University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Artificial Intelligence and Robotics with Advanced Research

7COM1039 MSc Masters Project

December 2, 2022

Can you tell me how to get to senitient street? (UMAP it out) :

Explainable Language Models as a Path to AGI

Name : Gemeny Givens

Student ID : 19022296

Supervisor: Iain Werry

Abstract

This study explores the possibility that explainability in natural language processing might guide us to AGI, or human level AI, via the improvement of knowledge representation. It has been suggested that human-level AI requires human-level knowledge representation and a natural language of thought. By diving into the high-dimensional word embeddings created by today's statistical language models, the experiments in this paper aim to find insights about what language models know, and how we might help them understand even more. A description is given for a model agnostic, post-hoc explainability framework for language models. The framework is made up of a principal component analysis (PCA), a similarity score tool, and a relatively new technique called a uniform manifold approximation and projection (UMAP). It was found that word embeddings from statistical models can encode a great deal of nuanced information, however they possess some counterintuitive qualities as well which could be improved. The UMAP visualisation is found to be the most intuitive and insightful tool for exploring what information is represented in the word vectors. However, some valuable insights are gained through cross-tool analysis as well. To conclude, the findings in this study help illustrate the need for ways to improve word representation in vector form, and an intelligible multimodal embedding is proposed as a potential path toward more robust AI.

Key words*: natural language processing, explainable AI, human level AI, AGI, knowledge representation, language models, word embeddings, principal component analysis, similarity score, UMAP*

Acknowledgements

I want to take a moment to acknowledge and thank God, my family and loved ones, everyone who helped make this research possible, and Oakland, California.

MSc Final Project Declaration

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Artificial Intelligence and Robotics with Advanced Research at the University of Hertfordshire (UH). It is my own work except where indicated in the report. I did not use human participants in my MSc Project. I hereby give permission for the report to be made available on the university website provided the source is acknowledged.

Contents

1 Introduction

Artificial general intelligence (AGI), or perhaps at least human level AI, has long been considered the holy grail of computer science and machine learning research. The idea behind AGI research is to develop a machine that can learn, think, communicate, and interact with a complex and dynamic environment like humans do (James, 2022).

In recent years, language has been deemed a potentially viable path to achieving AGI due to the performance of large language models (LLMs) on a variety of tasks such as sentiment analysis, question answering, and perhaps most impressively human language generation (Marzban and Crick, 2021).

While these LLMs are remarkable in their abilities on narrow tasks, scaling up model size alone has not proved sufficient for achieving high performance on tasks such as arithmetic, commonsense, and symbolic reasoning. On top of that, with billions of parameters and numerous layers, these large models are computationally expensive and very difficult to understand (Wei, 2020). The use of LLMs and other deep neural net based algorithms has led to a rise of interest in explainable AI, which aims to shed some light on the inner workings of these "blackbox" machines (Elton, 2020).

In this research, static, context independent word embeddings from the spaCY library are analysed in order to gain some insight on how statistical language models represent and associate words. The idea is that by analysing the word embeddings, also called word vectors, of language models, we will be able to better understand what types of things these models are able to learn, and make informed recommendations on how to improve them in terms of robustness, usefulness for language and reasoning tasks, and perhaps even in terms of computational efficiency. Ultimately, the vision is to use the results of this study to find ways to create better embeddings that can be useful for human level knowledge representation and potentially lead us to AGI.

2 Research Question and Aims

The research question (RQ) this project aims to address is "*What meaningful insights about language models can be derived from analysing word vectors?*"

Meaningful in this context means relating to linguistic concepts, relationships or structures that are either recognizable or already defined in human language. To frame the approach to answering this question, this research contains a literature review on works around natural language processing, explainable AI, and artificial general intelligence.

To further explore the topic, an artefact accompanying this secondary research is developed in Python using the spaCY library. SpaCY is a library for natural language processing (NLP). It comes preloaded with vector representations for a large vocabulary of common words in several languages. The tools that will be used to interpret the word vectors are:

- Principal Component Analysis (PCA)

- Similarity Scores
- Uniform Manifold Approximation and Projection (UMAP)

Finally, the results of the artifact will be analysed with regard to what is revealed in the structure of the data after being put through these models. This analysis will focus on identifiable patterns and linguistic concepts.

To conclude, the findings in this study help illustrate the need for ways to improve knowledge representation in vector form; an intelligible multimodal embedding is proposed as a potential path toward artificial general intelligence.

While rooted in natural language processing and eXplainable AI (XAI), the vision of this research ultimately is a contribution to the goal of AGI. This exploratory research aims to find a link between language and understanding of the natural environment in which one consciously exists. It has been suggested that human-level AI requires human-level knowledge representation and a natural language of thought (Jackson, 2021). This project is a step in the direction of a machine able to use vectorized sensory data to organize the information it comes across in practical ways. If a machine can develop a system of organization where associations between words and concepts can be represented, updated and utilized effectively, this machine could in theory self-correct, learn, and act in a more human fashion.

3 Related Work

This section will detail the relevant literature around the research in natural language processing, explainable AI, and artificial general intelligence.

3.1 Natural Language Processing

There are debates around whether the language models of today truly understand the text they process and generate (Arcas, 2022). While summarization tools and question answering performance seems promising in many ways, there are still glaring signs of brittleness when attempting to generalize knowledge to new data (Yin and Zubiaga, 2021). There are also some difficulties for these models when it comes to reasoning tasks and identifying nonsensical information (Gakhar, Chahal and Aggarwal, 2021). These examples illustrate some of the gaps that lie between today's AI and what some would call human level intelligence.

Research has indicated that scaling drives improvement in language models as larger models trained on more data tend to perform better on many NLP tasks. Big players in the space such as Google and Open AI all seem to be on board with the idea, continuing to add parameters regardless of how this impacts interpretability (Luitse and Denkena, 2021).

However, there is also extensive research around effective alternatives for scaling when it comes to improving language models in terms of task performance and computational efficiency (Givens et al, 2022). There is research indicating that methods

such as pruning, data quantization and incorporating external information in training are ways to make more compact models that rival the state of the art on various language tasks (Tang, T. *et al.*, 2020). Such studies have led to a growing community of those who believe that scale alone will only take us so far, and that data centric approaches and novel architectures may be necessary to create more robust and effective models (Strickland, 2022).

There are studies that have investigated improving word vectors in the past. A 2019 study found one could improve word vectors by feeding the models with morphologically informed vector representations of words (Gupta *et al.,* (2019). The more informed vectors were able to learn from a better starting point when computing semantically contextualized embeddings, and significantly reduced training time and data.

The vision for this research builds on these previous studies and follows a similar frame of thinking to that which is described by Piantadosi and Hill in their 2022 *Meaning Without Reference* paper. They describe the idea of conceptual roles being defined by the relationship of concepts to each other, and the notion that language is effectively a world modeling tool that captures our environment and allows us to communicate and interpret internal, external, and even theoretical states. They also stress the importance of studying internal states, both in cognitive science and in language models (Piantadosi and Hill, 2022).

With this in mind, exploring how language models represent words in relation to each other may give some valuable insight on how we as researchers might develop a system that uses these conceptual relationships as a world model which can be updated and generally applied to various tasks.

3.2 Explainable AI

As language models have been growing in their size and capabilities, there has also been an increasing interest in peeking under the hood of these models and the word vectors they produce (Rawal *et al.*, 2021). There has been an explainability model developed specifically for Transformers. In a recent study, Khanal et al. propose an inherently explainable Transformer model, providing more clarity on the inner workings of the machine learning architecture that drives many of the top performing models that exist today (Khanal *et al.*,2022). They found that instance-wise post-hoc causal explanation could provide better explainability results than opposing models without the need for training.

On the other hand, this research focuses on the vectors produced by language models and thus the same process and analysis can be applied to embeddings created by various architectures, making this explainability framework model agnostic as opposed to model specific.

Recently, it has been found that neurons in LLMs learn multiple features, as opposed to each neuron being responsible for one feature as some have previously assumed. This phenomenon, known as polysemanticity, makes it even more difficult to

decipher what blackbox models are learning because the information being processed is deeply convoluted (Scherlis *et al.,* (2022).

This study attempts to circumvent the issue of polysemanticity by focusing on the relationships between the word representations produced by language models, rather than focusing on the inner workings of these statistical models. In other words, instead of troubling ourselves with what these models are doing when they process text, in order to better understand these models the tools in this study focus on what appears to have been done with the information after the fact. For this reason, this research can be viewed as a post-hoc interpretability framework for language models.

In previous studies, it has been demonstrated that language models are able to represent semantic relationships mathematically, or put simply, do math with words. Using an early language model called word2vec: one can subtract the word *man* from the word *king*, then add the word *woman*, resulting in the word *queen (*Mikolov *et al.,* 2013*)*. In other words, as alluded to in this article, we could use algebra to solve the analogy:

man is to king as ***X*** *is to queen*

which can be mathematically represented as

king - man = queen - ***X***

With our words represented as vectors, one can solve for X like normal and the nearest word vector to the vector we end up with would be for *woman*. This is an example of using vector space, and vector arithmetic to gain insight into a word's meaning.

It has been hypothesised that the ability to solve, or perhaps more importantly construct, analogies could be key to unlocking robust AI that is able to effectively generalize information that it has learned previously to new tasks (Wang, 2009).

Diving deeper into the structure of language as it is understood by these algorithms and making it more intelligible may uncover a route to developing models that can recognize, label and utilize various types of analogies. This may in turn improve knowledge representation in machines and possibly even expand our own understanding of natural language.

3.3 Artificial General Intelligence

Since the days of Turing's Imitation Game and Asimov's robot code of conduct, the idea of an intelligent machine has fascinated mankind. However, for a long while, AGI has been viewed by many AI professionals as merely a matter of science fiction. It has been argued that the general problem solver does not exist, and that the quest to turn matter into mind is a futile effort (Garvey, 2021).

Today there seems to be an increased confidence in the research community that AGI, or at least human level AI, is achievable and that it's really a matter of *how* it will be achieved and how long it will take (Zhang *et al.,* 2022*)*. This could be due in part to the

properties of “emergent intelligence” we see demonstrated in today’s NLP tools (Arcas, 2022). Interactions with these AIs can be so remarkable that earlier in 2022, a Google employee claimed to believe that LaMBda, the company’s chatbot project which he had worked on, is actually sentient (Tiku, 2022).

In a 2021 study, Philip Jackson discusses the concept of a natural language of thought, which is described as the use of natural language to replace the formal languages that computers use. Jackson’s detailed review proposes that many of the arguments that are used against natural language as a language of thought for intelligent computers, can actually be used to advocate for its implementation.

For example, the argument that ambiguity is a barrier to the adoption of natural language in computers, can actually be reframed as advantageous when considering the flexibility and expressiveness of natural language. Also, while there is apparently more to intelligence than language, it can be argued that these concepts and observable phenomena that supposedly lie outside the realm of language, can all be described and interpreted through language. Though this may not be the way our brains actually work, language may potentially be effective as a proxy for AI to process and utilize information (Jackson, 2021).

Jackson and other professionals in the field have argued that a truly intelligent machine must be able to handle context and relevance changes, understand cause and effect, and be able to predict logical outcomes (Cao and Wooldridge, 2022). There have been debates around the importance of “grounding” words to real world experiences in order to understand them. This theory is supported by the school of thought that sees embodiment, physical form, as vital for achieving a true AGI. However, a recent study shows that grounding may not be necessary, and that navigating the relationships between concepts, or conceptual spaces, is what is important to understanding (Patel and Pavlick, 2022). Parallels drawn from the aforementioned studies, as well as patterns observed in the systematic literature review in Appendix III have given support and form to the author’s intuitions about language as a potential path to AGI.

This research aims to dive deeper into language models by exploring what they know and to what extent. The hope is that by understanding the relationships of the concepts that these models have encoded into word embeddings, we might be able to draw a path to developing autonomous systems that can effectively create and update representations of concepts based on data that they encounter and how it falls into their current understanding of the world.

4 Methodology: ***Tools, Techniques & Dataset***

4.1 Software Tools

The tools used for analysing the data are as follows:

The IDE used to develop this software is Google Collab, a cloud based software that allows you to run code cells and include text, similar to a Jupyter Notebook. The code is

written in Python, which was chosen for its simple syntax and broad set of tools specifically designed for machine learning and NLP. To access the Colab Notebook, click here.

The libraries used in this project are:

- Python - **Version 3.7.15** - *a general purpose programming language*
- Spacy - **Version 3.4.3** - *a library for NLP*
- Csv - ***Version 1.0*** - *a python library for working with CSV files*
- Whatlies - **Version 0.7.0** - *a library for visualising word embeddings*
- Scikit-learn - **Version 1.0.2** - *a machine learning library*
- Matplotlib - **Version 3.5.3** - *a library for plotting graphs*
- Numpy - **Version 1.21.6** - *a library for working with arrays and matrices*
- Pandas - **Version 1.3.5** - *a library for data manipulation and analysis*
- Umap-learn - **Version 0.3.10** - *a library for making UMAPS*
- Bokeh - **Version 2.3.3** - *a library for visualising data*

4.2 Techniques

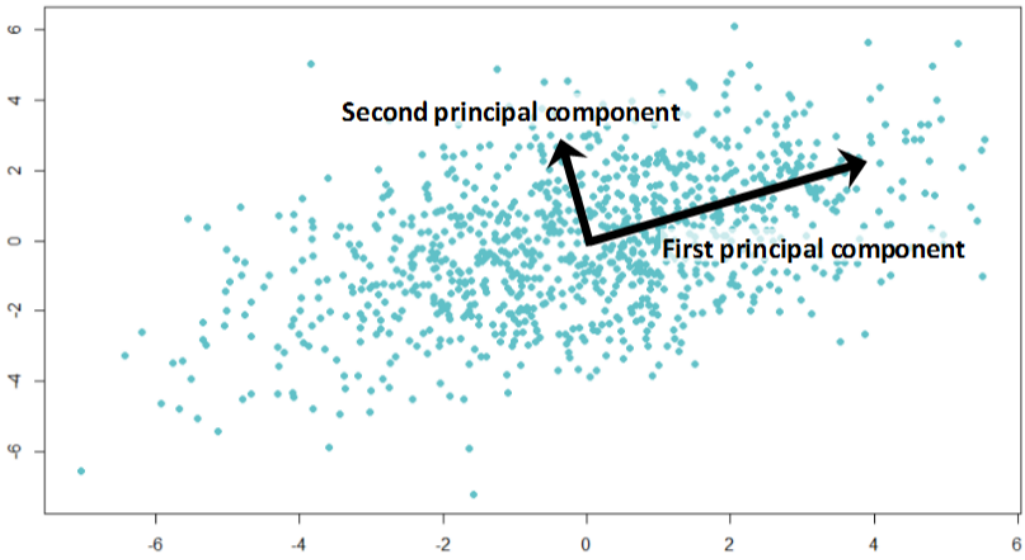
The techniques used for exploring the word embeddings are:

- a Principal Component Analysis (PCA)
- a comparison of Similarity Scores
- a Uniform Manifold Approximation and Projection (UMAP)

Principal Component Analysis

A principal component analysis is an unsupervised learning method used for dimensionality reduction and data exploration. The main idea of the algorithm is to compress the data into fewer dimensions while preserving as much statistical information as possible. A nice analogy for conceptualizing how a PCA works is a photographer taking a large group photo. Imagine you are trying to take a picture of 100 people at once. You want to find the angle (dimension) where you will be able to see the most faces in your photo (preserving the most possible information). This angle is the dimension of maximum variance, where the leftmost and rightmost person (or datapoint) are furthest apart. PCA achieves this by using the eigenvectors and eigenvalues of variables that are linear functions of the variables in the original dataset. The values of interest, the principal components, maximize variance and are uncorrelated with (i.e. perpendicular to) each other (Jolliffe and Cadima, 2016).

Figure 1.



source:https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

Figure 1. illustrates the main principle behind the PCA algorithm. The first principal component is the dimension of maximum variance in the data. The second principal component is the next dimension of maximum variance that is perpendicular to the first ranking dimension. This process continues for subsequent principal components until the desired amount of information is preserved.

The steps to the standard PCA algorithm are as follows:

- First the data must be standardized so that each variable contributes equally to analysis. This prevents bias from variables with larger scales gaining undue influence.

- Next, either a covariance or correlation matrix is calculated to identify the positive and inverse correlations between the variables in the dataset.

- After the matrix computations are complete, the matrix is used to find the eigenvectors and eigenvalues. The eigenvectors represent direction and the eigenvalues represent variance. They are computed using the formula

$$AV = \lambda V$$

  where A is a matrix, V represents the eigenvectors, and λ (lambda) represents the eigenvalue. The eigenvalue must be found first, This can be done using an identity matrix and some relatively simple matrix multiplications and algebra. One can then use the values to find the corresponding eigenvectors with a bit more matrix math and algebra. For the sake of being concise, further details will be spared but there is an excellent explanation of the process behind this here (Binieli, 2019).

- Finally, the eigenvalues are ranked in order of most to least variance, and a number principal components is selected based on the amount of

information one wishes to preserve. The corresponding eigenvectors can be used as the axes to display the data in graph form

For this experiment, the PCA is used to see the words based on the most "important" dimensions of their embeddings. In our case, the specific PCA algorithm used by the *whatlies* library is obscured, but it is assumed that the matrix *A* corresponds to a covariance matrix of our original 300 dimensional word vectors. A variable *n* is used to define the number of principal components (PCs) the algorithm returns. Though most information is usually concentrated in the top five or so principal components, in this experiment, this study observed up to 10 PCs in order to account for interesting insights that could be found in the tail end of the information. For example, certain groupings may be more clearly defined in a graph with the first and tenth principle components as the x and y axes, as opposed to the first and second principle components, depending on how information is captured and represented in the word embeddings.

The motivation behind this approach is that a PCA allows us to reduce the number of possible combinations for axes of the graph by reducing the data from 300 dimensions to the most relevant and representative dimensions. Ultimately, this makes identifying trends and patterns in the data easier and less time consuming.

Similarity Scores

A similarity score is a scalar (real number) value returned from some predefined function based on what the definition of similarity is in one's use case. In our experiment, we use the *.similarity()* function built into the spaCY library, which defaults to using cosine similarity between the vectors.

The formula for finding cosine similarity between two vectors is:

$$similarity(A, B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

where θ is the angle between the vectors, $A \bullet B$ is the dot product of a vector *A* and a vector *B*, and ||*A*|| ||*B*|| *i*s the L2 normalization or magnitude of the vectors *A* and *B*. The L2 norm can be calculated by finding the square root of the sum of the squared vector values (Karabiber, 2021).

In this study, we iteratively find the similarity for each word (A) in relation to every other word in the dataset (B). The results are then made into a table by putting A and B, along with their respective parts of speech, and their similarity score into a Pandas dataframe. Google Colab has a built in data_table module which allows for filtering and manipulating the data similar to how one can in Excel. Due to the scale of our dataset, we focus on the words that are most and least similar, the top and bottom 5 results for each

word, using the *.head()* and *.tail()* functions from Pandas respectively. The developed tool can display the data in a tabular form and show numerically the most and least similar words in the dataset as a whole, and the most and least similar words for each word. However, since a list of 3000 words has nearly 5 million possible combinations of word pairs, due to compute limitations, in this study subsets of the dataset (10 groups of 300 words) are used.

Figure 2.1.

```
client customer
0.8210183382034302


recommendation recognition
0.7242488861083984


recommendation orientation
0.7198281288146973


recognition orientation
0.7325651049613953


the highest similarity score is 0.8361517786979675
presentation orientation
```

Figure 2.2.

federal	representative	ADJ	NOUN	0.5017862319946289
federal	recommendation	ADJ	NOUN	0.5163717269897461
federal	agency	ADJ	NOUN	0.5918976068496704
federal	authority	ADJ	NOUN	0.6283546090126038
federal	legislation	ADJ	NOUN	0.7131893038749695

Figures 2.1. and 2.2. show the aggregate and word by word similarity tools respectively. The aggregate tool returns all the highest similarity scores based on a user selected threshold score. The scores are not returned in order but the highest and lowest scores are highlighted. The word by word similarity tool is in a flexible colab data table that allows for sorting and filtering much like Microsoft Excel.

Uniform Manifold Approximation and Projection

A UMAP, or Uniform Manifold Approximation and Projection, is a relatively new dimensionality reduction technique that captures both local and global structure in datasets. To do this, the algorithm uses a number of insights from algebraic topology and Riemannian geometry. For a deep dive into the math behind UMAP, see the original paper here (McInnes, Healy and Melville, 2018). Though the details of the mathematics are quite daunting, the intuitions behind the core principles are relatively simple.

The UMAP algorithm consists of two main steps: first the construction of a weighted neighbor graph from the high dimensional data, with weights representing how close a given point is to another. Then,using those weights the graph is projected down to a lower dimensionality while maintaining as much of the global structure as possible (Coenen and Pearce, 2019).

The steps to achieve this are as such:

- First, the manifold of the data is approximated by using a basic building block called a simplex to represent the topology combinatorially. A simplex is a *k*-dimensional object formed by connecting *k + 1* points. For example, a 0-simplex is a point, a 1-simplex is a line, a 2-simplex is a triangle, etc.

- We treat each point in our data as a 0-simplex. By extending out from each point by some radius *r*, then connecting points that overlap, we can construct sets of 1, 2, and higher-dimensional simplices. This does a reasonable job of approximating the fundamental topology of the dataset

- Rather than using a fixed radius *r*, a variable radius is determined for each point based on the distance to its *kth* nearest neighbor. Connectedness is then made "fuzzy" by making each connection a probability. The further the points are away from each other, the less likely they are to be connected. So, for example, the 4th closest neighbor to a point has a higher probability of connection than the 5th.

- To avoid any points being completely isolated, each point must be connected to at least its closest neighboring point.

- Finally, once the fuzzy simplicial complex is built, UMAP then projects the data into lower dimensions via fuzzy set cross entropy using stochastic gradient descent.

UMAP has several hyperparameters that will impact the visualisation of the data. The main ones are:

1. *n-neighbors, the number of neighbors to consider when approximating the local topology;*
2. *d, the desired embedding dimension;*
3. *min-dist, the desired distance between close points in the embedding space*
4. *metric- how distance is calculate*

In this study, the visualisation was produced using:

- n-neighbors = 15
- d = 2
- min-dist = 0.1
- and metric = ‘correlation’.

Figure 3.

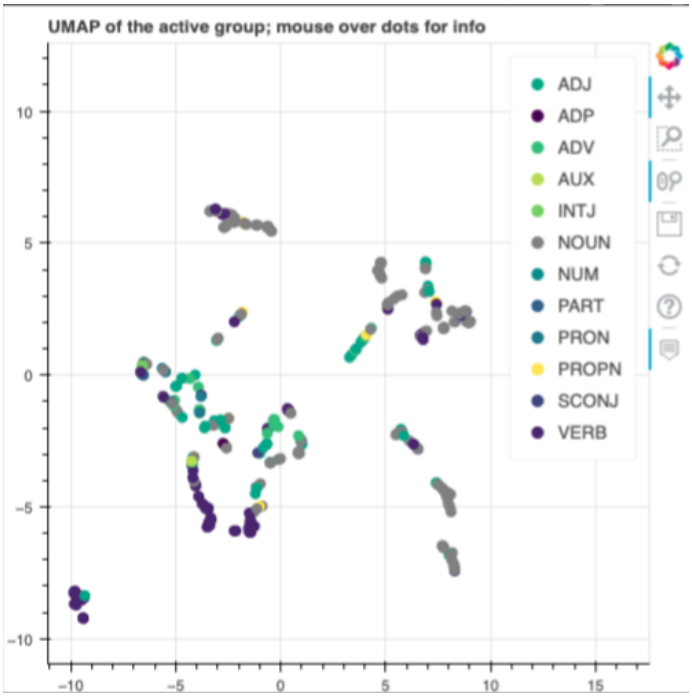


Figure 3. illustrates the UMAP produced by the aforementioned parameters. Notice the clustering by part of speech made visible by the color coding.

4.3 Dataset

The dataset for these experiments (Appendix I) is sourced from a list that is presented as the 3000 most common words in the modern English language. The list comes from, Education First (ef.com), a website with resources for learning American English. According to the site, the list is of course not meant to be exhaustive, but the words are selected based on frequency of use, and usefulness for new English speakers.This list is stored in a csv file called *mostcommonwords.csv*. Since the original list was in alphabetical order, there is also a list that is shuffled using the *random.shuffle()* function in Python, and named *preshuffled.csv*.

Since this study is exploring the relationships between words, no words were removed from the list; as stopwords (like ‘*the’*, ‘*a’* and ‘*is’*) and words with the same root can provide interesting information and possibly help illustrate patterns.

Before entering the analysis tools, each word in our .csv is converted into a spaCY doc object, then tokenized. From there, the data is vetted by ensuring that each word has a vector. After passing this check, the tokens, their parts of speech, vectors, normalized vectors, and strings are all put into their own lists for later use.

Figure 4.

```
from spacy.tokens import Token

with open('preshuffled.csv') as file:
   content = file.readlines()

normV = []
POS = []
CWV = []
CW = []
wordz=[]

def tokenify(content):
 for word in content:
   doc = nlp(word)
   token = doc[0]
   if doc.has_vector:
     CWV.append(token.vector)
     CW.append(token)
     normV.append(token.vector_norm)
     POS.append(token.pos_)
     word = str(word.rstrip())
     wordz.append(word)
tokenify(content)
```

Figure 4. is a code snippet that executes the process of preparing our data for analysis

Due to compute restraints and visualisation clarity issues, the words are broken into 10 groups of 300 before analysis. The selected group is first visualised in a vector space based on all 300 dimensions using the *whatlies* library. *Whatlies* is an interactive visualisation tool that is especially made for working with word vectors and is highly compatible with the spaCY library. One can zoom in and out, and pan through the graph to explore the space in great detail. A great feature of the library is that the axes of this graph can be altered in a myriad of ways.

The same library also handles the calculation and visualisation of the PCA algorithm, which allows you to use any of the constructed principal components as axes.

The data is then passed through the similarity scorer which returns the most and least similar words in the dataset, based on a threshold score that the user can set. The max and min scores in the dataset are highlighted as well as the pairs that produced them. The top and bottom similarity scores for each word are also returned and put into a Pandas dataframe. The number of results can be altered by the user, but this study observed up to 5 for each.

Finally, a UMAP is built and visualized using the *UMAP* library and *bokeh* for interactive plotting. *Bokeh* is similar to *whatlies* in that it allows for zooming and panning

to assist with in depth data exploration. However, *bokeh* goes a step further allowing for categorized coloring of data points, as well as revealing additional information on hover, This allows for a cleaner, more interactive visual.

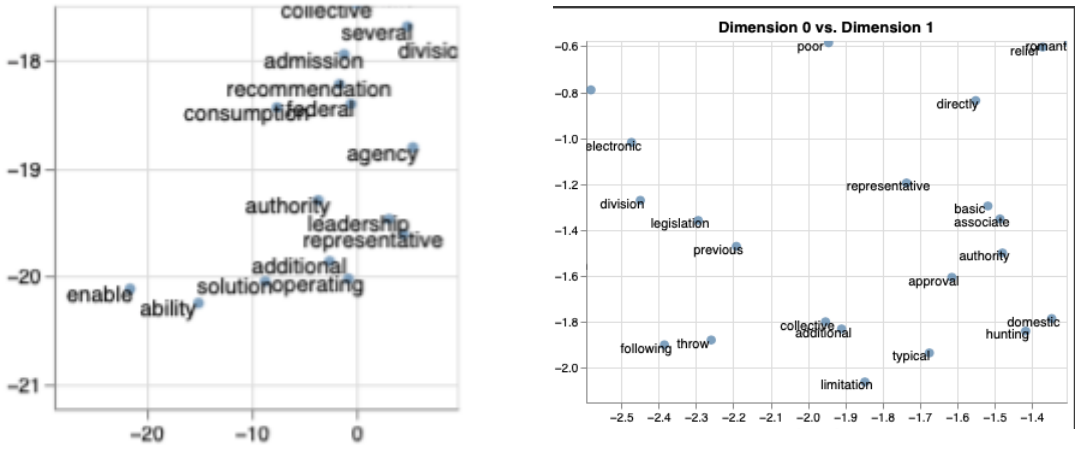
The outputs from these tools are then analysed and compared with each other, giving the results found in the following section. The results of the shuffled and non shuffled list both contribute to the findings described.

5 Results

5.1 PCA Results

While the dimensionality reduction is helpful in terms of limiting the number of axes combinations to iterate through, the PCAs definition of importance doesn't reflect much valuable information regarding language. There are some moments of clarity where a group of words seem related, but generally speaking there is little apparent logic to why it is grouping a certain way across the data as a whole.

Figures 5.1. & 5.2.



Figures 5.1 & 5.2: The image on the right shows the impact of the PCA algorithm. The left image shows a subset of the scatterplot produced by the original embeddings on the same set of words.

The somewhat disappointing performance of the PCA is likely due to the fact that the algorithm is simply looking for the space of most variance and not really manipulating the data points but rather showing them from different rotations. So while some dimensions end up displaying a few commonalities and relationships between a small set of data points, in the grand scheme of the data there are not many easily identifiable clusters or logical distributions that are worth noting. The PCA in this study was also a bit limited by the *whatlies* visualisation tool. Using a more robust tool like *bokeh* could perhaps produce more insights as there are more flexible and dynamic design options compared to what *whatlies* offers.

Similarity Score Results

The similarity score table is slightly more insightful. One interesting observed pattern is words that are synonyms or even abbreviations do not score as high in similarity as one might anticipate. As can be seen in *Figure 6.2*, sometimes words that are antonyms actually end up scoring very high in similarity, sometimes higher than words that are synonyms. This is likely because antonyms can often be used in the same contexts although they reflect completely different sentiments. For example, consider the sentences:

"I am very happy."

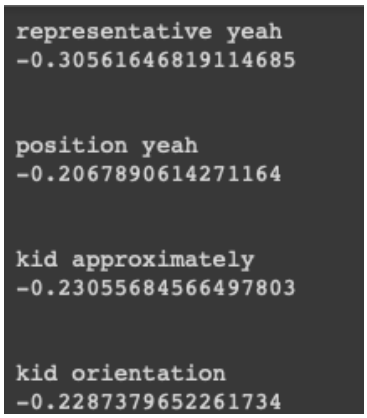
and

"I am very sad."

Though happy and sad, in our minds, represent *opposite* emotions; because they are frequently accompanied by similar words, they have a very high similarity score in the eyes of a statistical language model.

A downfall of the similarity score experiment is that outliers dominate the aggregate similarity tool so it is common that the same word will appear frequently in the lowest 5 to 10 scores. However, when looking at the more granular level, those outlier words follow a consistent logic. Even the highest scores for the outlier words are typically very low (below 50% similar), and they are usually still words that appear quite random.

Figures 6.1. & 6.2.



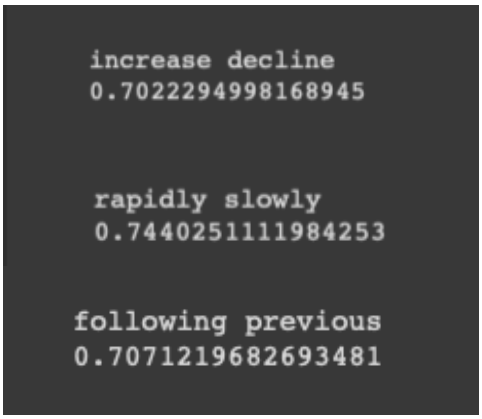


Figure 6.1. on the left, shows data outliers 'kid' and 'yeah' appearing frequently in the bottom results aggregate similarity score tool. Figure 6.2. on the right shows examples of antonyms with high similarity scores.

UMAP Results

The UMAP is by far the most intuitive tool for observing the relationships between words. The UMAP is able to cluster words by theme (or perhaps conceptual role) in a way which quite clearly illustrates that there is some level of understanding encoded into these embeddings. Similar words are grouped together thanks to the local k neighbors portion of

the algorithm. Also, the categorized color in our graph makes evident several groupings of parts of speech clusters; furthermore, the transition between clusters seems to follow some sort of logical progression from category to category, indicating that the global structure of the data is being represented quite well also.

An example of this low and high level clustering can be seen in Figures 7.1. & 7.2. In the UMAP visualisation, there is a progression from analog to digital communication methods. UMAPs ability to display the embeddings in this fashion shows impressive conceptual mapping encoded in the embeddings and could prove useful in further research around knowledge representation.

The biggest drawback for UMAP is that it's difficult to implement and fully understand at first. There are many adjustable hyperparameters, so figuring out what adjustments will produce a desirable graph is quite complex.

Figures 7.1. & 7.2.

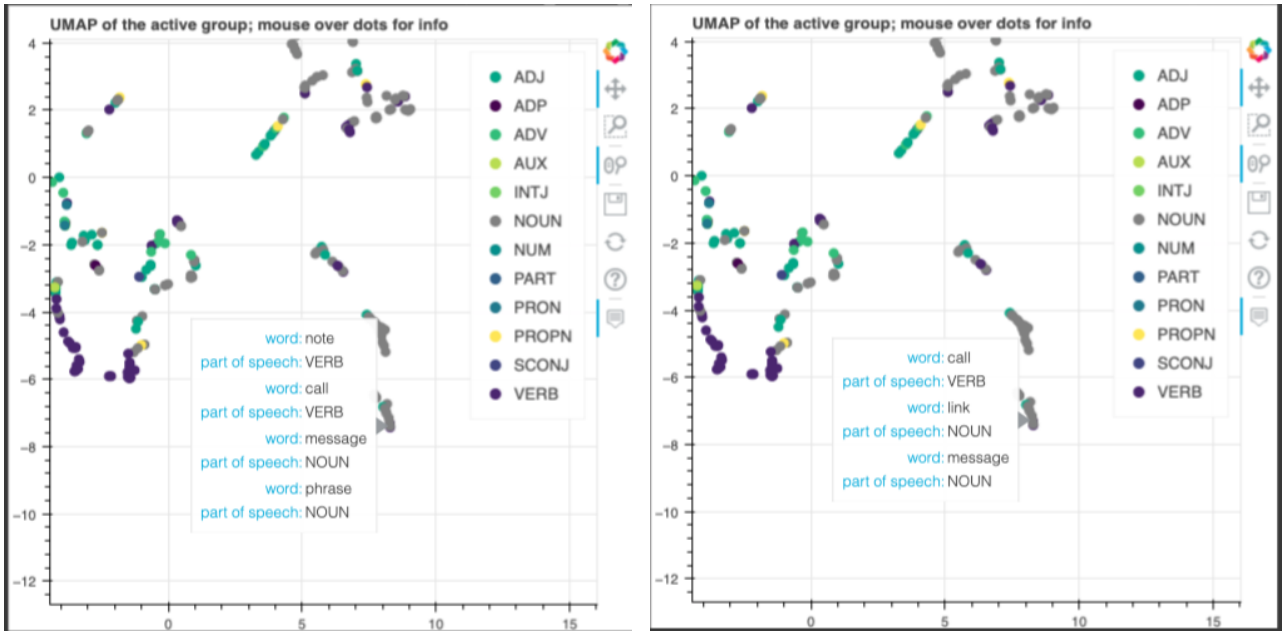


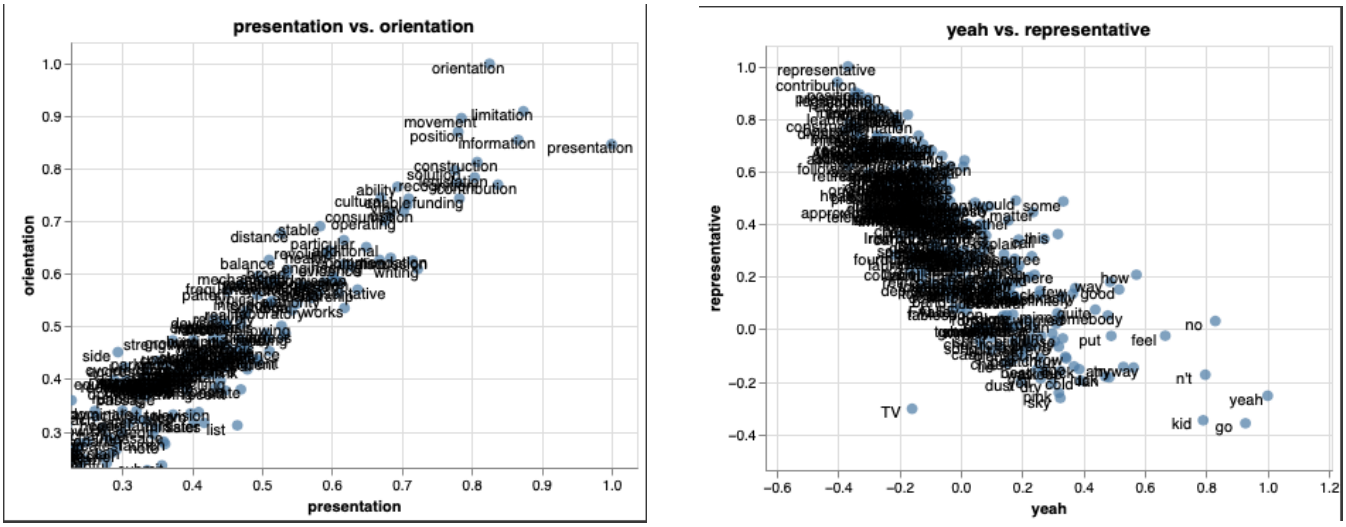
Figure 7.1. illustrates a progression from phrase, to note, to message, to call, which center around a theme of communication. Figure 7.2. shows an intuitive shift that occurs if you move up the graph, going from call to link, illustrating a trend of the communication becoming more digitized. Impressively the words 'online' and 'information' appear very nearby in this progression up the graph.

Cross-tool Analysis

In the cross tool analysis, an interesting recurring observation is that there are cases where words that appear as outliers in the similarity score never appear as such in the PCA graph. Similarly, words that appear as distant outliers in the PCA graph do not appear as such in the similarity score table. This indicates that the PCA has a different idea of what similarity means and what similarities are important to represent. The UMAP on the other hand seems to reflect more closely the scores that we see in our similarity table. This makes sense given how each algorithm is constructed.

A nice feature of the *whatlies* graphs that was possibly under-utilized is the ability to use a word as an axis. One can plot against two words just like any other dimension in the vectors space. Using this capability in conjunction with the similarity tool created some interesting effects in the data visualisation. When plotted along the most and least similar words from the aggregate similarity score tool, the returned graph showed a well distributed diagonal plot, with less dramatic outliers and less chaotic clustering than graphs from the standard dimensions of the vector columns.

Figures 8.1. & 8.2.



Figures 8.1. and 8.2 illustrate graphs with very similar and dissimilar words as axes. Figure 8.1 (left) shows a positive correlation between the word orientation and presentation, which have a similarity score of 0.8361517786979675. Conversely, Figure 8.2. (right) shows the negative correlation between the words 'yeah' and 'representative' with the downward slope and the points coming at opposite ends of the cluster. The latter pair of words has a negative similarity score of -0.30561646819114685

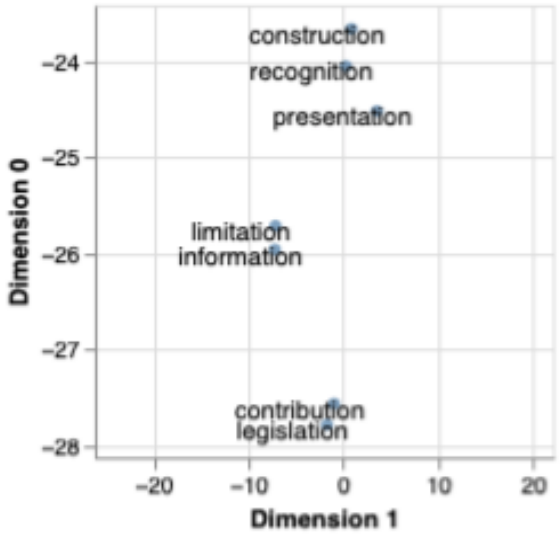
Also, across the analysis tools a somewhat strange observation is that suffixes seem to have a strong influence on similarity. An example that stood out is that the -tion suffix has a particularly strong influence. Words with this ending appear in strong clusters in both visualisation tools, and score very high in similarity. According to wiktionary, this suffix is used to form nouns meaning "the action of (a verb)" or "the result of (a verb)." With this in mind, it makes sense that such words would be used in a very specific context, making them especially similar in a way, particularly for a predictive statistical model.

However, outside of this observation, contrary to the original hypothesis, part of speech and phonetics are not as important as definition, co-occurrence, and associated concepts when clustering.

Figure 9.1.

Word 1	***Word 2***	***Similarity Score***
Identification	*Instruction*	*0.771188*
Reflection	*Consideration*	*0.753505*
Limitation	*Orientation*	*0.774931*
Identification	*Variation*	*0.799055*
Identification	*Definition*	*0.783396*

Figure 9.2.



Figures 9.1 and 9.2 illustrate the strong influence of the -tion suffix across analysis tools. The table in Figure 9.1 is a subset of the outputs from the similarity score tool. The graph in Figure 9.2 is a screenshot from a PCA graph showing the first 2 principal components.

5 Discussion and evaluation

5.1 Findings

Generally speaking, the findings were about as expected. In some ways, the vectors are spot on in their representations, correlating similar words and showing some signs of conceptual understanding. However, as seen clearly in the case of the similarity scores, words that exist on opposite parts of a spectrum are often viewed as more similar than they should be. This is due to the fact that today's word vectors are simply a reflection of the context in which words appear because of how statistical language models build their representations. The PCA showed us that since the components of our vectors do not correspond to any one concept, it is difficult to search our vector space for meaningful relationships. Furthermore, we cannot put a label to what the most "important" feature of a word is, because of the property of polysemanticity in current embeddings. The UMAP produces perhaps the most surprising results, displaying a remarkable understanding of nuance especially considering the use of static, statistically produced embeddings. The ability to represent local and global structure may be a beneficial knowledge representation tool for mapping more intelligible vectors and developing a world model from language.

5.2 Self-evaluation

While I was able to find out some interesting things about word vectors and language models, there are some elements of the project that I was unable to achieve in the allotted time frame. For instance, initially, the goal was to explore the word vectors using 5 more models. The missing models are:

- **A Growing Neural Gas algorithm (GNG),**
  - *To show the structure of the words in the vocabulary of the model. To show how adding a particular word to the network shifts the graph*
- **A graphical, predictive KNN**
  - *To see if a label such as part of speech, or even a word itself, can be predicted accurately by using the neighbors of a word.*
- **Permutation importance**
  - *To see what quality changes the structure of the graph the most and the least in order to see what dimensions/ features are most important to the grouping of the words*
- **K Means**
  - *To see how words are organized based on the number of clusters chosen*
- **H Clustering**
  - *To see how words are organized based on the number of clusters and grouping method chosen*

Also, the original plan included using embeddings from different contextual models such as GPT-3 or ELMO, and the multimodal model CLIP. Analysing various embeddings as opposed to just the static spaCY embeddings might give interesting results when compared to the findings of this study.

5.3 Project Management and Context Considerations

Plan vs Practice

The major difficulties initially came from a lack of a concrete vision for the endpoint of this project. A lot of time was spent refining a bunch of ideas around human level intelligence, language models, and explainability into a single coherent project. Additionally refining the research question and desired artefact to be developed was difficult due to the limited time factor.

The initial schedule of the Gannt chart (Appendix IV) that was submitted with the IPR (Appendix V) was significantly altered due to miscalculation of how difficult it would be to work with these word vectors and the spaCY library. Having some experience with machine learning libraries, I have encountered cases where very few lines of code could build out entire models, from training to analysis. This led me to believe it could be possible to work with several different algorithms in roughly the span of a month, as alluded to in the Self Evaluation.

However, due to the nature of my research and some compatibility issues with data types, I was unable to use the simple out of the box solutions that require limited code. Instead, a great deal of time was spent getting the data into various forms in order to be able to work with the visualisation tools that were ultimately selected. There were also some dependency issues with the visualisation tools selected, some libraries that were no longer being maintained, and difficulties using tools in Google Colab as certain things do not work the same in the web based IDE.

With that being said, the timeline described in the original Gannt chart was intentionally optimistic to encourage running into these problems early and account for inevitable delays in the coding portion of the project. Being resourceful, reading the documentation and searching error codes helped me resolve most of the issues. I learned a great deal about debugging and the patience and persistence required to effectively implement tools from various sources.

Regarding the literature review and secondary research more generally, I feel like I gave myself a decent head start by collecting papers and sourcing information as early as fall of 2021. Building on this preliminary research and refining the project throughout the year relieved a great deal of pressure toward the end of the project timeline, allowing me to simply add more recent relevant findings as I went along.

The most difficult part of this project actually turned out to be keeping myself from becoming too caught up in playing with the artefact. Being curious about what potential insights could be found if I was able to manipulate the data in more ways almost began to work against me. Once the technical difficulties had been overcome, wanting to explore the data and find more interesting insights from the experiments also contributed to the deviation from the original project timeline.

Commercial Context

As touched on in the Introduction, this report stands in the face of immense investment by large organizations into statistically based LLMs. However, some researchers and commercial users are realizing that smaller models have the appeal of being more accessible and "greener" as they aren't as compute intensive to train. But in order to make these models competitive with the data giants, alternative methods must be explored (Albalak *et al.,* 2022).

Explainability in AI is becoming increasingly valued in the commercial space as AI begins to take on more high stakes tasks. There are many studies around the use of explainable AI and how it can help with adoption and trust in the medical field particularly (Saraswat, 2022). The need to meet the demands of sensitive markets like healthcare as we expand and integrate AI capabilities will likely keep explainability in AI relevant for the foreseeable future.

While some still consider AGI a lofty goal, there are companies today such as Deepmind and OpenAI working toward reaching this goal. Regardless of whether AGI is achievable or not, it is certain we can still more closely simulate human level intelligence and big players in the public and private sector are investing heavily in doing so (Vykhodets, 2022).

ELPS Analysis

Ethical Issues

It's been said that one of the main difficulties with analysing the ethical impact of artificial intelligence (AI) is overcoming the tendency to anthropomorphise it (Ryan, 2020). We have a hard time wrapping our heads around the idea that the superintelligent AI of the future may not even come in an embodied form, let alone human form.

Nonetheless, there are already cases today where the capabilities of language models have been considered problematic. The problem is that our use of human language reflects our stereotypical biases, thus AI systems trained on human language carry these historical biases. Anyone who has ever been on Twitter knows just how dangerous it would be to have an AI trained on the thoughts of the masses (Schramowski, 2019).

Beyond language models, there are other software based AI's impacting our daily lives such as credit appraisal and fraud detection algorithms. As this technology becomes more embedded in our lives, our ability to trust and understand AI becomes even more important. For example, in medicine, a study showed that the most frequent issue of working with AI was communicating to patients how results were achieved or determined. (Ursin, Timmermann and Steger, 2022) For this reason among many others, one can argue that explainable AI is innately tied to AI ethics .

While the potential contributions to knowledge from AGI go beyond our wildest imagination, some say there is an equal risk that this technology could be disastrous for humanity. It has been argued that if we create superintelligence without any restriction of

its power, it could become impossible to control, and the AI or AGI civilisation might be able to replace human civilisation. On this ground, some people will even go as far as to say that we should stop researching AGI altogether. On the other hand, there are those who argue that replacing humans to an extent, for example in the manual labor workforce, could prove beneficial for humans long term *(*Maruyama, 2022).

The potential of these technologies coupled with the uncertainty that surrounds their future puts a lot of responsibility on the people developing and regulating these technologies.

Legal Issues

There are many legal issues surrounding AI in general, most of which center around a lack of knowledgeable regulatory bodies and the ambiguous legal status of automated systems (Rodrigues, 2019). Regarding NLP and AGI the issues center around data privacy and liability issues.

Legal scholars and data protection professionals believe that AI introduces many privacy and data protection challenges. There are issues of informed consent and surveillance. There have also been cases of infringement on data protection rights of individuals, e.g. the GDPR rights of access to personal data and the right not to be subject to a decision based solely on automated processing (Rodrigues, 2020). The public is not aware what data these models are being trained on and oftentimes may not even be aware that they are being subjected to the decisions of a machine.

There are also liability issues when it comes to AI. These autonomous systems can cause damages but in many cases there is a lack of accountability for these agents.

In a 2016 paper, Kingston discusses whether criminal liability could ever apply to AI, and to whom it might apply instead. Regarding civil law, there are debates that an AI program is a product and should be subject to product liability legislation, to which the tort of negligence applies. However, as there are many parties involved in developing, deploying and maintaining an AI system; liability is difficult to establish when something goes wrong. (Rodrigues, 2020). This becomes even more complex as agents become more autonomous. At what point can previously responsible parties be absolved?

Professional Issues

As previously alluded to, the data that AI algorithms are trained on has dramatic implications for how these technologies behave and ultimately impact the public. AI researchers and professionals need to be working to mitigate bias. Bias describes problems related to the gathering or processing of data that might result in prejudiced decision making on the bases of features such as race, gender, and financial status. There are many points in the lifecycle of a project where bias can be introduced, from data collection and preprocessing all the way to the process of making decisions based on model outputs (Ntoutsi et al., 2019).

Explainable AI has been increasing in popularity among researchers in recent years due to the rise of neural networks (Elton, 2020). However, some researchers are going deeper, asking the fundamental question: *What is an explanation?* In a 2019 study, Mittelstadt, Russell, and Wachter reject such usage of the term "explanation," stating that while it might be appropriate for a research professional, the outputs of many XAI tools still do not make sense for the average person (Ntoutsi et al., 2019).

In the case of NLP research, there has been heavy investment from large firms in scaling large language models. While the achievements are remarkable, this negatively impacts the democracy of developing this technology, making it more difficult for independent researchers and students to recreate studies and contribute new findings. Ultimately, it puts power in the hands of the companies with the resources and pockets to fund these projects (Luitse and Denkena, 2021).

It can be argued that increased transparency and more uniform standards are vital to safely developing and implementing tools with the potential societal impact that NLP and AGI possess.

Social Issues

The social implications for this research center around how human robot and human computer interaction will be impacted by the advancement of AGI. Currently there are debates around how much AI can be trusted to make decisions in high stakes environments such as medical diagnosis. There are some who argue that we cannot truly 'trust' AI because for one to be trusted they must be able to be held responsible for actions, which AI cannot. Instead, the author suggests, we should discuss whether AI is 'reliable' and shift the responsibility onto those developing, deploying, and using these technologies (Ryan, 2020).

However, as AI advances and we move toward self-aware decision-making agents, we must ask ourselves how we cope with the shift in the dynamic of our relationship with technology. At what point past the threshold of the Turing test should we shift from a dynamic of user and tool to being and (at least apparent) being? Will these clever machines ever be considered conscious enough to be held responsible for an action? Will we owe these autonomous machines the same common decency as humans and other naturally sentient beings? Does a machine who does wrong go to prison or get turned off? These questions are just scratching the surface of what should be considered about the implications of introducing this new class of being into our society.

6 Conclusion

Overall, despite the adjustments to the initial plan, the system was successful in illustrating some interesting insights of what language models know. The use of multiple tools to analyse the same data allowed for not only a variety of the results, but additional insights to be drawn from cross tool analysis. Ways to improve this study might be to clean the data, use a better visualisation tool for the PCA, and remove outliers from the similarity score table. The biggest success of this project was coming across the UMAP graph and learning how it might be valuable in future data exploration. Using these tools on

contextual embeddings and multimodal embeddings will likely be pursued in the near future.

While the results from the UMAP indicate levels of understanding, the results from this study still point to a need for more robust vectors that can capture important information about words. I would like to conclude this study by proposing the pursuit of embeddings that include numerical representations of a word's: definition, part of speech, synonyms, antonyms, associated images, and audible pronunciation, and potentially other relevant qualities. Studies have shown that multimodal embeddings have the advantage of additional context and perform better on some NLP tasks than traditional text based embeddings (Habibian et al., 2017). By giving more clarity and depth to the conceptual space, the intelligible embedding described may be able to concisely encode information that today's LLMs are struggling to grasp.

References

- Albalak, A. *et al.* (2022) ‘Data-Efficiency with a Single GPU: An Exploration of Transfer Methods for Small Language Models’.
- Arcas, B.A. y (2022) ‘Do Large Language Models Understand Us?’, Daedalus (Cambridge, Mass.), 151(2), pp. 183–197.
- Binieli, M. (2019) An overview of principal component analysis, Medium. We've moved to freeCodeCamp.org/news. Available at: https://medium.com/free-code-camp/an-overview-of-principal-component-analysis-6340e3bc4073 (Accessed: December 1, 2022).
- Cao, L. and Wooldridge, M. (2022) ‘A New Age of AI: Features and Futures’, IEEE intelligent systems, 37(1), pp. 25–37.
- Coenen, A. and Pearce, A. (2019) Understanding UMAP, PAIR. Available at: https://pair-code.github.io/understanding-umap/supplement.html (Accessed: December 1, 2022).
- Elton, D.C. (2020) ‘Self-explaining AI as an Alternative to Interpretable AI’. Cham: Springer International Publishing (Artificial General Intelligence), pp. 95–106.
- Fei, N. et al. (2022;2021;) ‘Towards artificial general intelligence via a multimodal foundation model’, Nature communications, 13(1), pp. 3094–3094.
- Gakhar, M., Chahal, N. and Aggarwal, A. (2021) ‘Understanding and Evaluating Commonsense Reasoning in Transformer-based Architectures’. IEEE, pp. 1–5.
- Garvey, S.C. (2021) ‘The "General Problem Solver" Does Not Exist: Mortimer Taube and the Art of AI Criticism’, IEEE annals of the history of computing, 43(1), pp. 60–73.
- Gupta, V. et al. (2019) “Improving word embeddings using kernel PCA,” Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019) [Preprint]. Available at: https://doi.org/10.18653/v1/w19-4323.
- Habibian, A., Mensink, T. and Snoek, C.G.M. (2017) ‘Video2vec Embeddings Recognize Events When Examples Are Scarce’, *IEEE transactions on pattern analysis and machine intelligence*, 39(10), pp. 2089–2103.
- Jackson, P.C. (2021;2020;) ‘On achieving human-level knowledge representation by developing a natural language of thought’, Procedia Computer Science, 190, pp. 388–407.
- James, A.P. (2022;2021;) ‘The Why, What, and How of Artificial General Intelligence Chip Development’, IEEE transactions on cognitive and developmental systems, 14(2), pp. 333–347.
- Jolliffe, I.T. and Cadima, J. (2016) ‘Principal component analysis: a review and recent developments’, Philosophical transactions of the Royal Society of London. Series A: Mathematical, physical, and engineering sciences, 374(2065), pp. 20150202–20150202.

- Karabiber , F. (2021) Cosine similarity, Learn Data Science - Tutorials, Books, Courses, and More. Edited by R. Lewis and B. Martin. Available at: https://www.learndatasci.com/glossary/cosine-similarity/ (Accessed: December 1, 2022).
- Khanal, S. et al. (2022) 'Causality for Inherently Explainable Transformers: CAT-XPLAIN'.
- Luitse, D. & Denkena, W. 2021, "The great Transformer: Examining the role of large language models in the political economy of AI", Big data & society, vol. 8, no. 2, pp. 205395172110477.
- Maruyama, Y. (2022) 'Moral Philosophy of Artificial General Intelligence: Agency and Responsibility'. Cham: Springer International Publishing (Artificial General Intelligence), pp. 139–150.
- Marzban, R. and Crick, C. (2021) 'Deep NLP Explainer: Using Prediction Slope to Explain NLP Models'. Cham: Springer International Publishing (Artificial Neural Networks and Machine Learning – ICANN 2021), pp. 447–458.
- McInnes, L., Healy, J. and Melville, J. (2018) 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'.
- Mikolov, T. et al. (2013) 'Efficient Estimation of Word Representations in Vector Space'.
- Musil, T. (2019) 'Examining Structure of Word Embeddings with PCA'. Cham: Springer International Publishing (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 211–223.
- Ntoutsi, E, Fafalios, P, Gadiraju, U, et al. Bias in data-driven artificial intelligence systems—An introductory survey. WIREs Data Mining Knowl Discov. 2020; 10:e1356. https://doi-org.ezproxy.herts.ac.uk/10.1002/widm.1356
- Patel, R. and Pavlick, E. (2022). Mapping Language Models to Grounded Conceptual Spaces. [online] openreview.net. Available at: https://openreview.net/forum?id=gJcEM8sxHK [Accessed 3 Oct. 2022].
- Piantadosi, S.T. and Hill, F. (2022) 'Meaning without reference in large language models'.
- Rajani, N.F. et al. (2019) 'Explain Yourself! Leveraging Language Models for Commonsense Reasoning'.
- Rawal, A. et al. (2021) 'Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges and Perspectives', IEEE transactions on artificial intelligence, pp. 1–1.
- Rodrigues, R. (2020) 'Legal and human rights issues of AI: Gaps, challenges and vulnerabilities', Journal of Responsible Technology, 4, p. 100005.
- Ryan, M. (2020) 'In AI We Trust: Ethics, Artificial Intelligence, and Reliability', Science and engineering ethics, 26(5), pp. 2749–2767.

- Saraswat, D. *et al.* (2022) ‘Explainable AI for Healthcare 5.0: Opportunities and Challenges’, *IEEE access*, 10, pp. 84486–84517.
- Scherlis, A. et al. (2022) ‘Polysemanticity and Capacity in Neural Networks’.
- Schramowski, P. et al. (2019) ‘BERT has a Moral Compass: Improvements of ethical and moral values of machines’.
- Strickland, E., 2022. Andrew Ng, AI Minimalist: The Machine-Learning Pioneer Says Small is the New Big. IEEE Spectrum, 59(4), pp.22-50.Tang, T. et al. (2020) ‘Incorporating external knowledge into unsupervised graph model for document summarization’, Electronics (Basel), 9(9), pp. 1–13.
- Tiku, N. (2022) ‘Google fires engineer who said its AI was sentient’, The Washington post.
- Ursin, F., Timmermann, C. and Steger, F. (2022) ‘Explicability of artificial intelligence in radiology: Is a fifth bioethical principle conceptually necessary?’, Bioethics, 36(2), pp. 143–153.
- Vykhodets, R.S. (2022) ‘China”s AI Strategy’, *Евразийская интеграция: экономика, право, политика*, 16(2), pp. 140–147.
- Wang, P. (2009) ‘Analogy in a general-purpose reasoning system’, Cognitive systems research, 10(3), pp. 286–296.
- Wei, J. et al. (2022) ‘Chain of Thought Prompting Elicits Reasoning in Large Language Models’.
- Yin, W. & Zubiaga, A. 2021, ‘Towards generalisable hate speech detection: a review on obstacles and solutions’, PeerJ. Computer science, vol. 7, pp. 1-38.
- Zhang, B. et al. (2022) ‘Forecasting AI Progress: Evidence from a Survey of Machine Learning Researchers’.
- Zhou, X. et al. (2019) ‘Evaluating Commonsense in Pre-trained Language Models’.

Appendices

Appendix I

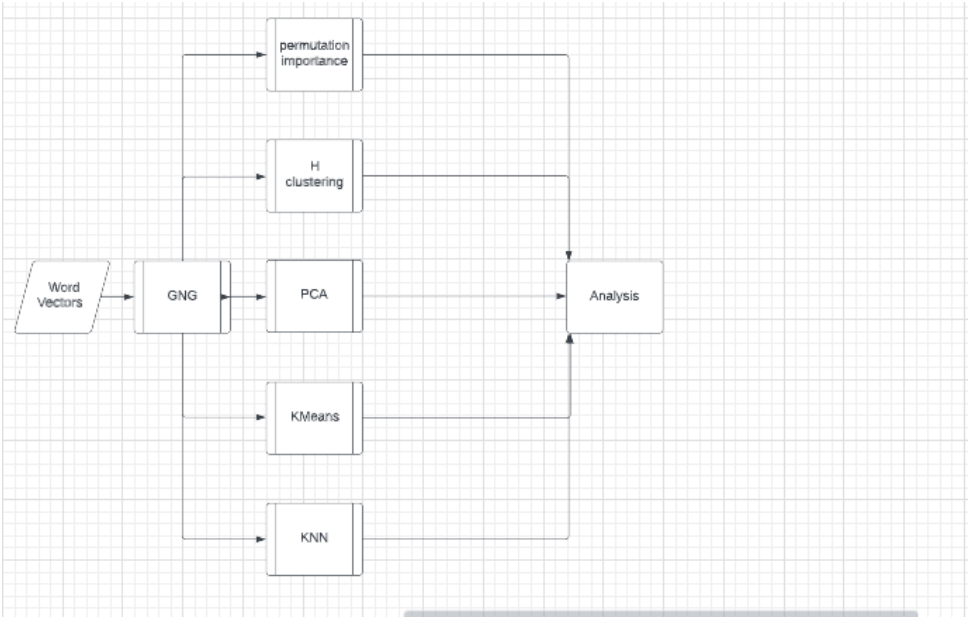
List of 3000 words used in the study

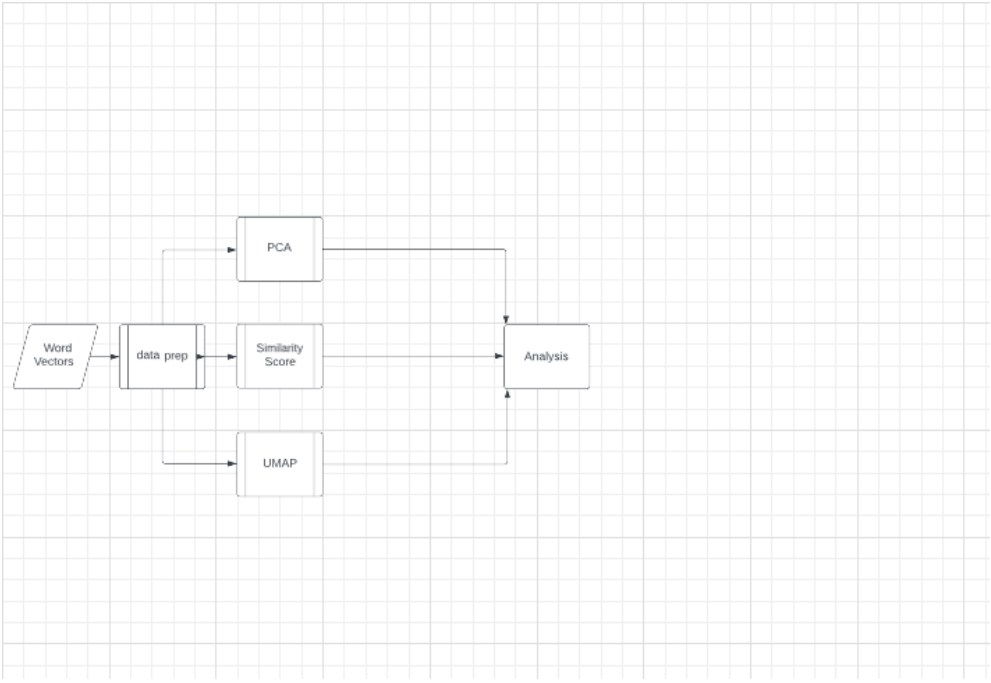
mostcommonwords.csv

preshuffled.csv

Appendix II

Diagrams of original plan and final project



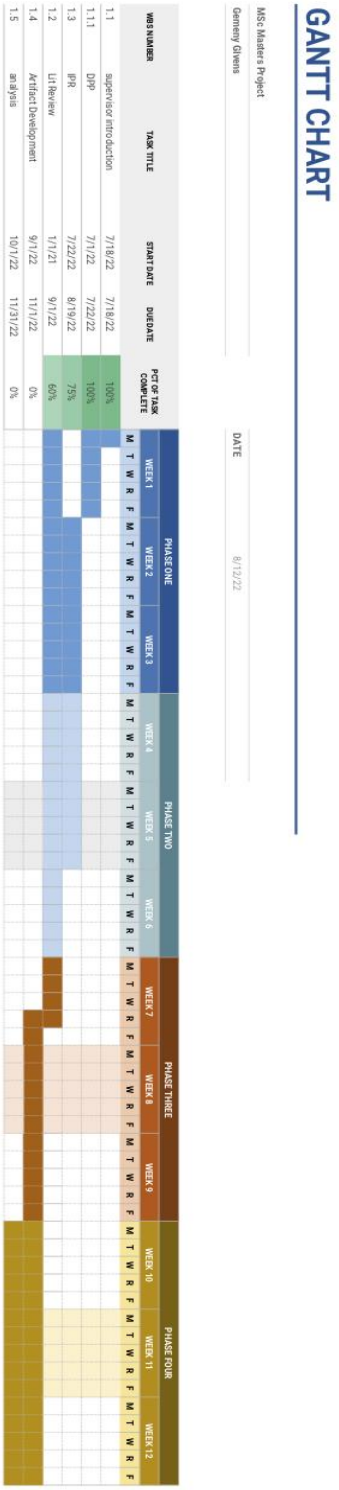


Appendix III

Tell me More: A Systematic Review on Improving Language Models

Appendix IV

Initial Gantt Chart



Appendix V

The IPR which sheds light on some of the early thinking behind this study

Appendix VI

Links that helped with code

https://datagy.io/python-split-list/

https://stackoverflow.com/questions/38565104/how-to-name-list-of-lists-in-python

https://www.geeksforgeeks.org/python-pair-iteration-in-list/

https://umap-learn.readthedocs.io/en/latest/plotting.html

https://docs.bokeh.org/en/2.4.0/docs/user_guide/annotations.html?highlight=legend

Appendix VII

AGI smells fear