

# Linear Dimensionality Reduction and Affect



GIANT OAK  
Timothy Ressler

August 5, 2022

# Word Embeddings as Features

- Words can be represented by semantic embeddings.

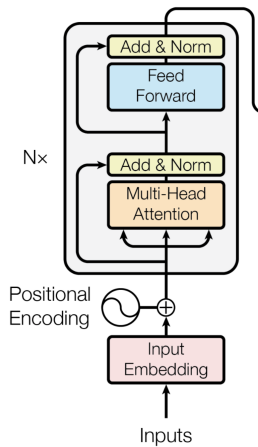
$$\bullet \left\{ \begin{array}{lcl} \textit{good} & : & \begin{bmatrix} -0.3 & 0.9 & 0.2 & -0.8 \end{bmatrix} \\ \textit{bad} & : & \begin{bmatrix} -0.4 & -0.8 & 0.1 & -0.8 \end{bmatrix} \\ \textit{terrorist} & : & \begin{bmatrix} 0.1 & -0.7 & 0.5 & 0.0 \end{bmatrix} \\ \textit{Tim} & : & \begin{bmatrix} 0.9 & 0.6 & -0.4 & 0.7 \end{bmatrix} \end{array} \right\}$$

- Words that appear in similar contexts have similar word embeddings, and a thus a higher cosine similarity.
  - This apple tastes *good*. This apple tastes *bad*.  $\text{CosSim}(\textit{good}, \textit{bad}) = 0.74$
  - This apple tastes *good*. This apple tastes *terrorist*.  $\text{CosSim}(\textit{good}, \textit{terrorist}) = 0.43$
- Embeddings can be created for words using different encoders: the most famous being Word2Vec, GloVE, ELMo, and BERT.
- These embeddings can be used as feature vectors for different downstream tasks.



# BERT

- BERT is unique in that it uses transformers in determining the embedding of a word. This allows BERT to use context to give more contextually accurate word embeddings and disambiguate between homonyms.
- The success of BERT as an encoder has led to many variants: we are using a XLM-RoBERTa (XLMR) longformer.
  - RoBERTa (Robustly Optimized BERT Approach) is similar to BERT, but trained longer, on more data, and with a modified learning objective.
  - XLM-RoBERTa is a multilingual version of RoBERTa.
  - A longformer takes more input tokens.



- Words have three different affect dimensions:
  - Sentiment (Valence): goodness/badness
  - Agency (Arousal): activeness/passiveness
  - Power (Dominance): strength/weakness
- This gives us the ability to more finely distinguish between synonyms.
- Hypothesis: Derog can be defined to be a combination of different affect dimensions.

Dimension	Word	Score↑	Word	Score↓
valence	<i>love</i>	1.000	<i>toxic</i>	0.008
	<i>happy</i>	1.000	<i>nightmare</i>	0.005
	<i>happily</i>	1.000	<i>shit</i>	0.000
arousal	<i>abduction</i>	0.990	<i>mellow</i>	0.069
	<i>exorcism</i>	0.980	<i>siesta</i>	0.046
	<i>homicide</i>	0.973	<i>napping</i>	0.046
dominance	<i>powerful</i>	0.991	<i>empty</i>	0.081
	<i>leadership</i>	0.983	<i>frail</i>	0.069
	<i>success</i>	0.981	<i>weak</i>	0.045

Table 2: The terms with the highest (↑) and lowest (↓) valence (V), arousal (A), and dominance (D) scores in the VAD Lexicon.



# Entity-Level Derog

- Architecture: input sentence  $\rightarrow$  **BERT**  $\rightarrow$  entity embedding(s)  $\rightarrow$  **decoder**  $\rightarrow$  affect scores
- Decoders: Field and Tsvetkov proposes two models for decoding:
  - Kernel Ridge Regression (KRR): similar to SVM, where a RBF kernel is used.
  - Affect Subspace Projection (ASP): uses extreme-valued words and linear decomposition to project scores.
- Why use ASP when KRR performs better?
  - Confounds are words that “trick” or “confuse” a model into learning incorrectly. For example, for agency, the KRR might learn to differentiate between animate and inanimate objects instead of high and low agency words.
  - ASP allows us to control for confounds, by selecting the most extreme-valued words, and “measuring” words of interest along the axis separating the two extremes.



- For each affect dimension:
  - From the lexicon, find the embeddings for the  $|\mathcal{H}|$  highest-valued words and  $|\mathcal{L}|$  lowest-valued words.
  - Use cosine similarity to find the  $N$  most similar pairs between  $\mathcal{H}$  and  $\mathcal{L}$ .
  - Subtract each embedding by the average of itself and its pair, and construct a matrix  $M$  out of all the embedding pairs.
  - Linearly decompose the matrix into a subspace of one dimension with the highest variance.
  - For each entity, project its embedding onto the subspace.



# Linear Decomposition

- Currently, ASP is using the first principle of principle component analysis (PCA) as its method of linear decomposition.
- There may be more suitable methods of linear decomposition than PCA, like UMAP.
- Objective: determine if UMAP or any other decomposition technique is better suited for ASP than PCA.

