Analogy Generation using Pre-trained Language Models

Christian Balevski, Julius Dsouza
Deep Learning Fall 2021
Tandon School of Engineering
New York University

Problem Description

Humans have an uncanny ability of finding analogical inspirations in distant domains which is an extremely powerful way of compressing huge chunks of data. Interpreting these analogies typically requires a strong intuition while identifying the deep underlying semantic relationships. Understandably, the results of this are subjective and tend to vary from person to person. Despite this, many possible use-cases for analogy generation still emerge, such as: representing a company's ideals and core values using analogies, using an analogy for a product marketing campaign, explaining concepts and topics for the purpose of education and so on.

This potential to find and develop useful analogies is crucial in driving innovation across a plethora of domains. Analogies serve a central role in human communication and provide a means to convey the relationship of a given sentence by expressing it in a different sentence while retaining its original relationship in order to highlight its significance. However, creating or identifying analogies in reality is very complex and an arduous task. Analogy generative models either rely on large and expensive data sets or produce models that are limited in functionality (word analogies). These approaches are typically built using data with predefined analogy structures and are often limited in scope. However, there has been a recent paradigm shift in NLP, leading to the development of sophisticated language models built using transformer networks such as PEGASUS, BERT or GPT-2. These models, after fine-tuning, produce state-of-the-art performance in a myriad of downstream NLP tasks.

In this paper, the authors investigate a new method for generating analogies by leveraging Natural Language Generation (NLG) models to produce valid target-word explanations with increasing levels of abstraction and analogation through restricting vocabulary sets, akin to the popular party game *Taboo*.

Literature Survey

Analogy discovery and reasoning has been a research-intensive topic in a multitude of disciplines, such as philosophy, psychology, cognitive science, linguistics and artificial intelligence since classical times. Metaphor interpretation and analogical reasoning are typically based on structure-mapping or on taxonomic abstraction. Both these standpoints have their own advantages and disadvantages and is dependent on the specific use case.

In one of the earlier models, Turney used analogies from the SAT dataset to measure the semantic similarity of the Latent Relational Analysis Model (LRA). This model was severely limited due to being solely focused on relation mapping and this was insufficient for generating satisfactory lexical analogies. (Turney 2005). Crowd sourcing answers has also been attempted, with no performance gains over statistical methods (Lofi, El Maarry, and Balke 2013). Neither of these approaches produce interpret-able justifications, focusing only on providing correct answer choices.

At Cardiff University, researchers applied pre-trained language models to the task of analogy identification only to end up with mixed results. One of the challenges highlighted is the performance of these models due to varying levels of difficulty and a plethora of topics. The authors noted that state-of-the-art performance could be achieved provided the language models were carefully tuned. This suggests that further work can be done in the selection and preparation of training data. (Ushio et al. 2021)

Many researches have followed the approach to identifying analogies using structural similarity of sentences. However, this typically requires a rich data set with appropriate features and sufficient training samples.

Other approaches focus on a very specific domain and are usually limited to a four-term analogy problem in which the first 3 words are given and the model identifies the relationship between the first and second word and uses such a relation in conjunction with the third word to find the fourth word. (Hope et al. 2017)

Dataset

Our dataset will consist of two parts: (1) word definitions and (2) source words with restricted vocabulary sets.

For word definitions we will be using the OPTED (Online Plain Text English Dictionary), which is based on "The Project Gutenberg Etext of Webster's Unabridged Dictionary" which is in turn based on the 1913 US Webster's

Unabridged Dictionary. It contains 54,555 words along with their corresponding definitions.

For the second part of our data set, we will be using a list of the 1000 most common English nouns which we have extracted from the British National Corpus, which includes a large collection of documents in British English. These will be the source words (S). The source words will provide the context for which the NLG model needs to produce an explanation for. In order to move from explanation to analogy, we will restrict the vocabulary the NLG model can use. The restricted vocabulary set will consist of both synonyms and collocations for the given source word. For example, for a given source word $s_i \in S$, $s =$ "government".

$$\text{synonyms} = [\text{authority, law, politics, ministry...}]$$

$$\text{collocations} = [\text{central, federal, local, national...}]$$

We chose to include both synonyms and collocations in the restricted vocabulary as we suspect that limiting domain specific terminology will force the model to produce explanations with increasing levels of abstraction. (Both synonyms and collocations can be sourced from NLTK package in python.) The number of terms included and ratio of terms between synonyms and collocations in the restricted vocabulary set will be one of the parameters that will be modified during this project to determine optimal performance.

Model

The OpenAI's GPT-2 language generation model as the pre-trained model has been used for this project. GPT-2 has been shown to be highly effective at generating cogent text especially through fine tuning.

1. First, split the (1) dictionary dataset into train, validate, and test sets to evaluate performance.
2. Fine-tune the GPT-2 model on producing definitions by feeding in the word-definition pairs from the dictionary dataset and optimizing accordingly.
3. Input words from (2) source word with restrictions and have the GPT-2 generate a definition
4. Vary the number of words in the restricted vocabulary set to produce definitions with varying levels of abstraction.
5. Use a reverse-dictionary (www.datamuse.com/api/) to approximate the original source word from the generated definition
6. Convert the predicted source word and true source word into vectors (GloVe) and compare similarity using various evaluation metrics (cosine-similarity, dot-product).
7. Lastly, we update GPT-2's definition analogy prediction using various optimization methods.

   The Fine-Tuned GPT-2 Model is used to generate a number of definitions for the desired word where the definitions serve as candidate analogies words. This results in definitions with the highest confidence similarity. This is then used as an input to our subsequent transformer model which computes the scores for the synonyms generated by the weights of the previous model. The GPT-2 model learns using the byte pair encoding of the words which stores the context as well as the meaning of the word. This provides us a lot more insight instead of using classical text vectorization techniques such as Term Frequency-Inverse Document Frequency, Information Gain which leads to sparse matrices with higher dimensions, with no semantic information and a strong possibility of overfitting taking place.

Loss Function

The Cross Entropy Loss Function is the default loss function that is being used for this project. The cross entropy loss is a measure of the difference between two probability distributions. In the case of this project, the model tries to predict the next token in a sequence given the previous tokens in that sequence. It therefore captures the accuracy of the predicted next token given then ground truth token and the previous tokens.

For a method of optimization, AdamW (Adam with weight decay) was selected. AdamW is advantageous over Adam because regularization term does not end up in the moving averages and this creates more favorable train and validation loss. The specific AdamW parameters used are learning rate $lr = 5 \times 10^{-4}$, weight decay of $wd = 10^{-2}$ and epsilon of $\epsilon = 10^{-8}$ are used.

Hyper Parameters

For fine-tuning the model to generate definitions, the definition length has been set to 50 tokens. If the length of the definitions are below this limit, they are padded to meet this requirement. GPT-2 langauge model includes Top-K sampling which is a sampling protocol where only the K most likely next words are considered and then the probability mass distribution is redistributed across these K words. It practice, this generates more natural sounding language by limiting the model's ability to generate new text to only the most likely next few words. A value of $k = 50$ has been used for the generative model as mentioned earlier.

Th Top-K method on its own, however, can create issues where words are sampled from a very sharp distribution as compared to a flat distribution. In practice, another sampling protocol is introduced called Top-P sampling. Top-P sampling dynamically adjusts the size of the sample by selecting the smallest set of possible words whose cumulative probability exceeds the probability p. For a sharp distribution of words, only the first few will be filtered while a more uniform distribution of words will result in a larger sampling. The model in this project uses a value $p = 0.95$. Together, these two sampling methods help produce the most human-readable text.

Scoring Metric

The scoring function used in this method presents a crucial ingredient in determining the success of the model. The scoring function needs to be both objective, but also flexible

Target Word	Analogy	Best Guess	Cosine Similarity	Avg of all Guesses
chemistry	Knowledge of the nature, character, powers, and phenomena which can be concealed from view, as by means of an instrument, or by man employed in making it; knowledge of poisons; intelligence. Bacon.	science	0.6224	0.2008
biology	A description; explanation. Bp. Hall. Sir T. Browne. "Biology has not been a mode of thought or action, but it is the foundation of our understanding." Burke.	philosophy	0.464	0.158
physics	A science which treats of the origin, qualities, and forces peculiar to bodies; physical science. The body is the body itself, or its relation to an external power. Darwin.	physics	1.000	0.272
ocean	Of or pertaining to the Oceans. -- n. An ocean boat of war and full length; a vessel for sea voyages, in which it carries a large number of vessels at a great price, -- distinguished from a port on the Oceans, at which all vessels are conveyed at a great price with a large number of men; as, the coast of China is a port of call; the coast of Europe is a port of call.	oceanic	0.766	0.302
germany	One of two or more European marine reptiles inhabiting the Great Lakes. The commoner eel is often brownish black; the gray cedar, usually a dark white shell, like that on which a true European bird was born.	congo eel	0.499	0.132
Germany	The German language. See German, 1st German language, under German. [Written also Germanw.] 2. A dialect spoken by German inhabitants of Great Britain and Ireland for various purposes; also, a dialect, with or in German. It is native-specked only in its own language and dialect, so that, in any dialect, English is spoken only on Germanic language or dialect.	ticino	0.731	0.276
teacher	One who instructor. [R.] Johnson. One who teaches in his classes; a teacher of old age, or as distinguished from a student in the school.	pupil	1.000	0.301

Figure 1: Definitions produced by the fine-tuned GPT-2 Language model after training on 2 epochs on a dataset of 9000 definition samples.

enough to encourage the generation of analogies. This is notably a hard problem since analogies take on many different forms and structures. To overcome this challenge, the authors developed a scoring function which takes into account and number of different factors.

1. After the model produces a definition for a given word, the definition is passed to the Datamuse API which returns 25 guess words which represent Datamuse's best guess (G_B) as to what the original word is based on the definition.
2. Computing the average of the most confident guess (G_1) and the best guess (G_B) and scaling it by a factor of 10.
3. Reward the model for using words that often appear in analogies [like, such as, similar to,]. This count (C_a) is scaled by a factor of 3
4. Penalizing the model for using synonyms and the original word within the definition. This is done by counting the occurrences of the top 5 closest synonyms for each word in its respective definition (C_s). This quantity is scaled by a factor of 2.
5. Penalize the model based on the length (L) of the definitions produced.

The main goal with the scoring function is to encourage the emergence of analogies rather than enforcing their structure from the onset.

Theoretically, the argument for the approach taken in this paper is that an analogy can be viewed as an abstract definition; if the degree of abstraction can be balanced with maintaining the original meaning of the definition, this could provide a good candidate for an analogy. Thus, the goal of the scoring function is to capture and reward this nebulous relationship between definition and analogy.

Practically, this scoring function can take on many different forms considering a myriad of factors. For instance, modulating the relative magnitude of each factor, including the potential for non-linear scaling functions, could have a meaningful effect on the outcome of the model. Additionally, while the parameters for generating a definition requires an output length of 50 tokens, most often the definition produced by the model was shorter than 50 tokens with the remaining space padded out to reach the required length. As a result, forcing the model to produce definitions of a specific length can prove to be beneficial.

For this project, there appears to be an ideal definition length of around 30 tokens. After this, the definitions become long-winded or irrelevant, and anything shorter than this can be extremely hard to interpret. To this end, the authors evaluated the performance of numerous scoring functions as follows:

$$S_1 = 10(G_B + G_1) \tag{1}$$

$$S_2 = 10\frac{G_B + G_1}{2} - 2C_s + 3C_a \tag{2}$$

$$S_3 = 10\frac{G_B + G_1}{2} - 2C_s + 3C_a - \left|\frac{L-30}{5}\right| \tag{3}$$

(The performance of S_1 and S_2 are highlighted in fig 3. (a)+(b) and (c)+(d) respectively.)

One of the challenges of a complex reward function is that it requires significant time and computational resources in order to improve. A complex reward function creates an expansive search space which is difficult to navigate and optimize with relatively few training samples and iterations. This is a challenge across reinforcement learning and can be seen in reinforcement learning for video games.

Training

The method described in this paper uses a multi-part training and optimization process. First the pre-trained GPT-2 model is fine-tuned on a dictionary dataset where the model is tasked with producing definitions given an word. The second, uses a reinforcement learning approach to further

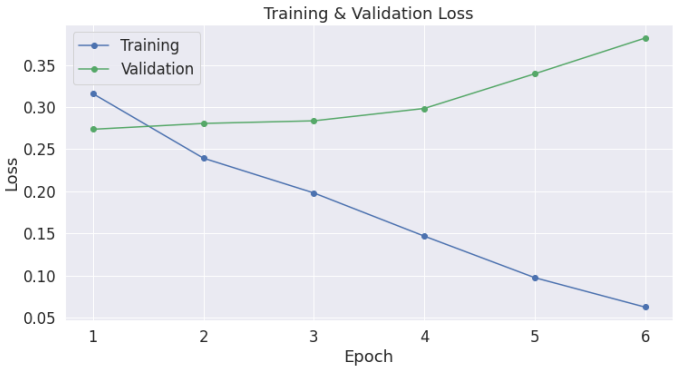


Figure 2: Fine-tuning of the GPT-2 model

train the already fine-tuned model to encourage it to produce definitions which exhibit analogical properties. Overall, the language model was trained for approximately 7 hours on a Nvidia Tesla V100 GPU provided by Google Colab.

Fine-tuning Part 1: Huggingface

As discussed in the model section of this paper, the GPT-2 model has been fine-tuned using the Huggingface transformer library on an English dictionary dataset: a set of English words along with their corresponding definitions. The Webster's 1913 English dictionary is used as the fine-tuning dataset. The entire dictionary contains over $50,000$ and therefore the first $10,000$ words were sampled for training due to training time limitations. A $90:10$ training-validation split has been taken. The model was trained for 6 epochs with training and validation error illustrated in fig 2. The model has been fine-tuned for 6 epochs, with each epoch taking 23 minutes. From the train and validation loss shown in fig 2., there is some apparent over-fitting taking place since the train loss continues to decrease while the validation error increases.

Models can overfit their data for a number of reasons. Since the model was trained on a subset of the overall dictionary vocabulary, it is possible that the model was simply remembering the definitions of each of the words it had been trained on during each epoch and as a result, was able to accurately reconstruct those definitions. However, on the validation set, the model performed worse since it was used to generating definitions based on only the seen examples. A strategy to limit overfitting in this case would be to limit the number of training epochs or to use a larger definition dataset. Since training the model on only 20% of the dataset required 3 hours, training the model for any longer would typically be met with issues of Google Colab timing out. As a result, the training time for this model was limited.

For intermediate data, a sample of word-definition pairs was passed to a reverse dictionary (Datamuse api) to evaluate the the performance of the model in producing definitions. The reverse dictionary produces a set of words which serve as *guesses* for the original target word. The authors then used the cosine similarity metric to determine the similarity between each *guess* and the corresponding ground truth. Select results are highlighted in figure 1.

For example, the first analogy produced by the model for the target word 'chemistry', for the definition it produced, it did not contain the words 'chemistry' or 'science' yet the best guess 'science' was remarkable close.

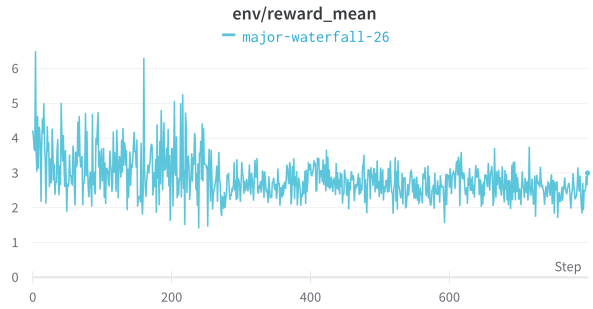
Further consideration can be placed on the use of cosine similarly (implemented through the SpaCy library) as a measure of word similarity. Specifically because based on the results from the figure 2. 'germany' and 'congo eel' are more closely related than 'biology' and 'philosophy'.

Fine-tuning Part 2: Reinforcement Learning

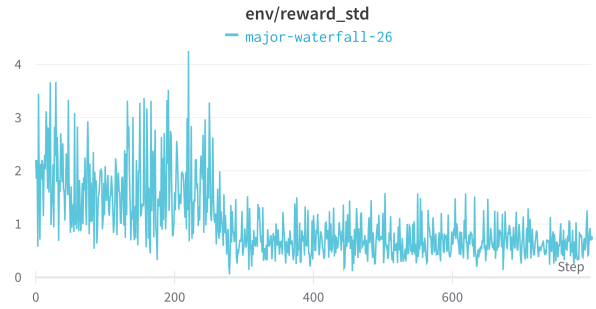
The reinforcement learning architecture used in this project is based on the transformer reinforcement learning (trl) method available on GitHub. The library provides a wrapper which allows for Proximal Policy Optimization (PPO) to be used in improving the performance of the langauge model. PPO is typically and ideal choice for reinforcement learning since it produces good results with minimal fine-tuning. Unfortunately, as discussed later, the method in this project wasn't able to realize those benefits.

During the reinforcement learning process, the authors tried a number of different scoring functions and datasets in order to evaluate for the best performance. The results of training using scroring functions S_1 and S_2 on the 1000 common English nouns dataset are shown in figure 3. While training using both of these different strategies, unfortunately, the mean reward did not seem to increase with each batch and epoch. This could be for a number of reasons, for example the scoring function was either too complex or there was not enough training to notice any improvements (for instance, say after 100 epochs instead of 4). More data typically also helps. Second, fine-tuning the hyper-parameters for the proximal policy update method could have yielded better results.

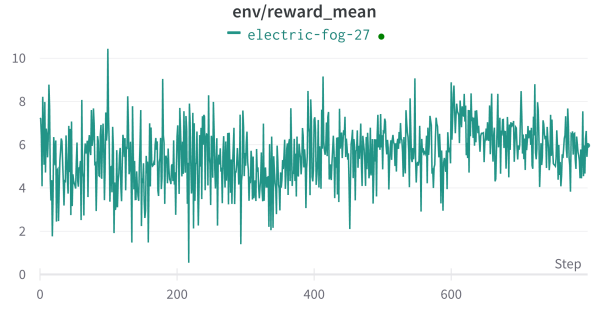
Additionally, a modified dataset was used which included $n = 5$ duplicates for each word in the nouns dataset and the set of duplicates was used to produce multiple definitions for each word. Then, each definition was scored and the results were used to update the model. Intuitively, this seems as a favorable approach since the model gets multiple tries



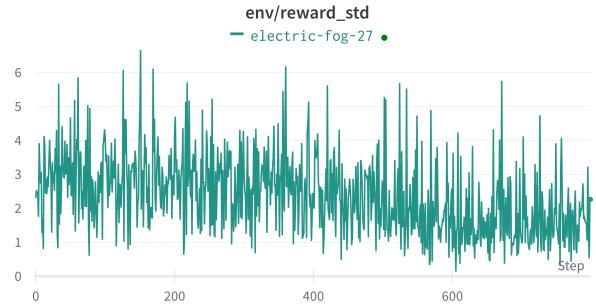
(a) Model trained using complex scoring function described in Training section part 2. Rewards converge but do not increase.



(b) Model trained using complex scoring function described in Training section part 2.



(c) Model trained using simple scoring function described in Training section part 2. Rewards increase slightly but do not converge.



(d) Model trained using simple scoring function described in Training section part 2.

Figure 3: Average reward per batch for the reinforcement learning protocol

to produce a good definition and then is updated in the direction of that definition. This could help improve the models performance considering the complexity of natural language and analogies. For this duplicated dataset training attempt, scoring function S_3 was used. Select results are highlighted in fig. 5.

Final Results

The initial goal of this project was to produce analogies by proxy; where definitions are generated with increasing levels of abstractness.

Figure 4 highlights some of the analogies that were produced after reinforcement learning. While these anaologies are not perfect, they do contain the seeds of what appear to be analogies. For example, the analogy of the word *project: A view of a future state or condition; esp., a future state of being; a future event; a future event* does seem to explain the concept of *projecting* or *projection* without directly using the word or close synonyms. Something similar can be said for *house* and *news: a publication which appears regularly every day*.

While these results may be impressive, they were usually overshadowed by dozens of less impressive, or nonsense results. This ca be attributed to a variety of reasons.

The GPT-2 model produces so much variability as a result of the diverse and disparate dataset it was trained on. As a result, it is difficult to train and coax the model into producing structured analogical definitions. If the parameters are set so the model has limited variability, the definitions produced are not abstract enough to be considered an analogy. However, if the parameters are relaxed and the model produces text with more variability, the resulting output can vary wildly from one iteration to another, as shown in [insert excerpt below].

Additionally, Without knowing how the reverse dictionary operates and handles misspelled words, it is next to impossible to structure an appropriate scoring function. For instance, since the model is penalized for using synonyms in the definition, the model can intentionally misspell the synonyms in the definition so that it is not penalized for using them but the reverse dictionary might still interpret the misspelled synonyms correctly.

Additionally, the team faced challenges with managing the memory used by the GPU during the training process. Memory issues prevented the authors from test additional parameters and setups for the overall system.

Conclusion

Overall, the results of this project were definitely interesting albeit mixed. While the model was able to occasionally

Word	Analogy	Score	Top guesses
house	A structure intended for habitation; esp., a building used by a layman or sculptor; esp., one such a building used as an exhibition of love or friendship.	17.04	house, dwelling residence, home
school	A place for pupils to receive instruction in a reading or reading at a reading table; a reading or spelling place.	14.09	classroom, schoolhouse, school
point	In a direct manner, and as if with a direct course, as a horse or a tree. Take the points round and share the wood to thy brother. Shak. The points round did not bear The loss. Addison. The points round did	11.13	compass, straight, right, directly
project	A view of a future state or condition; esp., a future state of being; a future event; a future event. His eye shall be as varied as the next leaf of the bele. Sir T. More.To think;	10.84	think, consider, contemplate, reflect
news	A publication which appears regularly every day; as, the News of God; the News of religious houses. [R.] May we not have as news of religion or of religion as of religion, or of the quality of being religious;	9.88	newspaper, journal, periodical
top	The part of a ship where the sail is first hoisted and trimmed so as to give a sail sail edge. Ham. Nav. Encyc.	9.02	halyard
structure	1. That which is fixed; a stipulation; a stipulation; a part or agency to limit the quantity or capacity of; as, the stipulation of a jury in their trial. 2. That which serves; an agent to complete a	3.74	condition
heart	That part of the alimentary canal between the stomach and the pharynx; the alimentary canal.	2.8	esophagus, gullet, throat, pharynx

Figure 4: Definitions produced by after reinforcement learning on GPT-2 Language model on duplicated dataset and scoring function S_3.

produce analogies that appeared human in nature, most often than not, the analogies were nonsense. Continued research using larger datasets and more scoring and training parameters would be of interest.

Repo Link

Google Colab.

References

[Hope et al. 2017] Hope, T.; Chan, J.; Kittur, A.; and Shahaf, D. 2017. Accelerating innovation through analogy mining. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 235–243.

[Lofi, El Maarry, and Balke 2013] Lofi, C.; El Maarry, K.; and Balke, W.-T. 2013. Skyline queries in crowd-enabled databases. In *Proceedings of the 16th International Conference on Extending Database Technology*, 465–476.

[Turney 2005] Turney, P. D. 2005. Measuring semantic similarity by latent relational analysis. *arXiv preprint cs/0508053*.

[Ushio et al. 2021] Ushio, A.; Espinosa-Anke, L.; Schockaert, S.; and Camacho-Collados, J. 2021. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? In *ACL 2020 Main Conference*.