SLANGuage Models

Jacob Matzke, Saif Kader, Pravar Annadanam

Abstract

Slang words are a subsect of language that reflect pop culture and current events of the time. They're known for their mercurial nature, often coming into and out of the popular vocabulary quickly. In our findings, we used BERT and Gensim FastText to fine-tune pre-trained embeddings of tweets over a time period to gain insight on a list of slang words. We found that certain categories of slang words were more likely to be found with a consistent set of context words, but as a whole, slang words were not embedded strongly into any vocabulary niche. We did find that slang words and their neighbors with high similarity could be used to reflect current events. Our code is linked here.

Introduction:

As young adults who have grown up in an increasingly online world, one major benefit we get is instantaneous communication. We can share ideas with more people than ever before in seconds, and it only takes days, if not hours, for concepts, phrases, and ideas to go viral and reach huge audiences. One such concept that spreads quickly is slang. We have seen slang evolve tremendously in the last decade, and words enter and leave pop culture before you know it. We wanted to use this project to gain some insight on what we often take for granted. We may be aware of how slang shifts in culture by participating in social media, but it's difficult to step back and understand why slang shifts, or sticks around. This project looked to gain insight into how slang shifts in popular use, in terms of what words are most similar to it at given points of time, and how unique or how niche of a role a word might occupy. To do so, we employed various machine learning techniques on a database of tweets, treating twitter as a reliable reflection of the slang used at that point in time.

Data:

For our data, we decided to stream data from several days in Twitter/X history, namely, days from 2018 to 2022 around the date of June 10th. We chose to have a delta of approximately 1 year between each data group because the general topics and events happening at any given point in a year will balance each other out. We found lots of weather data occupying our datasets and so we wanted to reduce the impact that yearly conversations or predictable topics given the climate and time of year have on our data. We brought in data, namely ~1.1 million tweets per day/year from the Twitter Stream Archive, to have a large pool of information that we could draw upon for linguistic analysis.

We chose Twitter for data because it is a place on the internet where people are largely free to communicate however they want, limited only by some community guidelines and a character limit. We thought that when discussing and analyzing slang, this was a place where we could freely examine how people communicate informally as well as where slang originates in large parts. Twitter has shaped how people speak, creating acronyms and words that have permeated into common speech in the world today, so using data from there would allow us raw, online communication data that we could analyze.

Furthermore, for our slang dataset, we compiled it using online resources like urban dictionary, a Generation Z dataset we found on GitHub, and of course, the vernacular and slang with which the members of our group are familiar. We therefore came up with a list of 40 slang words, and stratified them into 5 subcategories for analysis, those being Validity, Relationship, Pop Culture, Derogatory, and Compliments. We wanted to analyze not only the words we found, but how categories of words can be brought together and see if there were any larger patterns we could track.

Our slang lists looked as such:
Relationship: 'bae', 'bm', 'fam', 'homeboy', 'homie', 'squad', 'ghost'
Pop Culture: 'stan', 'binge', 'ship', 'ratio'
Excitement: 'fire', 'slaps', 'gas', 'cooked', 'cooking', 'lit', 'goat', 'w', 'hype', 'dope'

Derogatory: ‘clown’, ‘cringe’, ‘karen’, ‘extra’, ‘basic’, ‘yikes’, ‘haters’, ‘l’, ‘trash’, ‘salty’
Validity: ‘frfr’, ‘ngl’, ‘cap’, ‘deadass’, ‘lowkey’, ‘facts’, ‘tbh’, ‘bet’, ‘legit’

Finally, the last group of data that we used were 2 types of embedding models. Our first approach was using BERT Uncased Contextual Embeddings followed by Static GenSim FastText Embeddings, both of which we attempted to train on various different strata of our Twitter data.

Methodology

Data cleaning was a process we did throughout our approach. After we streamed the data into a dataset, we then took all the ~6 million tweets and separated them into their respective years that they were tweeted. We also cleaned out the retweets, usernames, and tokens/items that didn’t make sense to have, such as links and images.

For our approach, we used four separate methodologies to get the best slang analysis. We first tried using BERT Uncased Contextual Embeddings on the entire tweet dataset. When we ran cosine similarity for a slang word like ‘lit’; however, the resulting words were most similar in structure and length to lit, not a semantic similarity. We abandoned this approach because of this failure, as our dataset wasn’t large or verbose enough to leverage what these embeddings had to offer.

Our second attempt was using Static Embeddings from GenSim’s FastText. An issue that we saw with the BERT model was that it was getting bogged down by usernames and retweets, so we cleaned our tweet dataset to exclude retweets, as most of them were not substantive in length or verbiage, often containing usernames and irrelevant words to the overall Twitter lexicon. When we trained these static embeddings on our no-retweets dataset, we saw a drastic improvement in the quality of retrieval and these embeddings as words were matching with each other based on context and not visual similarities. Once we were clear of these ‘lazy matches’ and spelling attributions of similarity, we moved onto an improved version of this solution.

This third attempt was continuing to use the static GenSim embeddings, but with FastText’s 300 Dimensional English Vectors as pretrained embeddings on intersecting vocabulary. This methodology helped our output and similarity grow further. When we queried this system using a most_similar words approach, there were more slang words in the output, which was something we expected to see as an indication of improvement. In this solution, we were still training our data on the no-retweets dataset.

When we finally introduced all tweets back into the training, including retweets cleaned of usernames and the ‘rt’ token, we saw our largest improvement and stuck with it. This final solution used static GenSim embeddings with the FastText 300 Dimensional English Vectors and utilized GenSim’s most_similar function again so we could use cosine similarity for the most similar n words in a given year and filtered these amongst the slang dataset that we created.

Results

Our system is not intended to have a metric of performance to measure it by. The goal of this project was to consolidate large sets of language to get a better understanding of the movement of words over time. The first track we looked at was an overview of slang by category. There may be some room for error in the categories themselves, as slang and its ever changing nature make it difficult to put under one label, especially for a time span of five years.

Category	Avera ge	Average Cosine	Average Number of

	Cosine Similarity	Similarity Delta Year to Year	Persistent* Neighbors Per Word
Validity	0.3148	0.0035	5.667
Derogatory	0.3175	0.0028	4.500
Relationship	0.3156	0.0040	3.1250
Pop Culture	0.3211	0.0004	6.5000
Excitement	0.3252	0.0016	6.4000

The first category we looked at was average cosine similarity. From the embeddings of the words per year, we were able to create a list of the 30 nearest neighbors of a given slang word for that year. Each neighbor has a cosine similarity score to the slang word. We wanted to average the cosine similarity of the 30 neighbors for each slang word, and then average those averages across a category. We were looking to determine how 'embedded' or niche a certain slang word might be. A high cosine similarity score may indicate that the 30 neighbors are strongly coupled with the given slang word, or that when you see the slang word, you're very likely to see the neighbors as well. This might indicate that a slang word is found in a very niche environment. The slang word may not be very versatile; it might only see use in a certain context. However, a lower cosine similarity indicates the opposite. There's no dedicated context words that are strongly associated with the slang, and thus it can be used anywhere. As the results show, all of the slang categories had relatively low similarities of around 0.3000. This reinforced our primary assumptions. Slang, in order to become popular, has to be accessible. It has to be able to be widely used, in multiple contexts, to let it skyrocket to virality and become part of everyday vocabulary. This inherently seems to require versatility, and the cosine similarities back up this belief.

The next category we wanted to analyze was the average cosine similarity delta from year to year. For each slang word, when we found the average cosine similarity of that word from its neighbors for the year, we wanted to analyze if there was a change in average for the year between years. This could indicate a shift in the word's usage, whether it be in popularity or the meaning of the word. These numbers saw relatively more difference between the categories than just average cosine similarity, but we believe there are some caveats as to why. Below are two examples of the changes in neighbors for the words ship and ratio. In terms of slang, ship is used commonly as shorthand for relationship, and would be expected to see neighbors relating as such. However, the word relationship only shows up once in ship's neighbors, and only in 2018. In fact, there are very few neighbors that seem to match our assumption; 'fanfiction' in 2020 at 0.3521, and 'fandom' in 2022. 2020 saw a rise of relationship related words at the low end of its neighbors, but the words above it, and taking control of most of the neighbors for all of the other years, were actually shipping (naval) related. We also see that in 2022, there's a dramatic shift from words like 'sailing' and 'shipping' in previous years to 'harpoon', 'navy', 'warship', and other more militaristic words. This reflects the news of the time with Ukraine and other global conflicts. This is a pattern we'll see more for other words, and we'll revisit it soon.

See Figure 1 and 2

Note that in these PCA graphs, red represents 2018, green is 2020, blue is 2021, and yellow is 2022.

The final statistic we looked at was the average number of persistent neighbors per word. Although the neighboring words for each slang word were likely to change significantly because of the nature of the dataset (sampling from one day on Twitter gave us plenty of data, but it was often biased because of virality), we found that certain words did still stay in the neighboring

set. We wanted to see, as another measure of the first statistic, how strongly coupled a slang word was with its neighbors. A neighbor was considered persistent if it showed up in at least 3 of the 5 years as a neighbor. We averaged the number of neighbors for slang words in the category, and as you can see, pop culture and excitement had the most repetition of neighbors. This is another indicator of how niche or embedded a category, and this made sense for pop culture. Pop culture tended to stay around the sphere of 'Stan Twitter', where people mostly tweeted about celebrities they liked following. This is typically a secluded, dedicated side of Twitter, so it made sense that it would be strongly embedded. Relationship slang being low also made sense; anyone can refer to pretty much anyone with these words, and what words they use in context could be so varied (positive or negative) that it's expected to see the category less embedded.

We graphed each word using PCA dimensionality reduction, showing the embeddings over the timespan, as seen above. This was an easy way to visualize the changing typical context of the words, and we noticed an interesting phenomenon in the excitement words.For validity words, like 'deadass', we saw that the words tended to be related to other slang like as below, which was what we were expecting when starting this project.

See Figure 3

What we noticed, however, was that the excitement words didn't contain other slang words; instead, they reflected more on the current events of the time. This opened our eyes to an alternative use of these graphs. For example, the graph for 'gas' showed a relatively unorganized set of neighbors for 2018. However, in 2020 and 2021, when we were experiencing heavy protests, the words became related to TEAR gas; you see riots, armor, bullets, etc. Then, in 2022, when gas prices skyrocketed, we see that the words become related to CAR gas; barrel, gouging, and inflation. You can see a similar pattern with 'goat', where 2018 has lebron, bryant, and nba, relating to the NBA finals of the time, while 2020 with the World Cup sees messi and neymar. We thought this was an incredibly interesting use case for these, and felt that it established again how heavily involved slang is with pop culture, and how it is the tool we use to reflect on and discuss current events.

See Figure 4 and 5

When trying a different dimensionality reduction, specifically t-SNE, we got the results seen in figure 6.

Another observation we were able to make was the overall movement and semantic drift of individual slang words. For example, the images below of 'deadass' and 'ghost' demonstrate clear semantic drift of slang. One pattern that we noticed that emerged was that of predicting low vs high semantic drift. With words that are aligned to certain meanings before being slang, they have higher semantic drift like ghost, but with words that start as slang, as they become used more, they tend to follow a pattern of finding a niche instead of just jumping around, having lower semantic drift and more of a linear/predictable pattern.

See Figures 7 and 8

What We Did

All three of us were working on the project together as we stated in our initial proposal. However, each of us took the lead on certain sections. Jacob led on data processing and the BERT model development. Saif led on Gensim and embeddings. Pravar led on visualization and dimensionality reduction. All three of us worked on data analysis equally to put together this final product.

References:

Figure 1

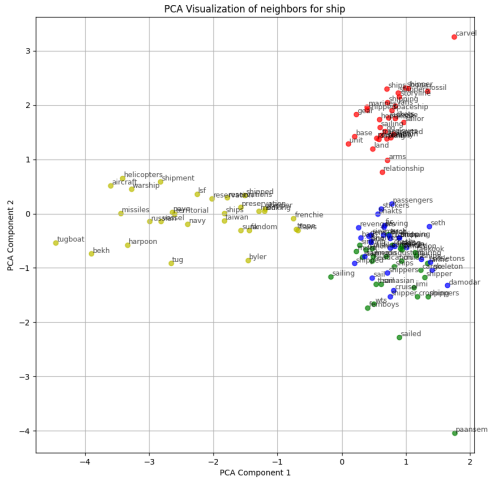


Figure 2

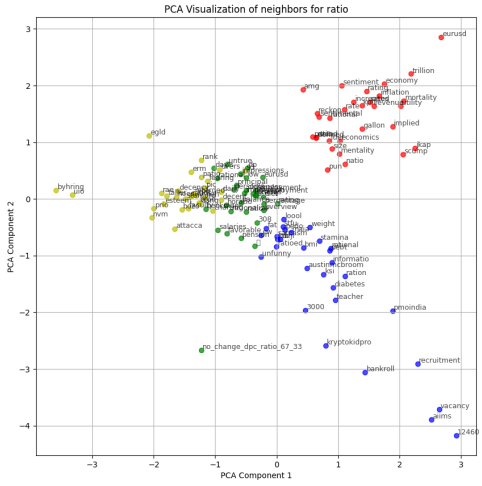


Figure 3

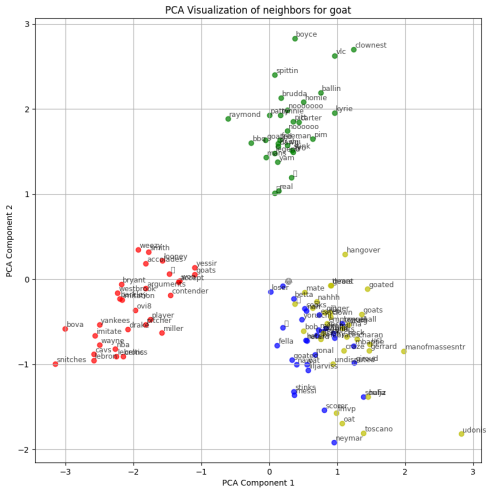


Figure 4

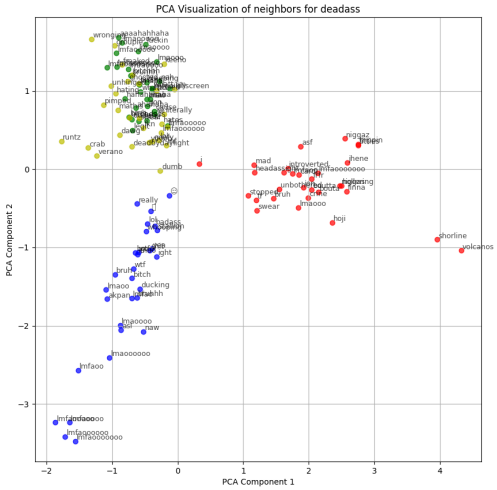


Figure 5

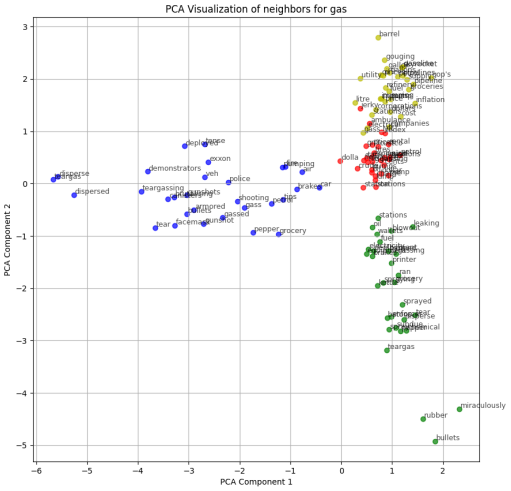


Figure 6

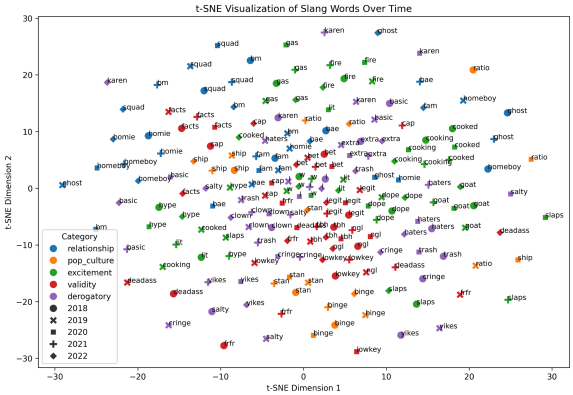


Figure 7

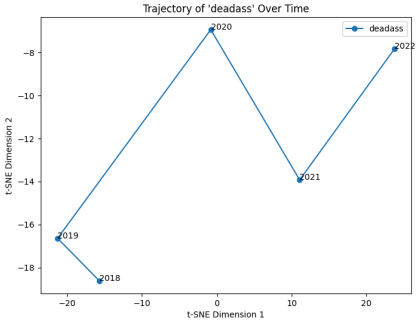
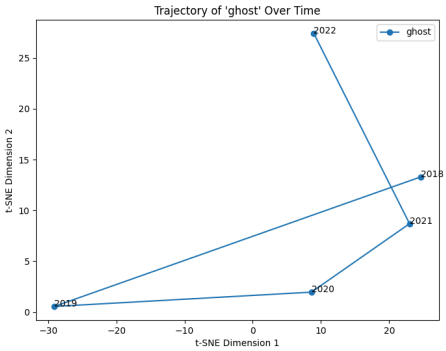


Figure 8



Python Module references:

https://github.com/matplotlib/matplotlib
https://github.com/numpy/numpy
https://github.com/piskvorky/gensim
https://github.com/NVIDIA/DeepLearningExamples/blob/master/PyTorch/LanguageModeling/BERT/README.md
https://github.com/googlecolab
https://github.com/scipy/scipy
https://github.com/scikit-learn/scikit-learn
https://github.com/tqdm/tqdm
https://github.com/pandas-dev/pandas
https://github.com/nltk/nltk

Data:

https://archive.org/details/twitterstream
https://radimrehurek.com/gensim/models/fasttext.html