Probing the Dynamics of Language
Using Word-Embedding and Text-Generation Models

Final Writeup & Reflection

Team: Gopal Iyer, Roman Hall, Zaul Tavangar

Github: https://github.com/zaultavangar/DL-final-project

Introduction

In this project, we probe the evolution of ideas as captured by the embeddings and prompt completions (for fixed prompts) of word-embedding and text-generation (language) models trained on time-sliced data. Our goals are twofold. First, we use independently trained and rotationally aligned word-embedding models on time-sliced text data to determine the variation in the cosine similarity between pairs of ideas, for example, 'Korea' and 'war.' This portion of our project is inspired by Haoxiang Zhang's master's thesis, *Dynamic Word Embedding for News Analysis*. We then use independently trained time-sliced text-generation models to study the shifting 'contexts' of particular prompts as captured by the models. This is a problem of interdisciplinary relevance that utilizes fairly simple ideas in machine and deep learning—word-embedding models and transformers.

Methodology

Dataset

We begin with a Kaggle dataset containing 1.2 million Australian newspaper headlines. These headlines are organized by date. We divide this dataset into 19 sub-datasets, each corresponding to one year of headlines, ranging from 2003 to 2021.

Embedding Models

First, we determine the optimal hyperparameters of the word-embedding model, Word2Vec, by scanning them on the full dataset. Using these hyperparameters, we then train 19 different word-embedding models, each on one year of headlines. This results in 19 different embedding models.

As an initial analysis, we use the t-SNE nonlinear dimensionality reduction algorithm to visualize embeddings for each of the models. These visualizations clearly indicate the shifting semantics of the embeddings, and by extension, of the news itself. As expected, we observe qualitative agreement between the nearest neighbors of a word using both the t-SNE visual and cosine similarity.

Since the latent semantics of these models are not all the same, it becomes necessary to rotationally align them in order to maximize the cosine similarity between words in the shared vocabularies of all 19 sub-datasets. This enables us to make principled comparisons between the embeddings of a word in different models. To carry out this rotational alignment, we employ the orthogonal Procrustes algorithm, which finds a rotation corresponding to each year's embedding that maximally aligns it with some reference embedding:

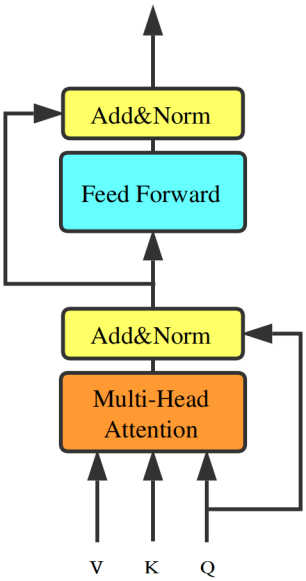
$$\mathbf{R}^{(t)} = \arg\min_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}} \|\mathbf{W}^{(t)}\mathbf{Q} - \mathbf{W}^{(t+1)}\|$$

Once these rotations have been obtained, it becomes possible to identify words whose meaning—i.e. position in embedding space—changes a lot, vs. words whose meaning changes very little. This is done by summing the cosine similarities of a word in time-adjacent embedding models (t, t+1) over the entire span of the dataset, i.e. 2003–2021. We call this metric the 'self-similarity' of the word.

Using these trained and rotationally aligned word-embedding models, we probe the cosine similarity of semantically interesting pairs of words and its evolution across time. This reveals interesting correlations between occurrences in the news cycle and corresponding shifts in the similarity or difference between words.

Language Models

Using the similarly partitioned dataset, we train 19 different transformer models for text generation. The transformer architecture is fairly simple for ease of repeated training and uses two attention heads. Using sparse categorical cross-entropy as the loss function, we find that the loss is approximately the same across these models for the same training hyperparameters. This suggests that the models do not overfit specific parts of the data distribution and serves as a good heuristic for validating the models. For each of these models, we use the fixed pre-trained embedding models obtained in the *Embedding Models* analysis.



Our next goal is to use these language models to probe the shifting contexts surrounding specific prompts. We measure this in the following way. Given a prompt, for example, 'the police are,' we allow each transformer model to generate up to 20 words to complete the prompt. From this completion, we select the first descriptive noun, adjective or verb that

appears in the completion. We take this completing word (noun, adjective, or verb) to be the 'context' of the prompt. For each completing word, we then measure its cosine similarity—in the embedding space corresponding to the year of the model—with some 'invariant' word. The invariant word is intended to serve as a reference frame of the embedding spaces and is chosen to be the word in the common vocabulary of all the models that has the highest self-similarity defined above. Finally, we plot the cosine similarity of the completing word (which varies by year) with the invariant word (which is fixed across years) to quantify the shifting context of the prompt across time.

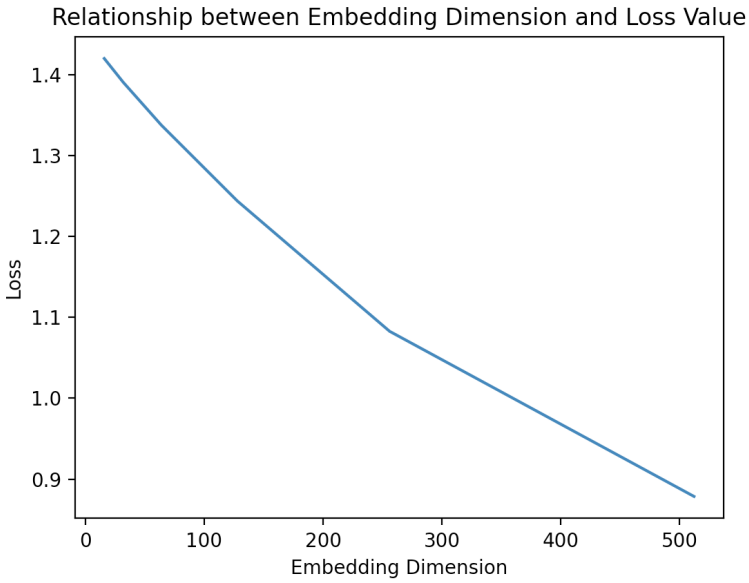
Experiments with LLMs

As an epilogue to our analysis, we feed the prompt used to probe the language models to ChatGPT. We first 'fine-tune' ChatGPT by feeding it 100 randomly chosen headlines from each year and then ask it to most meaningfully complete the prompt given the information contained in the headlines. We then make qualitative comparisons between the outputs of our language models and that of ChatGPT.

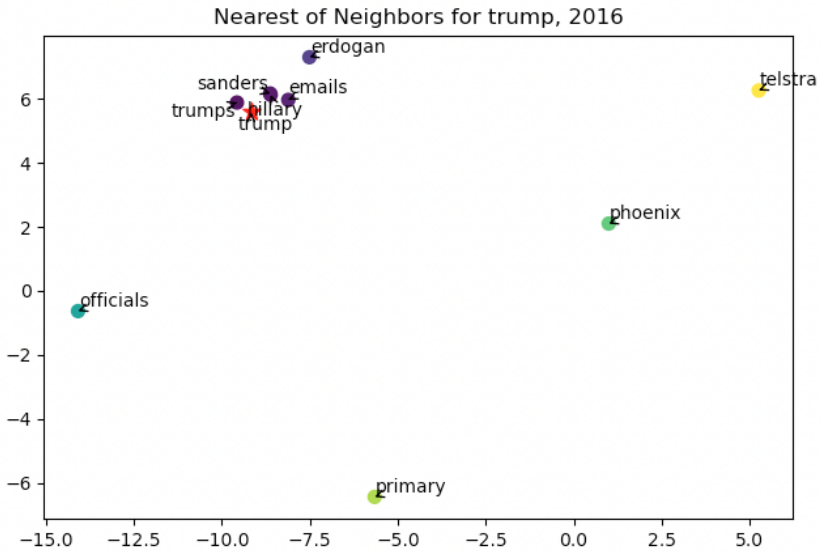
Results

Embedding Models

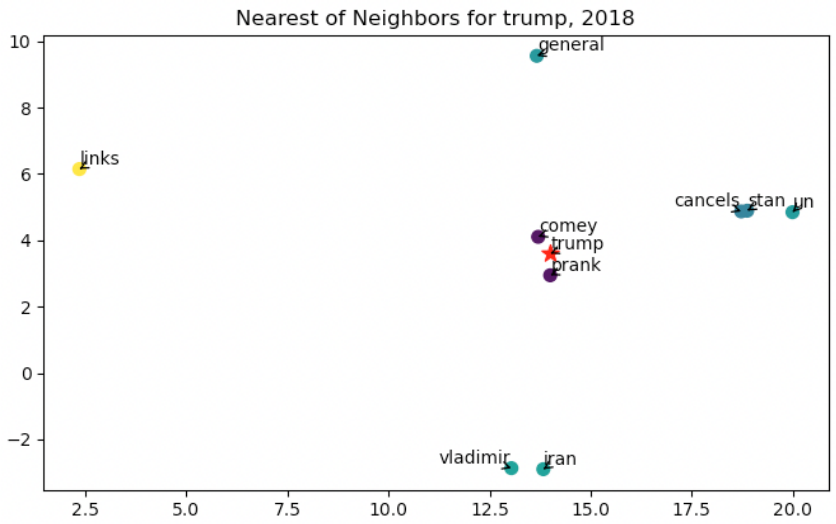
Our tests of the hyperparameters of the embedding models reveal that the error goes down monotonically on increasing the embedding dimension (see the relationship graphed below). Increasing the embedding dimension also increases the computational cost of training and inference. In order to balance accuracy and efficiency, we choose an embedding dimension of 256. We find that this is the only significant hyperparameter that affects model performance.



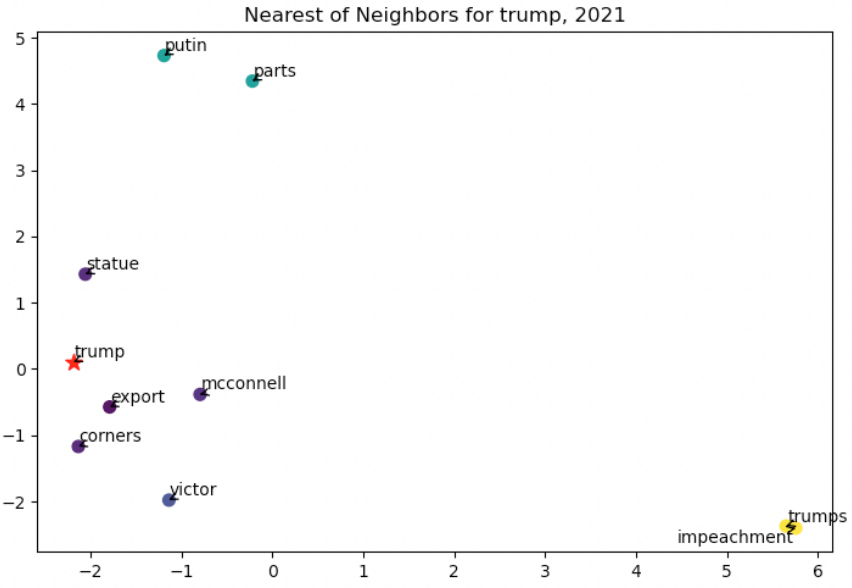
As mentioned above, the t-SNE visualization reveals the shifting proximity of different words to a given query word for different embedding models trained for different years. As an example to demonstrate the contextual/semantic shift over the years, here are some interesting plots of the nearest embedding neighbors of “Trump” in 2016, 2018, and 2021:



In 2016, we see words like “hillary,” “sanders,” and “primary” as neighbors of “trump,” which makes sense given that this was the election year. We also see “emails” as a close neighbor, likely due to the email scandal surrounding Hillary Clinton which arguably helped propel Trump to being elected as President.



In 2018, we see that the 2016 election opponents have long since shifted away from “trump,” and now “comey,” “vladimir,” and “iran” appear as neighbors reflecting current events/actions taken in office (think of Trump pulling out of the Iran nuclear deal in 2018, for instance).

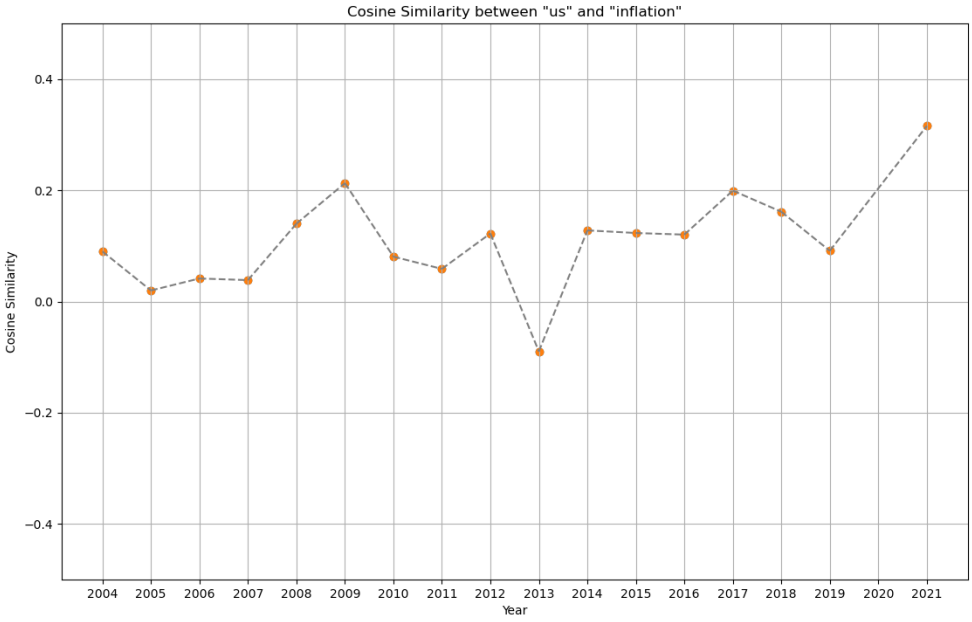


In 2021, the first year after Trump lost the 2020 election, we see “mcconnell” appear as a neighbor as well as “victor,” perhaps reflecting Trump’s refusal to acknowledge his defeat. Additionally, we see “impeachment” starting to creep in as the prosecution of the former President was set in motion and this shaped his public image.

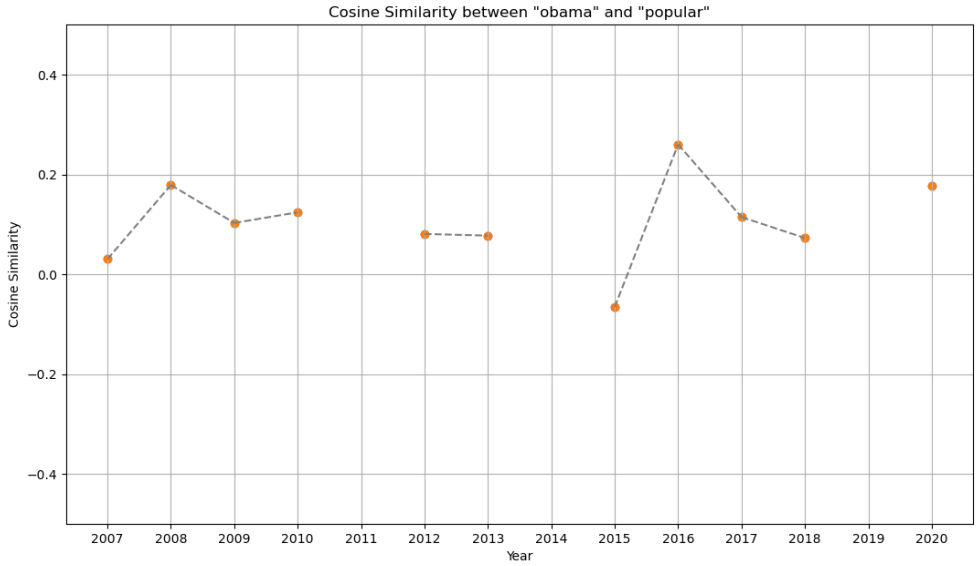
Using the orthogonal Procrustes method, the embeddings are aligned in order for the embeddings of specific words in different year-wise embedding models to be mutually comparable. The word with the lowest cosine self-similarity, i.e. highest shift in meaning across time, is ‘canberra.’ This makes sense intuitively because one would expect that the capital of Australia experiences the greatest shift in its association with other words across a dataset of Australian news headlines.

Given the year-wise trained and aligned embedding models, we now present plots showing the evolving cosine similarity between pairs of words across time. These plots reveal interesting correlations in the associations between words as the news changes over time.

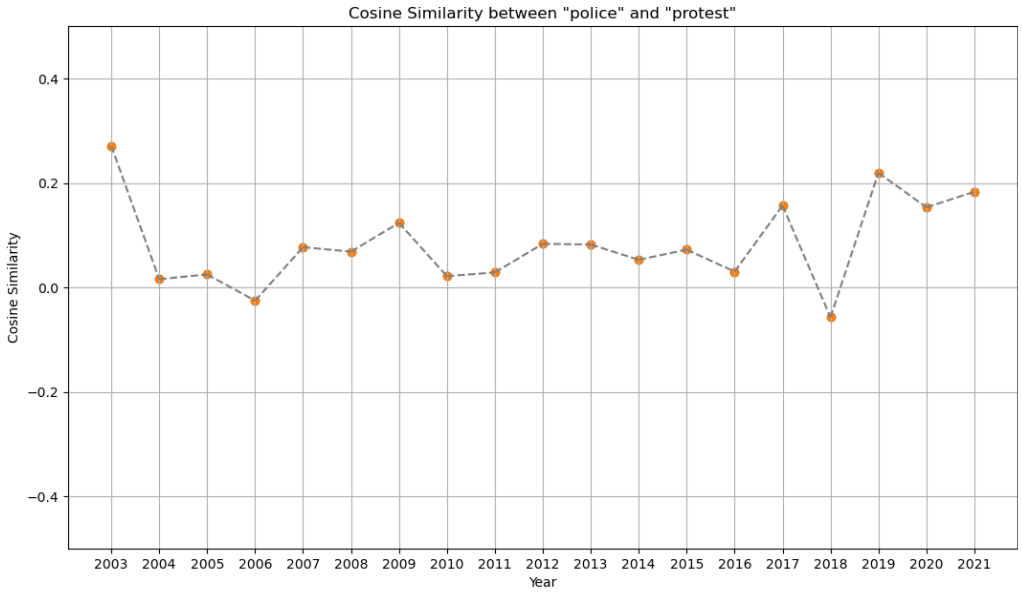
Here we plot the cosine similarity of “US” and “inflation.” Note the shifts in the similarity and the significant jump between 2019 and 2021, when the country passed a multi-trillion dollar stimulus package during the pandemic:



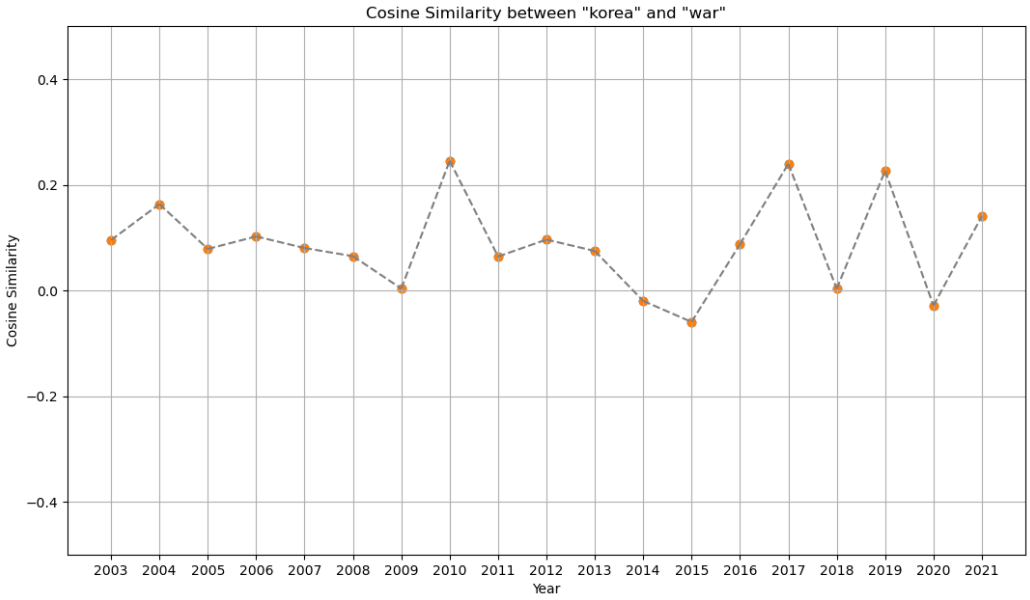
Here we plot the cosine similarity of “Obama” and “popular.” Note the uptick in 2016, the year when Trump was elected:



Here we plot the cosine similarity of “police” and “protest.” The relationship is somewhat steady from 2004 - 2016, but then we see a jump around 2019/2020, which brings to mind the murder of George Floyd in May 2020 and other acts of racial injustice which spurred on the Black Lives Matter movement / anti-police protests that year.



As a final example, we plot the cosine similarity of “Korea” and “war” and notice upticks in years when North Korea made stronger threats of war/missiles, reflecting an increased potential for war with North Korea in the public eye:



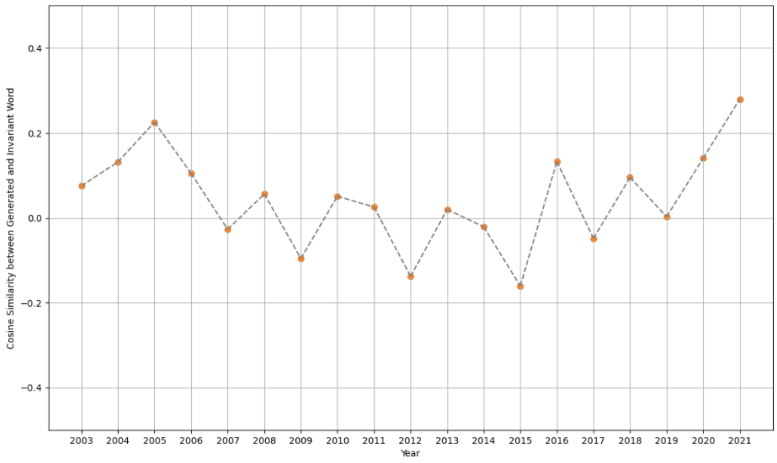
Language Models

The results from our language models were mixed in quality. Given the relatively small size of our training data, we were not expecting our language models to perform anywhere near as well as large language models such as GPT-3 (in terms of outputting grammatically/semantically correct sentences). We were more interested in seeing how well the predictions of these models could give us insight into evolving meanings/relationships among words in the headline datasets, which we tried to do by comparing descriptive words predicted as the completions of a given prompt in the models corresponding to different years. After we conducted the experiments we described above – plotting the cosine similarity of the completing word (which varies by year) with the invariant word (which is fixed across years) to quantify the shifting context of the prompt across time – we noticed some interesting trends in our results which quantitatively demonstrate shifts in latent semantics, as we set out to do. Here are two such graphs from different iterations of the same experiment:

First, we conducted this experiment with the starting prompt “the police are”:

Descriptive generated words by year:

- 2003: “release”
- 2004: “new”
- 2005: “assault”
- 2006: “court”
- 2007: “jet”
- 2008: “woman”
- 2009: “good”
- 2010: “school”
- 2011: “new”
- 2012: “cut”
- 2013: “helping”
- 2014: “taking”
- 2015: “ongoing”
- 2016: “behind”
- 2017: “ashes”
- 2018: “man”
- 2019: “trying”
- 2020: “celebrating”
- 2021: “trump”



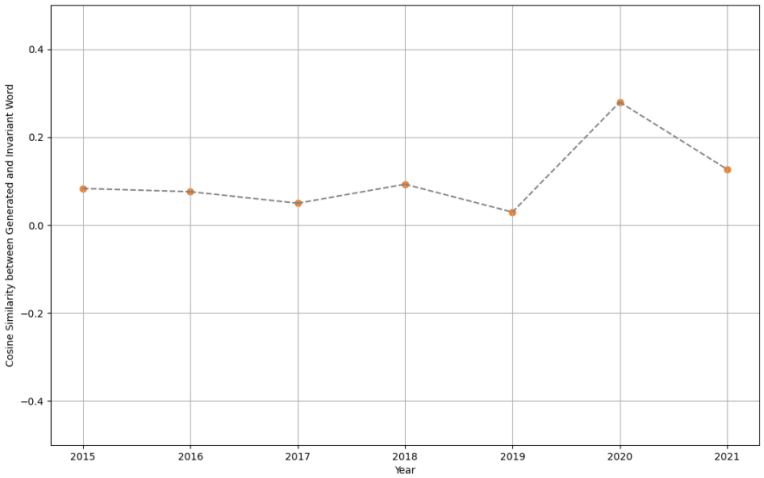
We see some movement in the graph over the years, and notice the highest cosine similarity of the context word and the invariant word in 2021, when the language model predicted “trump” as the first meaningful word relating to “police.”

Another interesting iteration of this experiment was with “Trump” as the starting prompt:

Same experiment but this time with the prompt “Trump”

Descriptive generated words by year:

- 2015: “forced”
- 2016: “warns”
- 2017: “threatens”
- 2018: “rejects”
- 2019: “meets”
- 2020: “tweets”
- 2021: “minister”



Note that “Trump” only started appearing frequently in the headlines in 2015. We see that the graph here remained fairly steady from 2015 - 2019, but then in 2020 there was a spike when the predicted related/context word was “tweets,” which was a departure from the more aggressive action verbs of the preceding years (“forced,” “warns,” “threatens,” “rejects,” and “meets”). One could interpret these results as highlighting the shifting platform/role of the President during the onset of the pandemic, when even world leaders were reduced to their

online presence/voice (we all know Trump was especially vocal/rampant on Twitter during this period).

As a further qualitative comparison of our models with popular large-language models, we fed 100 randomly selected headlines from the years 2003, 2009, 2015, and 2021 to ChatGPT and asked it to complete the prompt "the police are," given the headlines for each of those years. Understandably, the prompt completions of ChatGPT were a lot more coherent than our model. We reproduce its responses below:

- 2003: "the police are investigating multiple incidents, including attempted child abductions, possible gangland murder links, and mining vandalism, while also working to improve training and addressing safety concerns"
- 2009: "the police are **investigating** various incidents such as murder cases, missing divers, a restaurant attack, and suspecting a murder in the case of a **missing** man. They are also dealing with concerns over vacancies and working to improve officer **safety** with new bulletproof vests."
- 2015: "the police are investigating incidents such as a man shot in the leg in Ascot Vale, searching for a missing person, seeking information about a stabbing in Fairfield, and dealing with concerns related to public safety and bans."
- 2021: "The police are investigating a series of burglaries in the neighborhood"

Certainly, it is unreasonable to expect that there would be significant similarity of content between our year-wise language model outputs and those of ChatGPT, primarily because ChatGPT is only given access to 100 randomly selected headlines per year, whereas our model is given access to ~70000 headlines per year. However, it is interesting to observe the appearance of certain common themes between the two outputs. As an illustration of this, consider the completion of the same prompt in our 2009 model:

- 2009 model: "the police are good more for and **safe** being new peter after the now to on burning a over **missing** in **investigating** [UNK]"

Despite being incoherent, we see that the words 'safe,' 'missing,' and 'investigating' all appear in the prompt completions of both models, as highlighted in brown in the passages above. Not only does this illustrate that our language model is learning the right information, it also provides us with a useful qualitative roadmap for improving the capabilities of small language models. This illustrates a synergy between the development of small language models and fine-tuning LLMs that can be exploited to improve the performance of smaller models.

Challenges

The main obstacle we encountered during our work was in implementing the orthogonal Procrustes method. We ran into some programming difficulties trying to properly and efficiently rotate and align our embeddings so that we could compare relationships between words across time periods, but we ultimately succeeded in doing so. The rest of the challenges we faced, including time constraints, processing capabilities, and our relatively small and imperfect Australian-biased news headline dataset, had less to do with our actual implementation and are discussed in further detail in the reflection below.

Reflection

Ultimately, we are satisfied with the way our project turned out. With regards to our embedding models, our final word embeddings seem to capture meaningful relationships between the words in a given year's news headlines. We were able to successfully analyze these relationships using the metric of cosine similarity, as well as visualize them by employing the t-SNE method of dimensionality reduction. Our t-SNE plots give interesting insight regarding the qualitative and quantitative changes of a word's closest neighbors within the embedding space over time, and our cosine similarity evolution plots provide telling information about the evolution of the relationship between a given pair of words. With more time and processing capabilities, we would have been able to train our embedding models for more epochs (say, 20) with a higher embedding size (say, 512 – double the size we used), likely leading to even more accurate results.

As for our language model, our initial goal was to build a simple language prediction model that might be able to give us more insight, and certainly a different perspective, into the evolution of ideas over time. Given the relatively small size of the dataset overall and the small but nontrivial percentage of the headlines that seemed to be unfinished sentences, we were not expecting to see semantically and grammatically sound results, but were rather more interested in seeing if we could extract something meaningful by looking at the general context surrounding a word of interest. With this perspective, we managed to achieve some success—we noticed the context surrounding certain words changed in line with expectations, for example with the word "Trump", as discussed in our results section. Nevertheless, we acknowledge that our model's output sentences failed to convey much logical meaning, which forced us to think more creatively about ways to interpret our models' predictions and how we could glean meaning/sentiment from them.

In retrospect, thinking past our embedding models and venturing into the realm of language modeling, it would have been more prudent to:

1) employ a larger dataset—1.2 million news headlines is not quite sufficient to guarantee semantically coherent outputs from a language model
2) consider replacing our Kaggle dataset with news headlines sourced from more globally reputable outlets (free from any geographical/national bias), utilizing web scraping methods.

In conclusion, given more time and a larger, higher-quality dataset, we are confident that both our embedding and language models, particularly the latter, have the potential to produce even more insightful outputs, portraying a clearer and more comprehensive representation of the evolution of ideas over time. Despite this room for improvement, we feel our project provides valuable insights into the evolution of ideas, demonstrating the power and potential of word-embedding and language modeling for analyzing semantic shifts over time that could be applied in numerous applications such as social sciences, digital humanities, and any study of recorded human thought, more generally.

Sources:

Haoxiang Zhang, *Dynamic Word Embedding for News Analysis*
https://escholarship.org/uc/item/9tp9g31f

Apoorv Nandan, “Text generation with a miniature GPT”
https://keras.io/examples/generative/text_generation_with_miniature_gpt/

Transformer architecture diagram:
https://www.researchgate.net/figure/The-structure-of-a-Transformer-Block_fig1_336224014

Word2Vec TensorFlow documentation:
https://www.tensorflow.org/tutorials/text/word2vec

William L. Hamilton, Jure Leskovec, Dan Jurafsky, *Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change*, https://arxiv.org/pdf/1605.09096.pdf