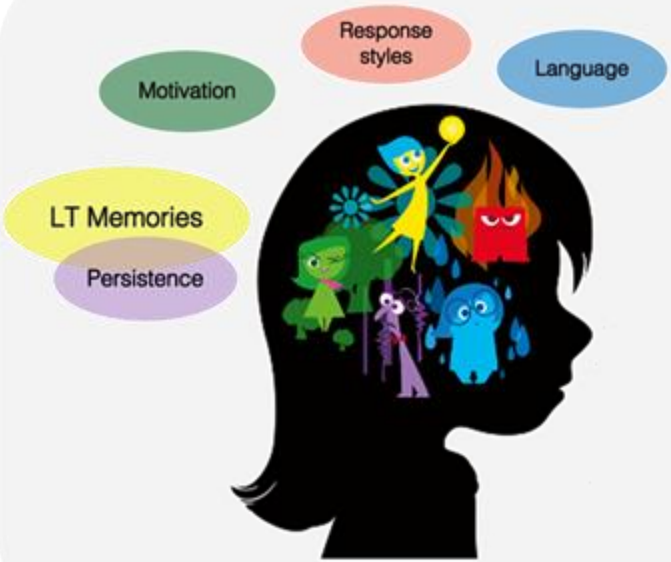
Query: The user's input or question expressed in a natural or structured language.

Document Corpus: The collection of documents or data sources to be searched.

Search algorithms by semantic similarity: The methods used to match queries with relevant documents based on words (past) or more recently contextual word vectors (embeddings)

Relevance Scoring: Ranking documents based on their semantic similarity to the query.

Retrieval: Fetching the most relevant documents based on the relevance scores



When confronted
with a problem:

- I give up easily ?
- I do more ?
- I remain interested?

The cognitive process of Self-report

Duckworth AL, Yeager DS. Measurement Matters: Assessing Personal Qualities Other Than Cognitive Ability for Educational Purposes. Educ Res. 2015 May;44(4):237-251. doi: 10.3102/0013189X15584327. PMID: 27134288; PMCID: PMC4849415.

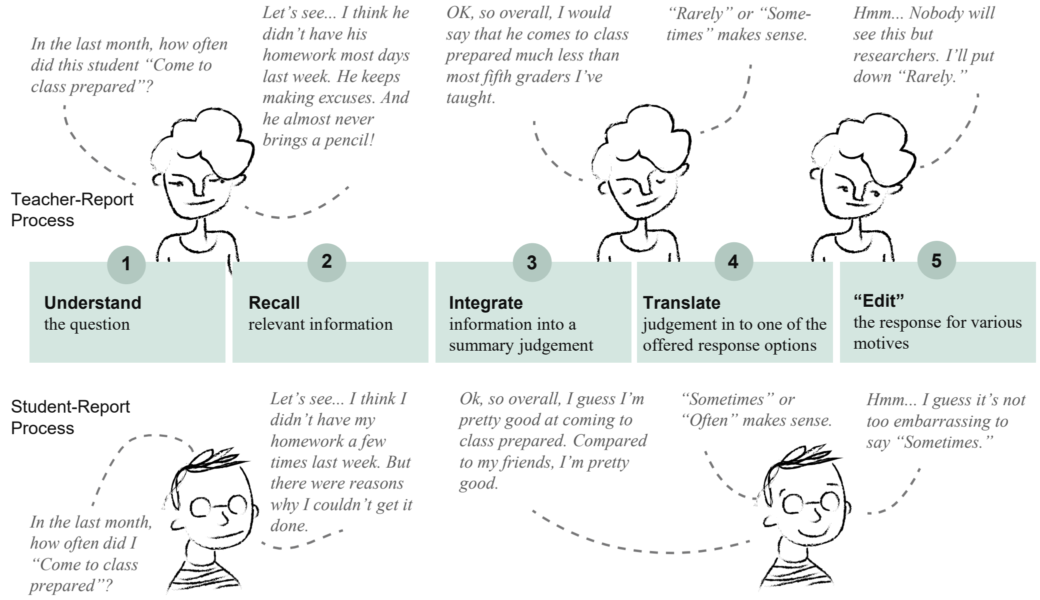


FIGURE 1. *The process by which students and teachers respond to questionnaire items*

Elements of Search on Self-report

Query

Search algorithms by semantic similarity

Retrieval

Relevance Scoring

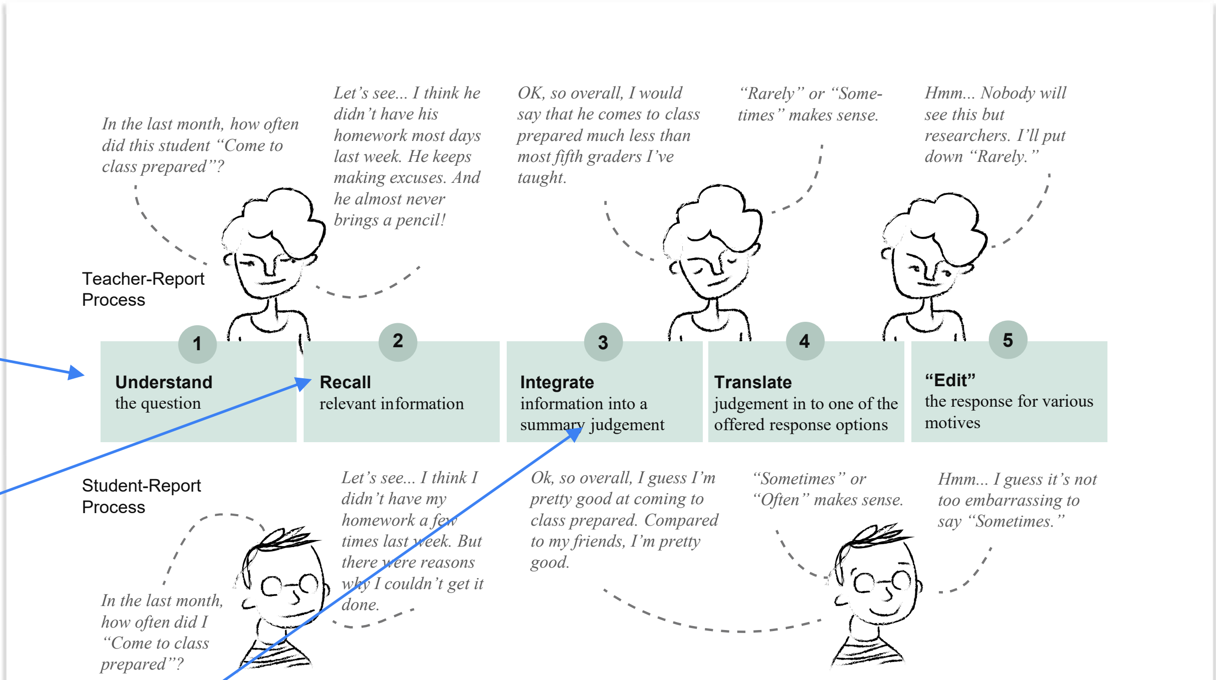
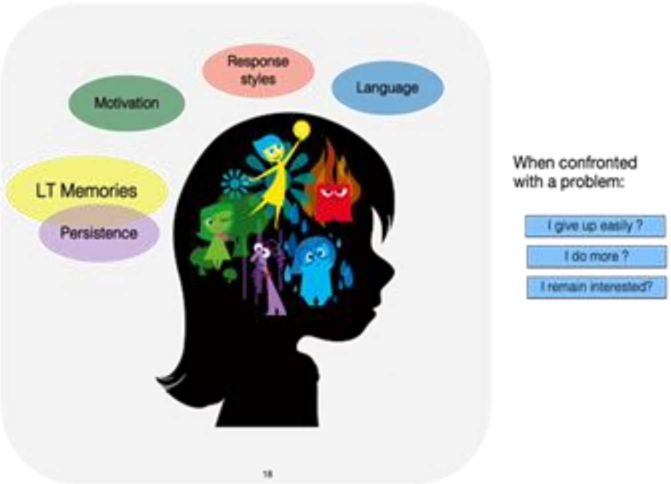


FIGURE 1. *The process by which students and teachers respond to questionnaire items*

Personality assessment and Information Search



Statements/items of a test are like questions/queries.

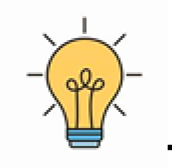
The life story stored in long-term memory consists of passages from a comprehensive document that represents the person's complete autobiography (**person life's story corpus**).

The Likert response to an item is a **relevance score** of the query (items) in signaling that relevant passage was retrieved !

A measure of personality is an aggregation of relevance scores by query category (extroversion, openness, etc.).

We rely on the person's **internal search mechanism** to find degree of similarity between items and memories

Elements of IS	Information Search	Personality Assessment	Digital footprints (posts in FB)
Purpose	Find relevant documents	Score queries (itens) items organized by a scheme (psychological theory)	Score queries (itens)
Query	Subject generated	Professional/research generated (limited)	Professional/research generated (potentially unlimited)
Document	An entire knowledge base from which we want to recover a few relevant docs	Personal History from which we want to detect if a query match relevant LT memories	All history of posts from which we want to score specific queries
Search process	Exhaustive	Selective filtering	Exhaustive
Retrieval	Retrieval of a subset with high relevance scores	Specific fact(s) / memory(ies) or experience of familiarity	Exhaustive
Relevance score	Number of words or sum of words semantic similarity between queries and documents	Likert response, score on scales (how much item words were part of personal history)	Semantic similarity between queries and posts



AI: Transformers 101
BERT

How to measure semantic similarity between texts (spontaneous texts and personality items)

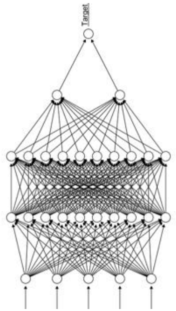
101 on Transformers

GP**T**: General Pré Trained Transformers

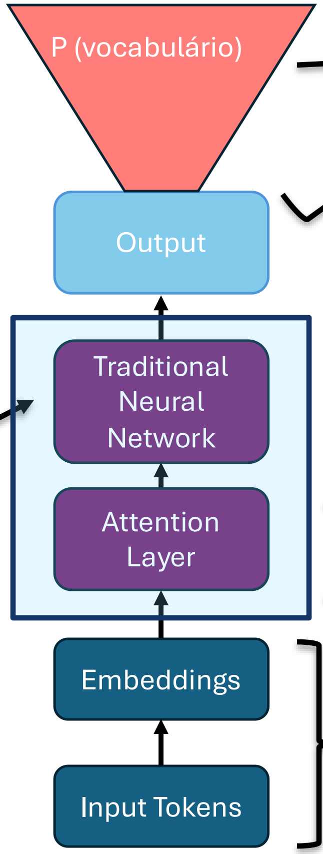
BER**T**: Bi-Directional Encoders from Transformers

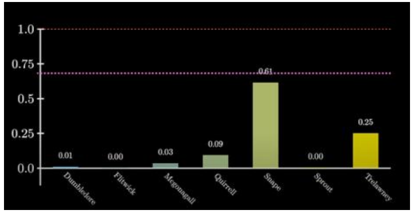
Rich Latent Representations of texts: Contextualized Embeddings

Simplified scheme of a Transformer (GPT and BERT)



N layers





Produces a distribution of the probabilities for the words in the vocabulary being the next word

Transformed word vectors representing the meaning of words/text

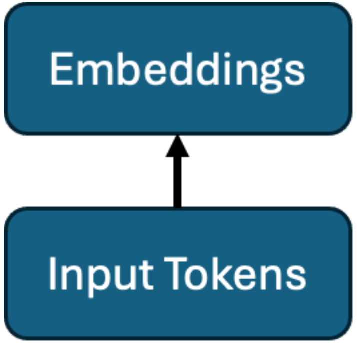
Calculates interaction between word vectors (relationship between word semantic meanings).

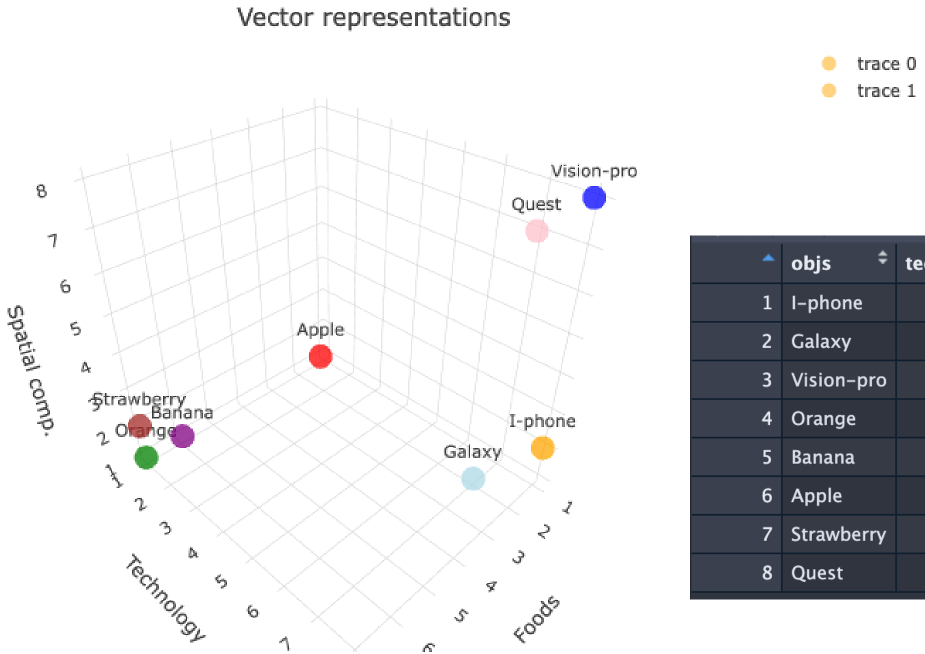
Transforms words into numerical vectors (semantic meaning)

Embeddings: From Words to Meaning: How LLMs Represent Language

- Words (tokens) are converted into **vectors of continuous numbers** → *embeddings*
- Embeddings capture **relationships** between words
- Encode **semantic meaning**, not just surface form
- Encode **deep abstractions structures**: grammar, code, symbolic math, logic systems
    - Emerged from training on **billions of words** from diverse sources and Large context windows that let models analyze **meaning across long passages**

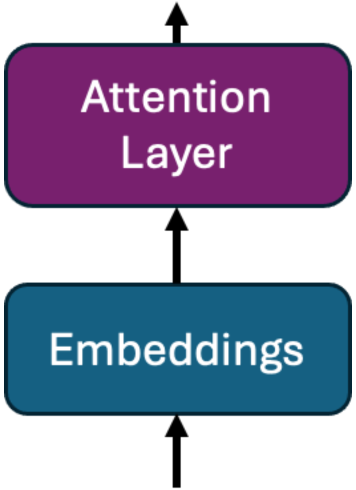
I bought a box of apples.
I bought an apple i-phone.



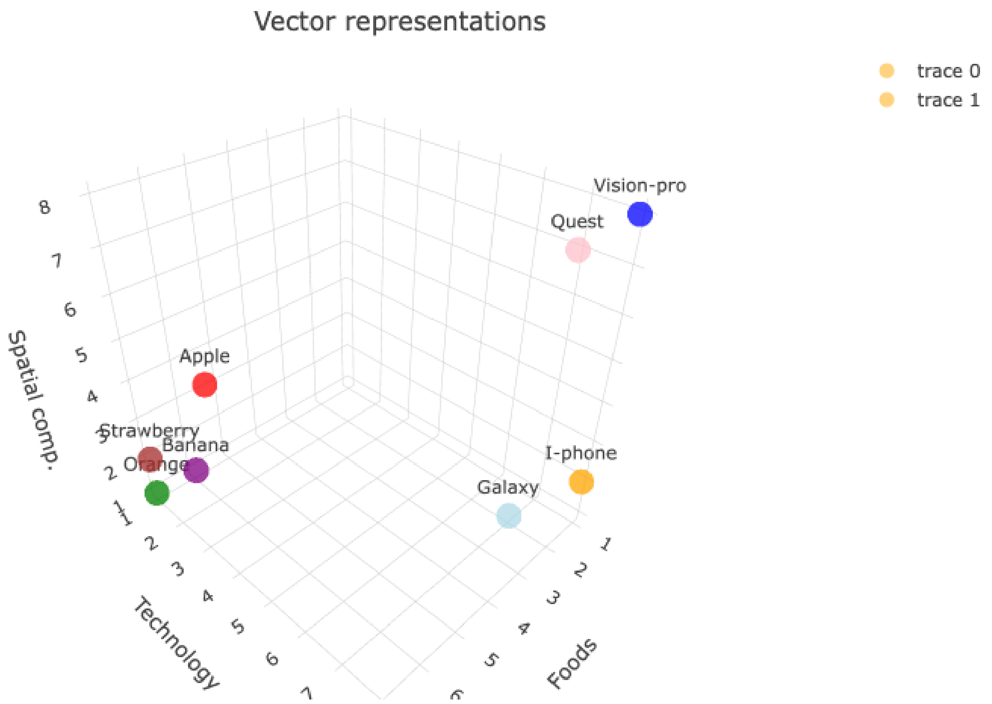


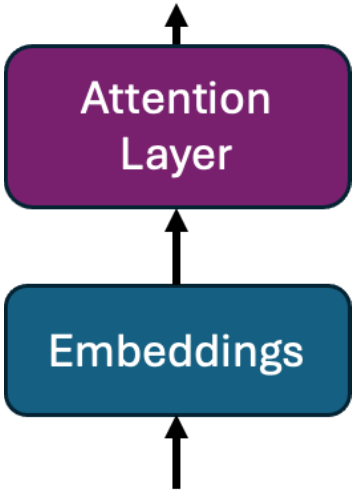
	objs	tech	alim	spatial	colors
1	I-phone	8.1	1.1	2.1	orange
2	Galaxy	7.0	2.0	1.0	lightblue
3	Vision-pro	8.0	1.0	8.0	blue
4	Orange	1.0	7.0	1.0	green
5	Banana	2.0	6.5	2.0	purple
6	Apple	4.0	4.0	4.0	red
7	Strawberry	1.0	7.0	2.0	brown
8	Quest	7.0	1.0	7.0	pink

- Each word has a numerical vector of dimension D (GPT-4: 1536 dimensions).
- Numbers are learned such that each dimension encodes a semantic attribute.
- Similar words have similar numerical vectors, which means they are close to each other in the D-dimensional space.
- Therefore, the correlation between vectors (cosine similarity) is higher for similar words.

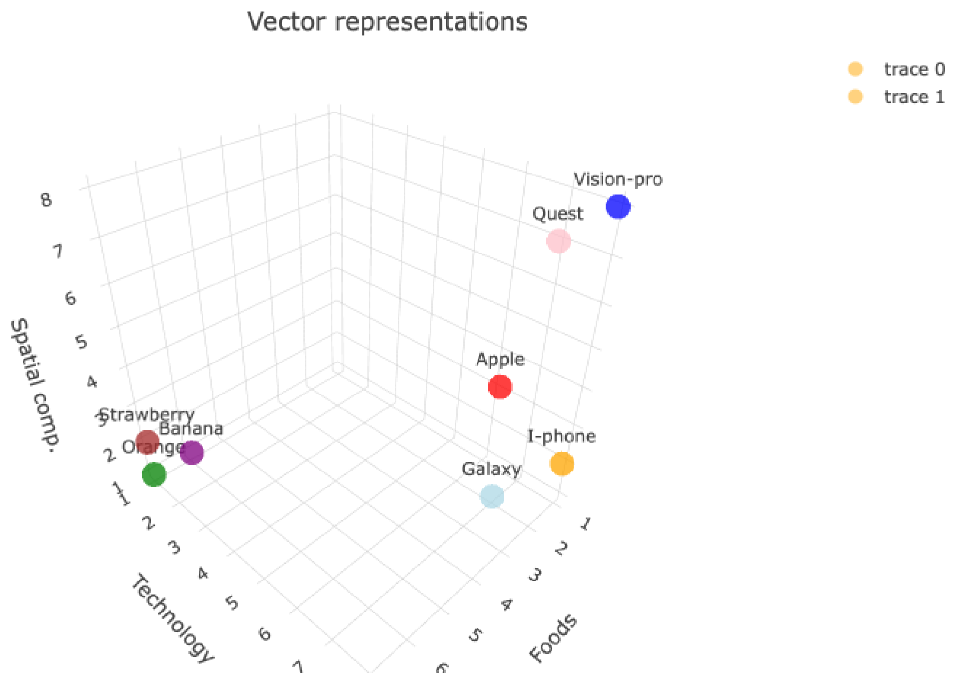


I bought a box of apples

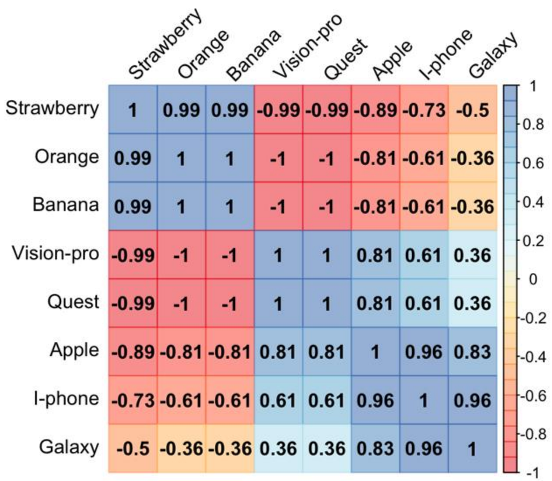




I bought an apple i-phone.



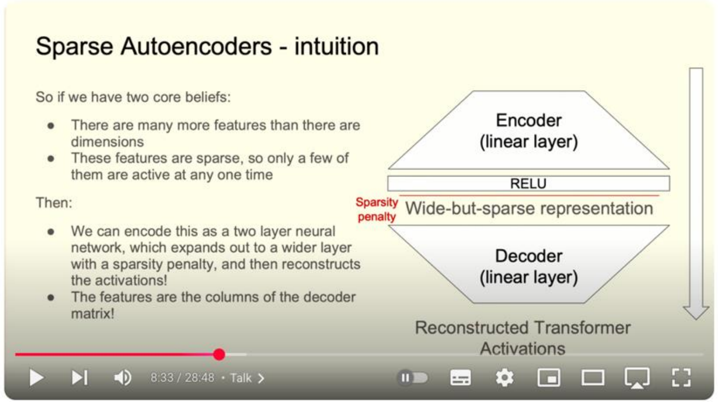
	objs	tech	alim	spatial	colors
1	I-phone	8.1	1.1	2.1	orange
2	Galaxy	7.0	2.0	1.0	lightblue
3	Vision-pro	8.0	1.0	8.0	blue
4	Orange	1.0	7.0	1.0	green
5	Banana	2.0	6.5	2.0	purple
6	Apple	4.0	4.0	4.0	red
7	Strawberry	1.0	7.0	2.0	brown
8	Quest	7.0	1.0	7.0	pink



Semantic similarity between words:
  Cosine similarity among embeddings (word vectors)
  Correlations / Profile similarity

Sentence Semantic similarity:
  Average embeddings of sentence words
  Calculate similarity between resulting vectors of sentences

But ..caution on interpreting embedding dimensions as latent variables .. there are much more features than dimensions .. superposition



Hoagy Cunningham — Finding distributed features in LLMs with sparse autoencoders [TAIS 2024]

Anthropic

https://www.youtube.com/watch?v=HPLII9ZOpUQ

https://transformer-circuits.pub/2025/attribution-graphs/biology.html

Empirical Study

Measuring Big Five on FB posts

How to measure Big Five with AI ?

BIG 5: Personality traits organized into five broad domains: **O**penness, **C**onscientiousness, **E**xtraversion, **A**mability and **N**egative Emotional Regulation

If we have sentence embeddings for big five personality items

Text (FB posts) embeddings

We can score semantic similarity of items-text pairs by calculating the cosine similarity

Goals

Fine Tune BERT to maximize detection of big five knowledge

BERT general language

Fine tunning BERT to the task of scoring Big Five

ColB5BERT / Sentence BERT

Initialize the model in its original weights

Train with a new data with a new specialized objective

Datasets documents

Building a Corpus for Personality-dependent Natural Language Understanding and Generation

R.M.S. Ramos, G.B.S. Neto, B.B.C. Silva, D.S. Monteiro, I. Paraboni, R.F.S. Dias

University of São Paulo, School of Arts, Sciences and Humanities

São Paulo, Brazil

Abstract

1. Introduction

2. Background

1138

B5 corpus (public dataset of facebook posts + responses to BFI)

N = 1,082 participants e 194,382 posts e 2,219,585 words (tokens)

We divided posts into chunks of approximately 250 words (11,537 texts segments)

415 items from: SENNA , BFI-2 and OECD Study of SEMS (domains and facets)

b5 corpus v.1.8.zip

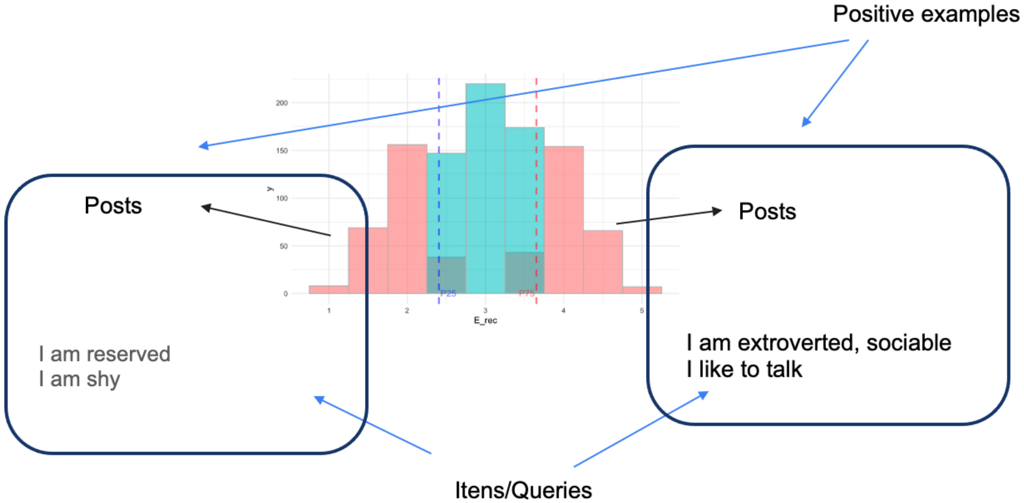
Dataset creation for finetuning BERT

I paired posts with items

415 items X 11537 texts = 4.787.855 possible interactions

Marked positive examples

Marked negative example



Then I created triplets:

post *vs* positive item example *vs* negative item example

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

Nils Reimers and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) has set a new state-of-the-art performance on sentence-pair regression tasks like semantic textual similarity (STS). However, it requires that both sentences are fed into the network, which causes a massive computational overhead: Finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations (~65 hours) with BERT. The construction of BERT makes it unsuitable for semantic similarity search as well as for unsupervised tasks like clustering.

In this publication, we present Sentence-BERT (SBERT), a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT.

We evaluate SBERT and SRoBERTa on common STS tasks and transfer learning tasks, where it outperforms other state-of-the-art sentence embeddings methods.[1]

1 Introduction

In this publication, we present Sentence-BERT (SBERT), a modification of the BERT network using siamese and triplet networks that is able to derive semantically meaningful sentence embeddings[2]. This enables BERT to be used for certain new tasks, which up-to-now were not applicable for BERT. These tasks include large-scale semantic similarity comparison, clustering, and information retrieval via semantic search.

BERT set new state-of-the-art performance on various sentence classification and sentence-pair regression tasks. BERT uses a cross-encoder: Two sentences are passed to the transformer network and the target value is predicted. However, this setup is unsuitable for various pair regression tasks due to too many possible combinations. Finding in a collection of $n = 10\,000$ sentences the pair with the highest similarity requires with BERT $n \cdot (n-1)/2 = 49\,995\,000$ inference computations. On a modern V100 GPU, this requires about 65 hours. Similar, finding which of the over 40 million existent questions of Quora is the most similar for a new question could be modeled as a pair-wise comparison with BERT, however, answering a single query would require over 50 hours.

A common method to address clustering and semantic search is to map each sentence to a vector space such that semantically similar sentences are close. Researchers have started to input individual sentences into BERT and to derive fixed-size sentence embeddings. The most commonly used approach is to average the BERT output layer (known as BERT embeddings) or by using the output of the first token (the `[CLS]` token). As we will show, this common practice yields rather bad sentence embeddings, often worse than averaging GloVe embeddings (Pennington et al., 2014).

To alleviate this issue, we developed SBERT. The siamese network architecture enables that fixed-sized vectors for input sentences can be derived. Using a similarity measure like cosine-similarity or Manhatten / Euclidean distance, semantically similar sentences can be found. These similarity measures can be performed extremely efficient on modern hardware, allowing SBERT to be used for semantic similarity search as well as for clustering. The complexity for finding the

[1] Code available: https://github.com/UKPLab/sentence-transformers

[2] With *semantically meaningful* we mean that semantically similar sentences are close in vector space.

ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Omar Khattab
Stanford University
okhattab@stanford.edu

Matei Zaharia
Stanford University
matei@cs.stanford.edu

ABSTRACT

Recent progress in Natural Language Understanding (NLU) is driving fast-paced advances in Information Retrieval (IR), largely owed to fine-tuning deep language models (LMs) for document ranking. While remarkably effective, the ranking models based on these LMs increase computational cost by orders of magnitude over prior approaches, particularly as they must feed each query–document pair through a massive neural network to compute a single relevance score. To tackle this, we present ColBERT, a novel ranking model that adapts deep LMs (in particular, BERT) for efficient retrieval. ColBERT introduces a *late interaction* architecture that independently encodes the query and the document using BERT and then employs a cheap yet powerful interaction step that models their fine-grained similarity. By delaying and yet retaining this fine-granular interaction, ColBERT can leverage the expressiveness of deep LMs while simultaneously gaining the ability to pre-compute document representations offline, considerably speeding up query processing. Beyond reducing the cost of re-ranking the documents retrieved by a traditional model, ColBERT's *pruning-friendly* interaction mechanism enables leveraging vector-similarity indexes for end-to-end retrieval directly from a large document collection. We extensively evaluate ColBERT using two recent passage search datasets. Results show that ColBERT's effectiveness is competitive with existing BERT-based models (and outperforms every non-BERT baseline), while executing two orders-of-magnitude faster and requiring four orders-of-magnitude fewer FLOPs per query.

ACM Reference format:
Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, July 25–30, 2020 (SIGIR '20)*, 10 pages.
DOI: 10.1145/3397271.3401075

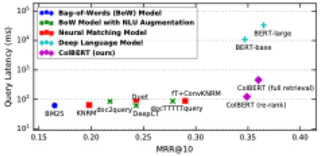


Figure 1: Effectiveness (MRR@10) versus Mean Query Latency (log-scale) for a number of representative ranking models on MS MARCO Ranking [24]. The figure also shows ColBERT. Neural re-rankers run on top of the official BM25 top-1000 results and use a Tesla V100 GPU. Methodology and detailed results are in §4.

1 INTRODUCTION

Over the past few years, the Information Retrieval (IR) community has witnessed the introduction of a host of neural ranking models, including DRMM [7], KNRM [4, 36], and Duet [20, 22]. In contrast to prior learning-to-rank methods that rely on hand-crafted features, these models employ embedding-based representations of queries and documents and directly model *local interactions* (i.e., fine-granular relationships) between their contents. Among them, a recent approach has emerged that *fine-tunes* deep pre-trained language models (LMs) like ELMo [29] and BERT [5] for estimating relevance. By computing deeply-contextualized semantic representations of query–document pairs, these LMs help bridge the pervasive vocabulary mismatch [21, 42] between documents and queries [30]. Indeed, in the span of just a few months, a number of ranking models based on BERT have achieved state-of-the-art results on various retrieval benchmarks [3, 18, 25, 39] and have been proprietarily adapted for deployment by Google[1] and Bing[2].

However, the remarkable gains delivered by these LMs come at a steep increase in computational cost. Hofstätter *et al.* [9] and MacAvaney *et al.* [18] observe that BERT-based models in the literature are 100-1000× more computationally expensive than prior models—some of which are arguably *not* inexpensive to begin with [13]. This quality–cost tradeoff is summarized by Figure 1, which compares two BERT-based rankers [25, 27] against a representative set of ranking models. The figure uses MS MARCO Ranking [24], a recent collection of 9M passages and 1M queries from Bing's logs. It reports retrieval effectiveness (MRR@10) on the official validation set as well as average query latency (log-scale) using a high-end server that dedicates one Tesla V100 GPU per query for neural re-rankers. Following the *re-ranking* setup of MS MARCO, ColBERT (re-rank), the Neural Matching Models, and the Deep LMs re-rank the MS MARCO's official top-1000 documents per query.

SIGIR '20, Virtual Event, China

978-1-4503-8016-4/20/07...$15.00
DOI: 10.1145/3397271.3401075

[1] https://blog.google/products/search/search-language-understanding-bert/

[2] https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-experience-using-azure-gpus/

Sentence Transformer

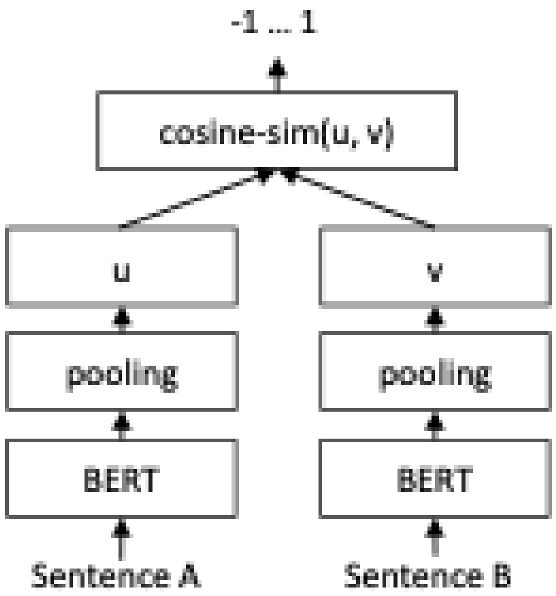


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

ColBERT

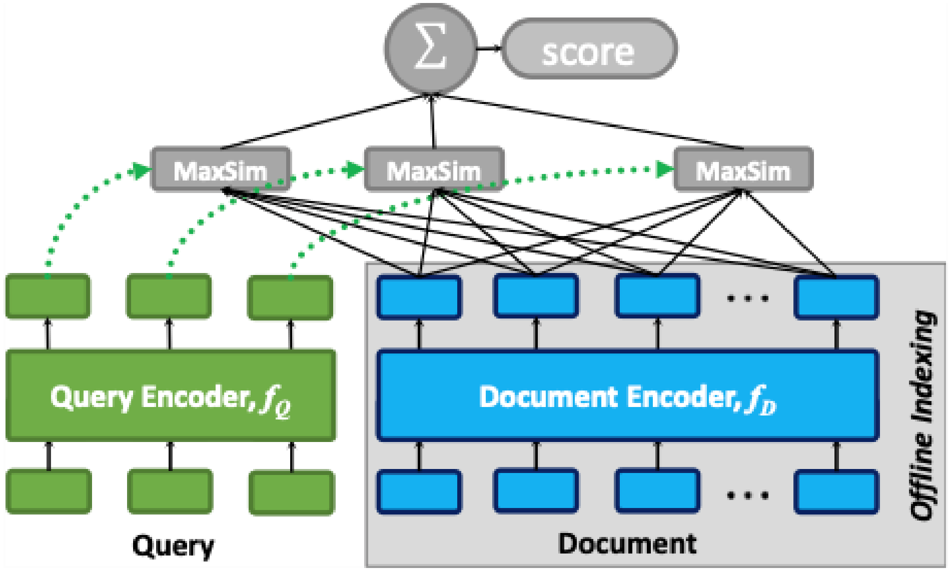
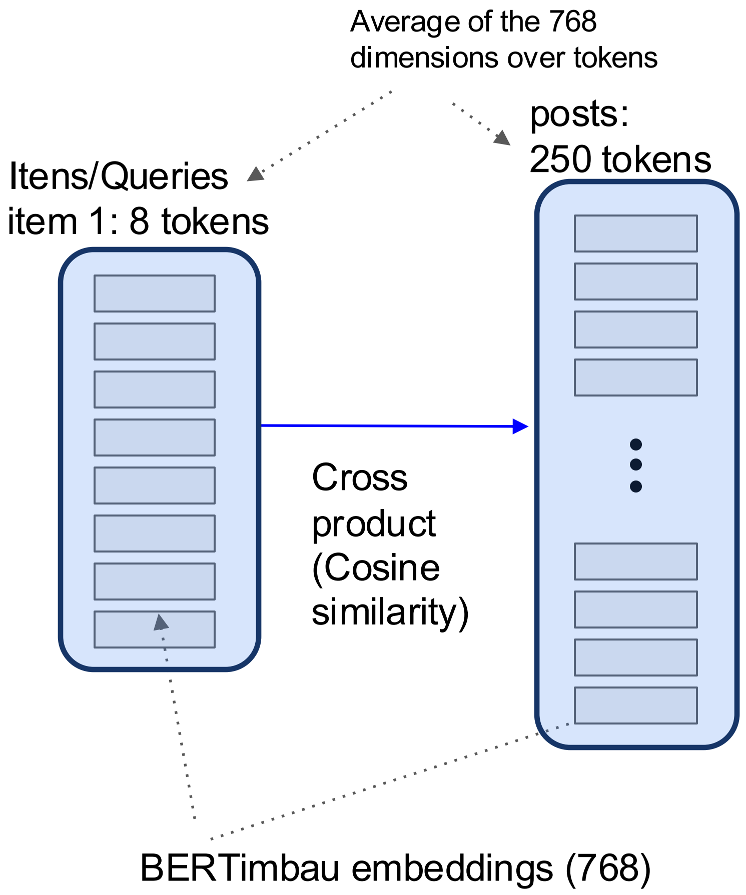


Figure 3: The general architecture of ColBERT given a query q and a document d.

Training objective (loss function)



Training Objective
Similarity between positive examples is greather than negaive examples

Training and Testing Data:
Training Set: 993,759 interactions (N=868)
Test Set: 158,594 interactions (N=152)

Training Details:
Epochs: 3 (each epoch took 6 hours and 40 minutes in a GPU Nvidia A100 batches of 80 triplets)

The model used is BERTimbau, pretrained in Portuguese (https://github.com/neuralmind-ai/portuguese-bert).

Model Application (scoring posts for Big Five):
Used the model to score each interaction item by post in the test set (Created semantic similarity scores: sim_scores for each item int he item database)

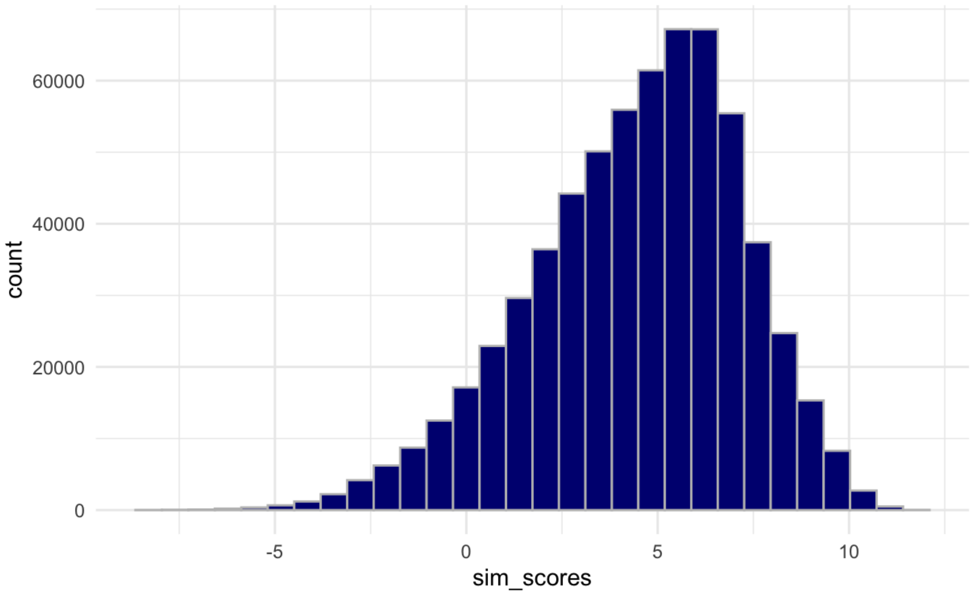
Generated scores for domains and facets by averaging sim_scores for each domain/facet

post vs itens of domain

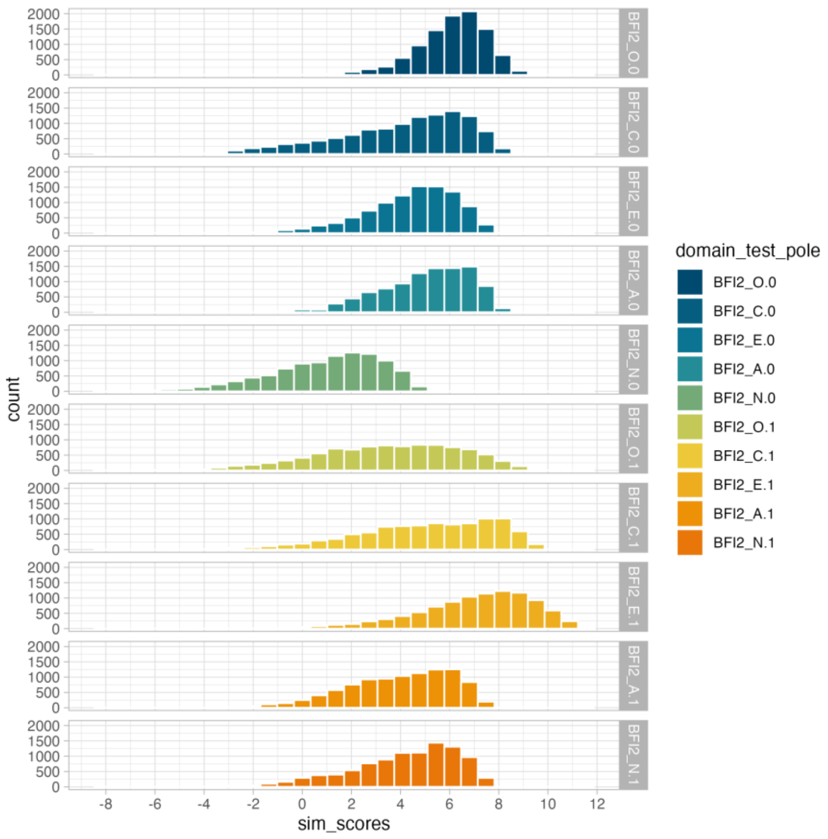
Findings

- What most common theme of big five was seen in posts?
- What is the relationship between subject BFI scores and personality scores from their posts ?

What is the distribution of sim_scores?



What is the distribution of sim_scores by domain / pole test of B5?



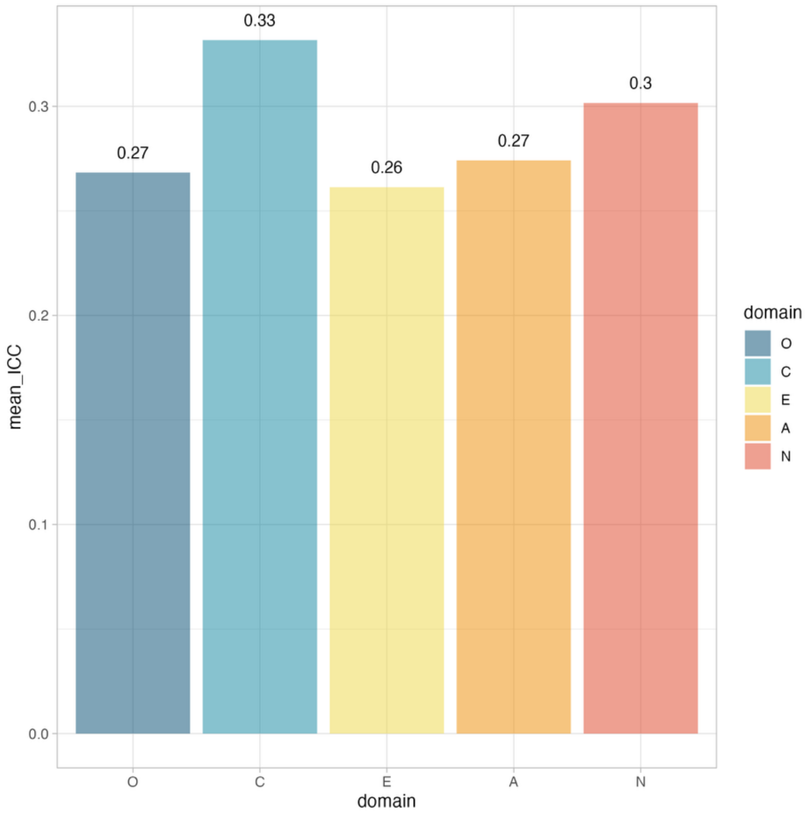
Multilevel structure of the data

Person (Level 2)

Persons + domain

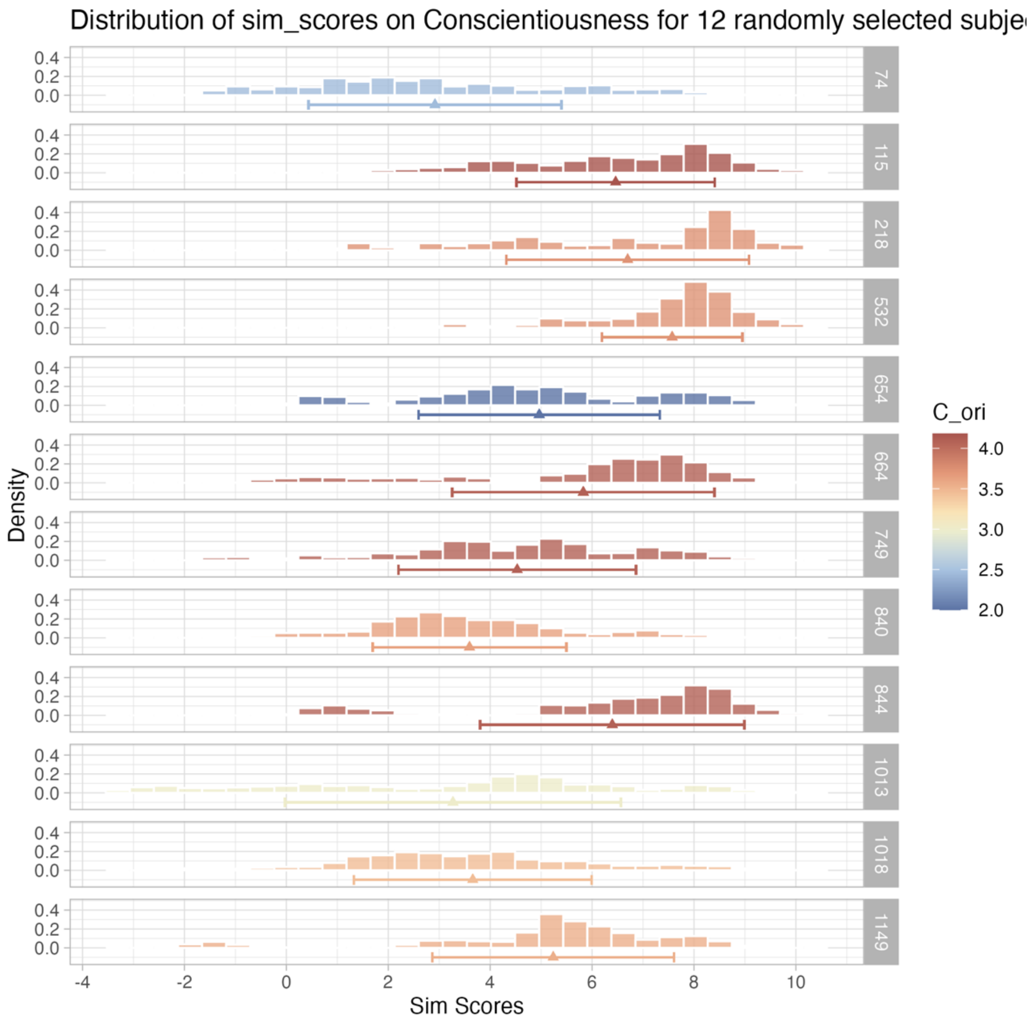
FB Posts (Level 1)

ICC's

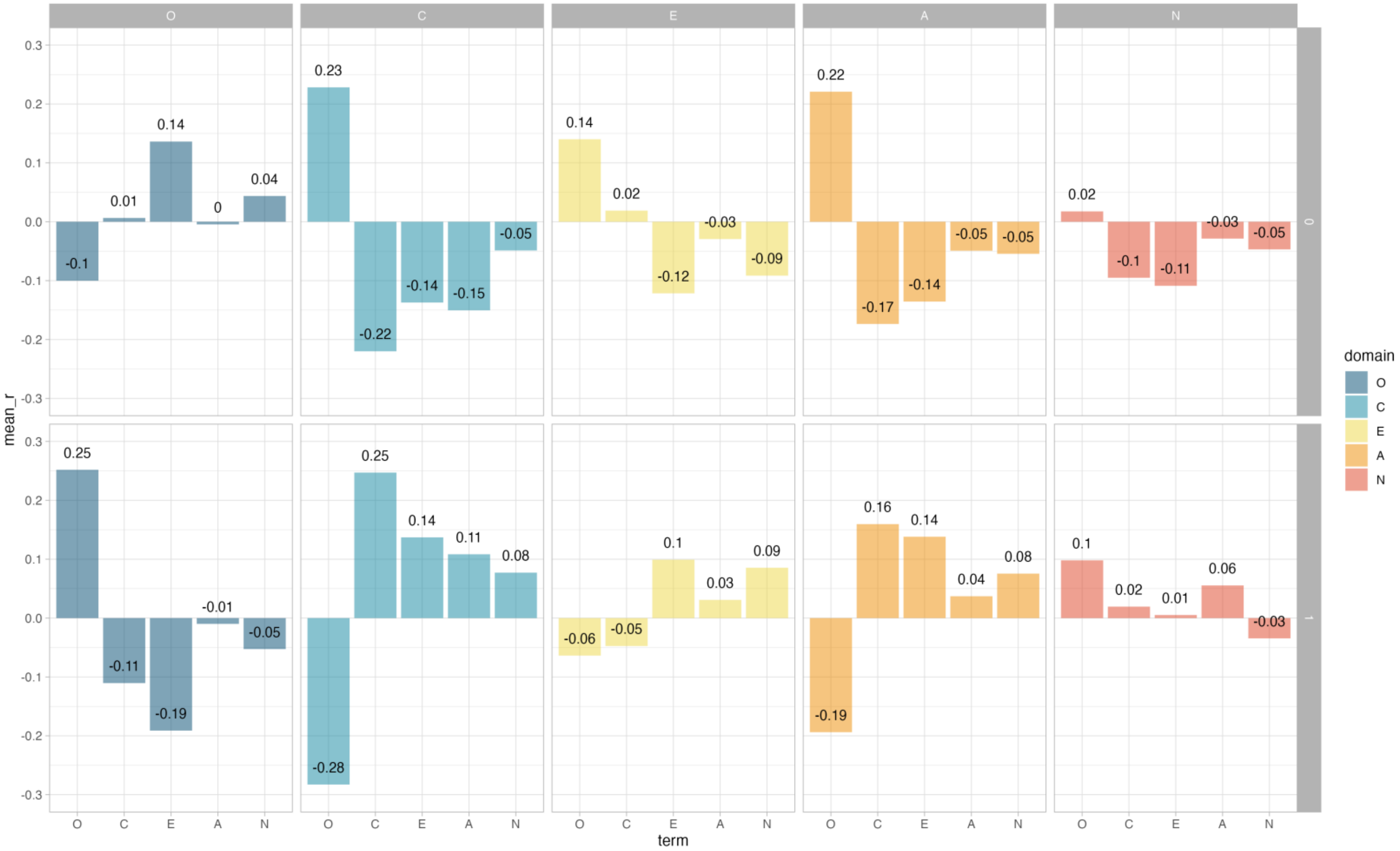


Probabilistic Density Function

Distributions of sim_scores for C of 12 randomly selected subjects



How self-reported B5 (BFI) correlates with sim_scores by domain and pole ?



Regression predicting self-reported B5 in BFI from average sim_scores of test/domain/pole

BFI2		
BFI	R2	R
O	0.249	0.50
C	0.138	0.37
E	0.151	0.39
A	0.059	0.24
N	0.081	0.28

Journal of Personality and Social Psychology

0022-3514/14/$12.00 http://dx.doi.org/10.1037/pspp0000020

Automatic Personality Assessment Through Social Media Language

Gregory Park, H. Andrew Schwartz,
Johannes C. Eichstaedt, and Margaret L. Kern
University of Pennsylvania

Michal Kosinski and David J. Stillwell
University of Cambridge

Lyle H. Ungar and Martin E. P. Seligman
University of Pennsylvania

Language use is a psychologically rich, stable individual difference with well-established correlations to personality. We describe a method for assessing personality using an open-vocabulary analysis of language from social media. We compiled the written language from 66,732 Facebook users and their questionnaire-based self-reported Big Five personality traits, and then we built a predictive model of personality based on their language. We used this model to predict the 5 personality factors in a separate sample of 4,824 Facebook users, examining (a) convergence with self-reports of personality at the domain- and facet-level; (b) discriminant validity between predictions of distinct traits; (c) agreement with informant reports of personality; (d) patterns of correlations with external criteria (e.g., number of friends, political attitudes, impulsiveness); and (e) test–retest reliability over 6-month intervals. Results indicated that language-based assessments can constitute valid personality measures: they agreed with self-reports and informant reports of personality, added incremental validity over informant reports, adequately discriminated between traits, exhibited patterns of correlations with external criteria similar to those found with self-reported personality, and were stable over 6-month intervals. Analysis of predictive language can provide rich portraits of the mental life associated with traits. This approach can complement and extend traditional methods, providing researchers with an additional measure that can quickly and cheaply assess large groups of participants with minimal burden.

Keywords: language, personality assessment, measurement, big data, social media

Supplemental materials: http://dx.doi.org/10.1037/pspp0000020.supp

Table 2
Correlations Between Language-Based Assessments and Self-Reports of Big Five Personality Traits

	Self-reports					Language-based assessments				
	O	C	E	A	N	O	C	E	A	N
Self-reports										
Openness										
Conscientiousness	*.00*									
Extraversion	*.13*	*.19*								
Agreeableness	*.07*	*.17*	*.19*							
Neuroticism	*−.08*	*−.31*	*−.34*	*−.36*						
Language-based										
Openness	**.43**	−.12	−.08	−.05	.00					
Conscientiousness	−.13	**.37**	.16	.17	−.17	*−.25*				
Extraversion	−.07	.12	**.42**	.10	−.15	*−.17*	*.33*			
Agreeableness	−.07	.17	.13	**.35**	−.14	*−.12*	*.44*	*.27*		
Neuroticism	.05	−.17	−.18	−.13	**.35**	*.06*	*−.41*	*−.43*	*−.34*	

Note. N = 4,824. O = Openness to Experience; C = Conscientiousness; E = Extraversion; A = Agreeableness; N = Neuroticism. Convergent correlations are in bold; discriminant correlations are in italics.

Conclusions and discussion

- Benchmarking Results:
  - Our methods consistently achieved the upper-level benchmarks
    Non trait-activated
- Innovative Methodology

  Step 1 Assessed texts scoring the big five.
  Step 2. Aggregated by domain to score subjects

  - General Text Analysis by psychological themes: Provides a new, effective way to study texts, harnessing the richness of spontaneous, user-generated content.
  - Automated Insights: Leverages AI to gain psychological insights
  - Scalability: The method is adaptable and can be applied to any test !
- Bridging AI and Psychology:
  - AI Enhancing Psychology: Utilizes advanced AI techniques to deepen psychological understanding, offering more nuanced and precise analyses of human behavior.
  - Psychology Informing AI: Psychological theories guide the development of AI models, ensuring they are more attuned to human nuances and behaviors.

Resources

Behavior Research Methods
https://doi.org/10.3758/s13428-024-02455-8

ORIGINAL MANUSCRIPT

A tutorial on open-source large language models for behavioral science

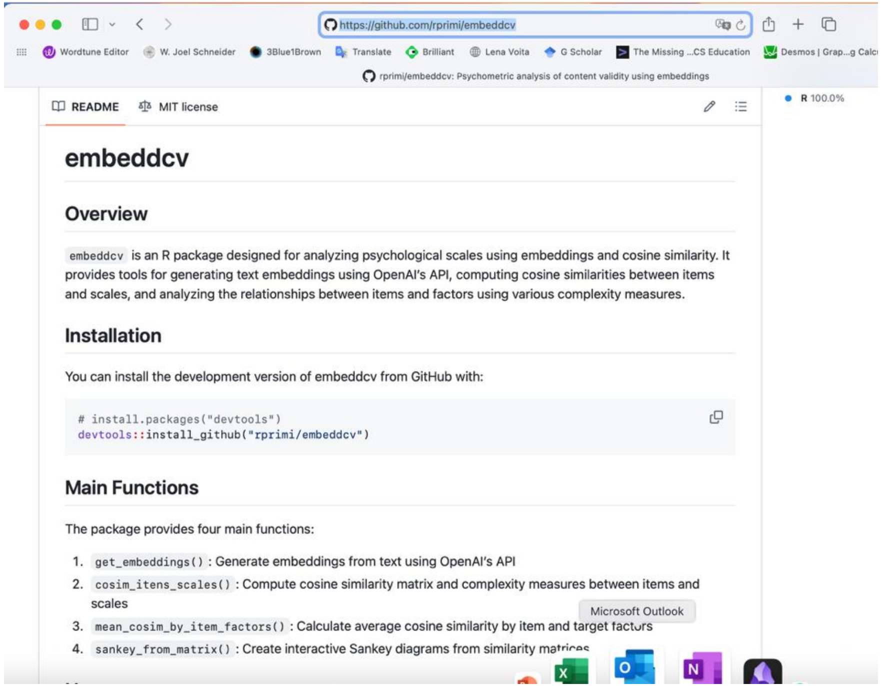
Zak Hussain[1,2] · Marcel Binz[3,4] · Rui Mata[1] · Dirk U. Wulff[1,2]

Accepted: 27 May 2024

Abstract
Large language models (LLMs) have the potential to revolutionize behavioral science by accelerating and improving the research cycle, from conceptualization to data analysis. Unlike closed-source solutions, open-source frameworks for LLMs can enable transparency, reproducibility, and adherence to data protection standards, which gives them a crucial advantage for use in behavioral science. To help researchers harness the promise of LLMs, this tutorial offers a primer on the open-source Hugging Face ecosystem and demonstrates several applications that advance conceptual and empirical work in behavioral science, including feature extraction, fine-tuning of models for prediction, and generation of behavioral responses. Executable code is made available at github.com/Zak-Hussain/LLM4BeSci.git. Finally, the tutorial discusses challenges faced by research with (open-source) LLMs related to interpretability and safety and offers a perspective on future research at the intersection of language modeling and behavioral science.

Keywords Large language models · Behavioral science · Hugging face

https://github.com/rprimi/embeddcv



embeddcv

Overview

`embeddcv` is an R package designed for analyzing psychological scales using embeddings and cosine similarity. It provides tools for generating text embeddings using OpenAI's API, computing cosine similarities between items and scales, and analyzing the relationships between items and factors using various complexity measures.

Installation

You can install the development version of embeddcv from GitHub with:

```
# install.packages("devtools")
devtools::install_github("rprimi/embeddcv")
```

Main Functions

The package provides four main functions:

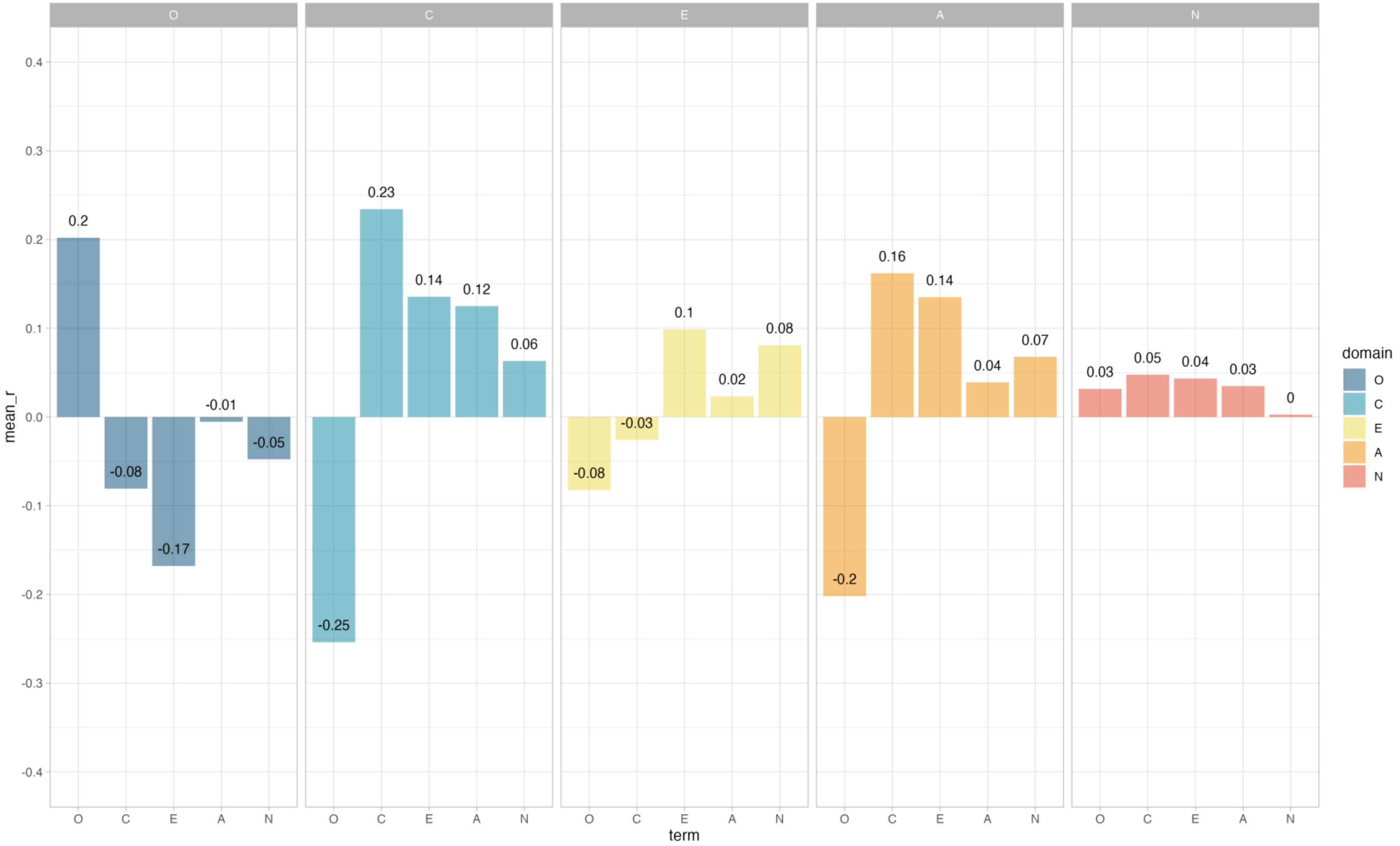
1. `get_embeddings()` : Generate embeddings from text using OpenAI's API
2. `cosim_itens_scales()` : Compute cosine similarity matrix and complexity measures between items and scales
3. `mean_cosim_by_item_factors()` : Calculate average cosine similarity by item and target factors
4. `sankey_from_matrix()` : Create interactive Sankey diagrams from similarity matrices

Thank You ... rprimi@mac.com

USF
UNIVERSIDADE SÃO FRANCISCO

Instituto
Ayrton
Senna

Aditional info


O
C
E
A
N
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
mean_r
0.2
-0.08
-0.17
-0.01
-0.05
-0.25
0.23
0.14
0.12
0.06
-0.08
-0.03
0.1
0.02
0.08
-0.2
0.16
0.14
0.04
0.07
0.03
0.05
0.04
0.03
0
term
domain

vars	O_rec	C_rec	E_rec	A_rec	N_rec
BFI2_A_cmp_0	0.15	-0.14	-0.08	-0.03	-0.03
BFI2_A_cmp_1	-0.13	0.12	0.06	0.03	0.06
BFI2_A_res_0	0.16	-0.14	-0.08	-0.03	-0.04
BFI2_A_res_1	-0.13	0.12	0.06	0.03	0.05
BFI2_A_trst_0	0.16	-0.14	-0.08	-0.03	-0.04
BFI2_A_trst_1	-0.13	0.12	0.06	0.03	0.06
BFI2_C_org_0	0.18	-0.16	-0.14	-0.10	-0.04
BFI2_C_org_1	-0.22	0.19	0.07	0.08	0.06
BFI2_C_prod_0	0.18	-0.16	-0.14	-0.10	-0.04
BFI2_C_prod_1	-0.22	0.19	0.07	0.08	0.06
BFI2_C_rsp_0	0.19	-0.16	-0.14	-0.10	-0.04
BFI2_C_rsp_1	-0.22	0.19	0.07	0.08	0.06
BFI2_E_ass_0	0.12	0.01	-0.18	-0.04	-0.06
BFI2_E_ass_1	-0.06	-0.04	0.14	0.04	0.04
BFI2_E_enrg_0	0.12	0.01	-0.18	-0.04	-0.07
BFI2_E_enrg_1	-0.06	-0.05	0.14	0.04	0.04
BFI2_E_soc_0	0.12	0.01	-0.18	-0.04	-0.06
BFI2_E_soc_1	-0.06	-0.05	0.14	0.04	0.04
BFI2_N_anx_0	0.07	-0.07	-0.11	-0.03	-0.04
BFI2_N_anx_1	0.06	0.02	0.02	0.04	0.01
BFI2_N_dep_0	0.07	-0.07	-0.11	-0.03	-0.04
BFI2_N_dep_1	0.06	0.02	0.01	0.04	0.01
BFI2_N_vlti_0	0.07	-0.07	-0.11	-0.03	-0.04
BFI2_N_vlti_1	0.06	0.02	0.01	0.04	0.01
BFI2_O_aes_0	-0.11	0.06	0.04	-0.01	0.02
BFI2_O_aes_1	0.22	-0.12	-0.10	0.01	-0.03
BFI2_O_crea_0	-0.11	0.06	0.04	-0.01	0.02
BFI2_O_crea_1	0.22	-0.12	-0.10	0.01	-0.03
BFI2_O_int_0	-0.11	0.05	0.04	-0.01	0.02
BFI2_O_int_1	0.22	-0.12	-0.10	0.01	-0.03

vars	O_rec	C_rec	E_rec	A_rec	N_rec
OCDE_SEMS_A_COO_0	0.15	-0.14	-0.08	-0.04	-0.03
OCDE_SEMS_A_COO_1	-0.13	0.12	0.07	0.03	0.06
OCDE_SEMS_A_EMP_0	0.16	-0.14	-0.08	-0.04	-0.04
OCDE_SEMS_A_EMP_1	-0.13	0.12	0.06	0.03	0.05
OCDE_SEMS_A_TRU_0	0.16	-0.14	-0.08	-0.03	-0.03
OCDE_SEMS_A_TRU_1	-0.13	0.12	0.06	0.03	0.06
OCDE_SEMS_C_MOT_0	0.18	-0.16	-0.14	-0.10	-0.04
OCDE_SEMS_C_MOT_1	-0.22	0.19	0.07	0.08	0.06
OCDE_SEMS_C_PER_0	0.18	-0.16	-0.14	-0.10	-0.04
OCDE_SEMS_C_PER_1	-0.22	0.19	0.07	0.08	0.06
OCDE_SEMS_C_RES_0	0.18	-0.16	-0.14	-0.10	-0.04
OCDE_SEMS_C_RES_1	-0.22	0.19	0.07	0.08	0.06
OCDE_SEMS_C_SEL_0	0.19	-0.16	-0.14	-0.10	-0.04
OCDE_SEMS_C_SEL_1	-0.22	0.19	0.07	0.08	0.06
OCDE_SEMS_E_ASS_0	0.12	0.01	-0.18	-0.04	-0.06
OCDE_SEMS_E_ASS_1	-0.06	-0.04	0.14	0.04	0.04
OCDE_SEMS_E_ENE_0	0.12	0.01	-0.18	-0.04	-0.07
OCDE_SEMS_E_ENE_1	-0.06	-0.04	0.14	0.04	0.04
OCDE_SEMS_E_SOC_0	0.12	0.01	-0.18	-0.04	-0.06
OCDE_SEMS_E_SOC_1	-0.06	-0.04	0.14	0.04	0.04
OCDE_SEMS_N_EMO_0	0.07	-0.07	-0.11	-0.03	-0.04
OCDE_SEMS_N_EMO_1	0.06	0.02	0.01	0.04	0.01
OCDE_SEMS_N_OPT_0	0.07	-0.07	-0.11	-0.03	-0.04
OCDE_SEMS_N_OPT_1	0.06	0.02	0.01	0.04	0.01
OCDE_SEMS_N_STR_0	0.07	-0.07	-0.11	-0.03	-0.04
OCDE_SEMS_N_STR_1	0.06	0.02	0.01	0.04	0.01
OCDE_SEMS_O_CRE_0	-0.10	0.05	0.04	-0.01	0.02
OCDE_SEMS_O_CRE_1	0.22	-0.12	-0.10	0.01	-0.03
OCDE_SEMS_O_CUR_0	-0.11	0.06	0.04	-0.01	0.02
OCDE_SEMS_O_CUR_1	0.22	-0.12	-0.10	0.01	-0.03
OCDE_SEMS_O_TOL_0	-0.10	0.05	0.04	-0.01	0.02
OCDE_SEMS_O_TOL_1	0.22	-0.11	-0.10	0.01	-0.03

vars	O_rec	C_rec	E_rec	A_rec	N_rec
SENNA_A_Cmp_0	0.16	-0.14	-0.08	-0.03	-0.04
SENNA_A_Cmp_1	-0.13	0.12	0.06	0.03	0.05
SENNA_A_Mod_0	0.15	-0.14	-0.08	-0.03	-0.03
SENNA_A_Mod_1	-0.13	0.12	0.06	0.03	0.06
SENNA_A_Resp_0	0.15	-0.14	-0.07	-0.04	-0.03
SENNA_A_Resp_1	-0.13	0.12	0.06	0.03	0.05
SENNA_A_Tru_0	0.16	-0.14	-0.08	-0.04	-0.04
SENNA_A_Tru_1	-0.13	0.12	0.07	0.03	0.06
SENNA_C_Achv_0	0.18	-0.16	-0.13	-0.10	-0.04
SENNA_C_Achv_1	-0.22	0.19	0.07	0.08	0.06
SENNA_C_Conc_0	0.18	-0.16	-0.14	-0.10	-0.04
SENNA_C_Conc_1	-0.22	0.19	0.07	0.08	0.06
SENNA_C_Ord_0	0.18	-0.16	-0.14	-0.10	-0.04
SENNA_C_Ord_1	-0.22	0.19	0.07	0.08	0.06
SENNA_C_SD_0	0.19	-0.16	-0.14	-0.10	-0.04
SENNA_C_SD_1	-0.22	0.19	0.07	0.08	0.06
SENNA_C_SofR_0	0.19	-0.16	-0.14	-0.10	-0.04
SENNA_C_SofR_1	-0.22	0.19	0.07	0.08	0.06
SENNA_E_Act_0	0.12	0.01	-0.18	-0.04	-0.07
SENNA_E_Act_1	-0.06	-0.05	0.14	0.04	0.04
SENNA_E_Assr_0	0.12	0.01	-0.18	-0.04	-0.07
SENNA_E_Assr_1	-0.07	-0.04	0.15	0.04	0.04
SENNA_E_Soc_0	0.12	0.01	-0.18	-0.04	-0.07
SENNA_E_Soc_1	-0.06	-0.04	0.15	0.04	0.04
SENNA_N_LAngrVol_0	0.07	-0.07	-0.11	-0.03	-0.04
SENNA_N_LAngrVol_1	0.07	0.02	0.01	0.04	0.01
SENNA_N_LAnx_0	0.07	-0.07	-0.11	-0.03	-0.04
SENNA_N_LAnx_1	0.06	0.02	0.01	0.04	0.01
SENNA_N_LDep_0	0.07	-0.07	-0.11	-0.03	-0.04
SENNA_N_LDep_1	0.06	0.02	0.01	0.04	0.01
SENNA_O_Aes_0	-0.10	0.05	0.04	-0.01	0.01
SENNA_O_Aes_1	0.22	-0.12	-0.10	0.01	-0.03
SENNA_O_CrImg_0	-0.11	0.06	0.04	-0.01	0.02
SENNA_O_CrImg_1	0.22	-0.12	-0.10	0.01	-0.03
SENNA_O_IntCur_0	-0.10	0.05	0.04	-0.01	0.01
SENNA_O_IntCur_1	0.22	-0.12	-0.10	0.01	-0.03

vars	O_rec	C_rec	E_rec	A_rec	N_rec
BFI2_A_cmp_0	0.26	-0.23	0.00	-0.09	0.01
BFI2_A_cmp_1	-0.22	0.21	-0.07	0.08	0.04
BFI2_A_res_0	0.26	-0.23	0.00	-0.09	0.01
BFI2_A_res_1	-0.21	0.21	-0.07	0.08	0.04
BFI2_A_trst_0	0.26	-0.24	0.00	-0.09	0.01
BFI2_A_trst_1	-0.22	0.22	-0.07	0.08	0.04
BFI2_C_org_0	0.20	-0.21	-0.13	-0.12	-0.08
BFI2_C_org_1	-0.34	0.27	0.02	0.11	0.09
BFI2_C_prod_0	0.20	-0.21	-0.13	-0.12	-0.08
BFI2_C_prod_1	-0.34	0.27	0.02	0.11	0.10
BFI2_C_rsp_0	0.20	-0.21	-0.13	-0.12	-0.08
BFI2_C_rsp_1	-0.34	0.27	0.02	0.11	0.09
BFI2_E_ass_0	0.09	-0.10	-0.24	-0.20	-0.03
BFI2_E_ass_1	-0.01	0.05	0.25	0.20	-0.02
BFI2_E_enrg_0	0.09	-0.10	-0.25	-0.20	-0.04
BFI2_E_enrg_1	-0.01	0.04	0.25	0.20	-0.03
BFI2_E_soc_0	0.08	-0.10	-0.25	-0.20	-0.04
BFI2_E_soc_1	-0.01	0.04	0.25	0.20	-0.03
BFI2_N_anx_0	0.03	-0.08	-0.18	0.00	-0.07
BFI2_N_anx_1	0.18	-0.07	0.08	-0.02	0.12
BFI2_N_dep_0	0.04	-0.08	-0.18	0.00	-0.08
BFI2_N_dep_1	0.19	-0.07	0.08	-0.02	0.12
BFI2_N_vlti_0	0.03	-0.08	-0.18	0.00	-0.08
BFI2_N_vlti_1	0.19	-0.07	0.08	-0.02	0.12
BFI2_O_aes_0	-0.22	0.13	-0.02	-0.06	0.01
BFI2_O_aes_1	0.35	-0.18	-0.03	0.07	-0.08
BFI2_O_crea_0	-0.21	0.12	-0.02	-0.06	0.01
BFI2_O_crea_1	0.35	-0.18	-0.03	0.07	-0.08
BFI2_O_int_0	-0.21	0.12	-0.02	-0.06	0.01
BFI2_O_int_1	0.35	-0.18	-0.03	0.07	-0.08

vars	O_rec	C_rec	E_rec	A_rec	N_rec
OCDE_SEMS_A_COO_0	0.25	-0.23	0.00	-0.09	0.01
OCDE_SEMS_A_COO_1	-0.22	0.22	-0.07	0.08	0.04
OCDE_SEMS_A_EMP_0	0.27	-0.24	0.00	-0.09	0.01
OCDE_SEMS_A_EMP_1	-0.21	0.21	-0.07	0.08	0.04
OCDE_SEMS_A_TRU_0	0.26	-0.23	0.00	-0.09	0.01
OCDE_SEMS_A_TRU_1	-0.22	0.22	-0.07	0.08	0.04
OCDE_SEMS_C_MOT_0	0.20	-0.21	-0.13	-0.12	-0.08
OCDE_SEMS_C_MOT_1	-0.34	0.27	0.02	0.11	0.09
OCDE_SEMS_C_PER_0	0.20	-0.21	-0.13	-0.12	-0.08
OCDE_SEMS_C_PER_1	-0.34	0.27	0.02	0.11	0.09
OCDE_SEMS_C_RES_0	0.20	-0.21	-0.13	-0.12	-0.08
OCDE_SEMS_C_RES_1	-0.34	0.27	0.02	0.11	0.09
OCDE_SEMS_C_SEL_0	0.20	-0.21	-0.13	-0.12	-0.08
OCDE_SEMS_C_SEL_1	-0.34	0.27	0.02	0.11	0.09
OCDE_SEMS_E_ASS_0	0.09	-0.10	-0.25	-0.20	-0.03
OCDE_SEMS_E_ASS_1	-0.01	0.05	0.25	0.20	-0.02
OCDE_SEMS_E_ENE_0	0.09	-0.10	-0.25	-0.20	-0.04
OCDE_SEMS_E_ENE_1	-0.01	0.05	0.25	0.20	-0.02
OCDE_SEMS_E_SOC_0	0.09	-0.10	-0.25	-0.20	-0.03
OCDE_SEMS_E_SOC_1	-0.01	0.05	0.25	0.20	-0.03
OCDE_SEMS_N_EMO_0	0.03	-0.08	-0.18	0.00	-0.08
OCDE_SEMS_N_EMO_1	0.19	-0.07	0.08	-0.02	0.12
OCDE_SEMS_N_OPT_0	0.03	-0.08	-0.18	0.00	-0.08
OCDE_SEMS_N_OPT_1	0.18	-0.07	0.08	-0.02	0.12
OCDE_SEMS_N_STR_0	0.03	-0.08	-0.18	0.00	-0.08
OCDE_SEMS_N_STR_1	0.19	-0.07	0.08	-0.02	0.12
OCDE_SEMS_O_CRE_0	-0.20	0.12	-0.03	-0.07	0.01
OCDE_SEMS_O_CRE_1	0.35	-0.18	-0.03	0.07	-0.08
OCDE_SEMS_O_CUR_0	-0.21	0.12	-0.03	-0.06	0.01
OCDE_SEMS_O_CUR_1	0.35	-0.18	-0.03	0.07	-0.08
OCDE_SEMS_O_TOL_0	-0.20	0.12	-0.03	-0.07	0.00
OCDE_SEMS_O_TOL_1	0.35	-0.18	-0.03	0.06	-0.08

vars	O_rec	C_rec	E_rec	A_rec	N_rec
SENNA_A_Cmp_0	0.26	-0.23	0.00	-0.09	0.01
SENNA_A_Cmp_1	-0.21	0.21	-0.07	0.08	0.04
SENNA_A_Mod_0	0.26	-0.23	0.00	-0.09	0.01
SENNA_A_Mod_1	-0.22	0.22	-0.07	0.08	0.04
SENNA_A_Resp_0	0.25	-0.23	0.00	-0.09	0.01
SENNA_A_Resp_1	-0.21	0.21	-0.07	0.08	0.04
SENNA_A_Tru_0	0.26	-0.24	0.00	-0.09	0.01
SENNA_A_Tru_1	-0.22	0.22	-0.07	0.08	0.04
SENNA_C_Achv_0	0.20	-0.21	-0.13	-0.12	-0.08
SENNA_C_Achv_1	-0.34	0.27	0.02	0.11	0.09
SENNA_C_Conc_0	0.20	-0.21	-0.13	-0.12	-0.08
SENNA_C_Conc_1	-0.34	0.27	0.02	0.11	0.09
SENNA_C_Ord_0	0.20	-0.21	-0.13	-0.12	-0.08
SENNA_C_Ord_1	-0.34	0.27	0.02	0.11	0.09
SENNA_C_SD_0	0.20	-0.21	-0.13	-0.12	-0.08
SENNA_C_SD_1	-0.34	0.27	0.02	0.11	0.09
SENNA_C_SofR_0	0.20	-0.21	-0.13	-0.12	-0.08
SENNA_C_SofR_1	-0.34	0.27	0.02	0.11	0.09
SENNA_E_Act_0	0.09	-0.10	-0.24	-0.20	-0.04
SENNA_E_Act_1	-0.01	0.04	0.25	0.20	-0.03
SENNA_E_Assr_0	0.10	-0.11	-0.24	-0.20	-0.03
SENNA_E_Assr_1	-0.01	0.05	0.25	0.20	-0.02
SENNA_E_Soc_0	0.09	-0.10	-0.24	-0.20	-0.04
SENNA_E_Soc_1	-0.01	0.05	0.25	0.20	-0.02
SENNA_N_LAngrVol	0.04	-0.08	-0.18	0.00	-0.08
SENNA_N_LAngrVol	0.19	-0.08	0.07	-0.02	0.11
SENNA_N_LAnx_0	0.03	-0.08	-0.18	0.00	-0.08
SENNA_N_LAnx_1	0.19	-0.07	0.08	-0.02	0.12
SENNA_N_LDep_0	0.03	-0.08	-0.18	0.00	-0.08
SENNA_N_LDep_1	0.19	-0.07	0.08	-0.02	0.12
SENNA_O_Aes_0	-0.20	0.11	-0.03	-0.07	0.00
SENNA_O_Aes_1	0.35	-0.18	-0.03	0.07	-0.08
SENNA_O_CrImg_0	-0.21	0.12	-0.03	-0.06	0.01
SENNA_O_CrImg_1	0.35	-0.18	-0.03	0.07	-0.08
SENNA_O_IntCur_0	-0.20	0.11	-0.03	-0.07	0.00
SENNA_O_IntCur_1	0.35	-0.18	-0.03	0.07	-0.08

References

Hommel, B. E., Wollang, F.-J. M., Kotova, V., Zacher, H., & Schmukle, S. C. (2022). Transformer-based deep neural language modeling for construct-specific automatic item generation. *Psychometrika*, *87*, 749–772. https://doi.org/10.1007/s11336-021-09823-9

Mehta, Y., Majumder, N., Gelbukh, A., & Cambria, E. (2020). Recent Trends in Deep Learning Based Personality Detection. *Artificial Intelligence Review*, *53*(4), 2313–2339. https://doi.org/10.1007/s10462-019-09770-z

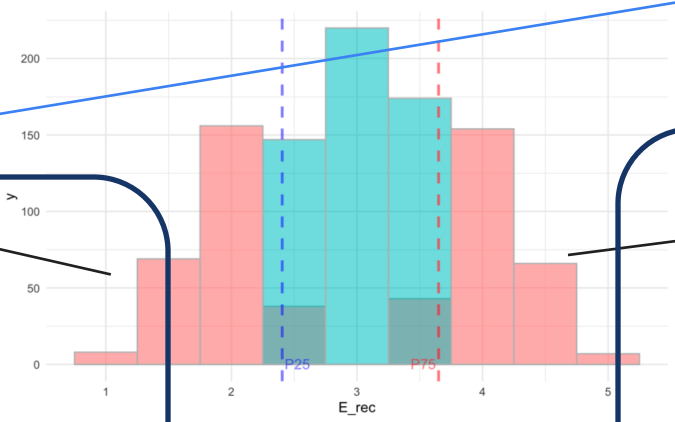
Plank, B., & Hovy, D. (2015). Personality Traits on Twitter—Or—How to Get 1,500 Personality Tests in a Week. *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 92–98. https://doi.org/10.18653/v1/W15-2913

Santos, W. R. dos, & Paraboni, I. (2019). *Personality facets recognition from text* (arXiv:1810.02980). arXiv. https://doi.org/10.48550/arXiv.1810.02980

Soto, C. J., & John, O. P. (2017). The next Big Five Inventory (BFI-2): Developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power. *Journal of Personality and Social Psychology*, *113*(1), 117–143. https://doi.org/10.1037/pspp0000096

Stajner, S., & Yenikent, S. (2020). A Survey of Automatic Personality Detection from Texts. *Proceedings of the 28th International Conference on Computational Linguistics*, 6284–6295. https://doi.org/10.18653/v1/2020.coling-main.553

Positive examples



Posts

I am reserved
I am shy

Posts

I am extroverted, sociable
I like to talk

Itens/Queries