# Modeling Generative Processing in Neural Networks to Improve Retention

Madelyn A Scandlen
GTID: 903271843
mscandlen3@gatech.edu

Megha Sharma
GTID: 903842895
meghasharma@gatech.edu

Reshma Anugundanahalli Ramachandra
GTID: 903815125
reshmaram@gatech.edu

Varshith Sreeramdass
GTID: 903571453
vsreeramdass@gatech.edu

CS-8803 HML Humans and Machine Learning

Final Course Project Report

16th December, 2022

## Abstract

The paradigm of proceduralism has addressed how perceptual features are processed in the mind for encoding and retrieval. This includes several empirical accounts in the form of generation effect, levels of processing, transfer-appropriate processing, and testing. Generative processing generates predictions based on how humans generate perceptual features that are associated with previously learned concepts. However, this does not provide a thorough explanation of the underlying mechanism. Deep learning research on generative processing has largely focused on computer vision tasks using surrounding image context to conditionally generate predictions and using the generated output to update a latent variable to improve the model. Motivated to connect these modern frameworks to cognition in the natural language domain, we propose a framework for conditional generative training for text. We do this by fine-tuning deep language models that learn to predict tokens based on contexts. Using the proposed training methodology, we compare the benefits of generative processing with those observed in human learning using the experimental setup from (Slamecka & Graf, 1978). We find that masked language models do not provide a plausible model of proceduralism in humans. We also perform additional experiments on a sentiment classification task with an aim to conduct comparative human-user studies in the future.

## Introduction

Proceduralism, in cognitive science, proposes that for a perceptual experience, memory is encoded in the same units that processed it in the first place. Among the theories that explore this argument, those like levels-of-processing from (Craik & Lockhart, 1972) or generation effect from (Slamecka & Graf, 1978) have enjoyed widespread investigation and significance. In deep learning literature, (Van der Ven et al., 2020) was inspired by replay of memories in the human brain and applied the idea of generative replay to mitigate catastrophic interference in neural networks. This inspiration has also provided, on occasion, even superhuman performance in natural language applications such as BERT from (Devlin et al., 2019). However, none of these models offer any insights into explaining the nature or details of the cognitive processes in the human mind.

We propose that by constructing deep learning models and comparing the potency of phenomena such as the generation effect with humans, we can argue which of the models, if at all, provide a plausible account of the underlying cognitive mechanisms. Such methodology formulates a method for evaluating competing theories and unraveling features of cognitive processes systematically, ultimately providing a complete mechanistic account. To the best of our knowledge, we are the first to perform a detailed comparative analysis between humans and natural language models for the generation effect. We find that BERT, while offering significant benefits in natural language tasks, does not provide a plausible account of proceduralism in the human mind.

## Related Works

### I. Cognitive Science

Proceduralism has been adopted in the field of cognitive science because of its accordance with many theories of memory learning. (Craik & Lockhart, 1972) developed the *levels-of-processing* theory as one of the first to introduce the idea that encoding of a stimulus can factor into the cognitive process of retrieval. How a person processes a stimulus such as a word can allow it to be encoded in short-term store (STS) or long-term store (LTS), where the stimuli in STS are more quickly displaced by new inputs and the stimuli in LTS can be better retrieved later. For an input such as a word, an acoustic encoding would send the input to STS, whereas semantic encoding would send the input to LTS, and the words processed on the deeper semantic level were more retrievable at later time periods. (Craik & Tulving, 1975) found that retrieval greatly varied when subjects were prompted to encode words as a function of the levels of processing.

(Slamecka & Graf, 1978) developed the theory of the generation effect from the earlier theories of proceduralism. Generating a word proved to encode that word for better retrieval than simply reading it. Subjects either read two words with a semantic connection, or subjects were given the first word, the semantic connection, and the first letter of the second word and were asked to generate the second word. In five experiments targeting the generation effect, words that were generated by the subject under a conditional stimulus, the semantic rule, were more often recognized at a later time. This theory is still popular in learning practices today as flash cards or fill-in-the-blank problems.

## II. Deep Learning

The concepts from proceduralism have been incorporated in new research in machine learning, especially as deep learning grows. Since encoding a stimulus matters for the cognitive process of retrieval, encoding of data as more appropriate inputs to a neural network affects outputs. Neural networks use an embedding layer as the first hidden layer of the architecture, transforming the input to send to deeper layers, while also being learned and updated as the model trains.

Neural networks suffer from a problem that the human brain does not: the stability-plasticity dilemma, where networks are in conflict to be plastic to adapt to new information and to be stable to retain old information (Carpenter & Grossberg, 1988). Catastrophic forgetting is the phenomenon where neural networks are unable to retain old information for old tasks upon learning new tasks, undesirable because it limits the expansion networks as well as unrealistic because human brains do not face the issue.

A generation task called generative replay has been proposed as a solution to catastrophic forgetting. Replay of old information as inputs in a neural network is a plausible implementation of the reactivation of neuronal activity patterns in the human brain when memories are retrieved (Robins, 1995). However, neural networks are limited by memory storage, so retraining on old information and old tasks is unmanageable. (Van der Ven et al., 2020) propose generative replay, where the model generates examples for replay to bypass the issue of memory storage while shifting the problem to the generator model. In the domain of computer vision, an artificial neural network can easily distinguish between all MNIST digits. Difficulty appears when training the neural network to distinguish between all digits while only observing two classes at a time. Van der Ven et al. find that generating and replaying low quality examples of previously learned classes was enough for the model to be competitive at the new task.

Generative replay was used as inspiration for a new architecture for artificial neural networks, the generative adversarial network (GAN) which utilizes a minimax approach where a generator model produces examples and a discriminator model learns the examples and reinforces the generator to improve generation. When including generation for reinforcement learning, the problem shifts to creating an adept generator. Conditional generation, using a conditioning input to guide generation and classification, has been successful in computer vision tasks, but text generation has been more difficult as text is sequential and discrete. (Mirza & Osindero, 2014) use images as contexts and generate textual tags to describe the images using a conditional GAN, where the generated text is required to be valid words in the vocabulary.

Because there is a limited set of valid words that could be both in the conditioning context or be generated by the model, word embeddings are necessary in generation to properly capture semantic information. Pre-trained language models are useful for language tasks as deep embeddings are learned over large amounts of data. BERT is a pre-trained masked language model which masks a word in a textual sequence and leverages the surrounding context in both directions to learn an embedding and predict the masked word (Devlin et al., 2019). Incorporating pre-trained language models to generation tasks has improved generation with deeper and richer embeddings.

## Methodology: Evaluating Predictive Models

It is vital to note that the generation effect is a phenomenon of memory, not reasoning. It argues about improved retention of the perceived inputs, and not necessarily more robust higher order reasoning on those perceived inputs. On the other hand, modern predictive models such as neural networks are built to capture patterns between inputs and outputs. At least, those without an explicit accommodation for memory such as Feed Forward Networks or RNNs function purely as reasoning engines. There also exist models such as Variational Autoencoders that construct a latent space that captures the distribution of inputs, but the notion of memory is not definite or episodic. The closest analogues are Hopfield Networks, but it is unclear if they have the capacity to perform complex reasoning required in natural language domains.

So, how should we evaluate a predictive model (a reasoning framework), for instance, for recognition performance in (Slamecka & Graf, 1978) (a memory test)? To answer this, we pose the predictive model a reasoning task and argue that if the model can perform well, it has “remembered” the inputs correctly. This argument is analogous to the idea that high-capacity neural network models “store” the dataset in their weights. In essence, we extend the evaluation of the generation effect, from memory to higher order reasoning that operates on memory.

Let us now consider how we can extend the generation effect to training predictive models. In (Slamecka & Graf, 1978), subjects are exposed to pairs of words (e.g., crow, bird) that are related by a certain rule (e.g., example). Those in the treatment group are exposed to a pair where the second word is partially blanked out (e.g., crow, b___), prompting the subjects to “generate” the missing letters. This generation causes better encoding of the stimulus in memory. Whereas, with predictive models, without an explicit change in their parameters (for instance, through backpropagation), we do not have a provision of altering the encoding mechanisms. Therefore, we explicitly train our models to generate the missing stimulus.

With the two ideas in place, we present our complete pipeline. We have a base model appropriate for a domain. We choose those architectures that have provision for encoding parts of the input, as well as perform task specific reasoning. In our case, we target natural language tasks with the model BERT that provides contextual word embeddings. Using this base model, we optionally perform generative training to alter encoding of the inputs (e.g., word pairs related by a rule, sentences) for the target domain. Following this, we perform task-specific training for the target task at hand (e.g., rule prediction, sentiment classification). Finally, the model is evaluated on the task.

To correctly argue about the effect of generative training with predictive models, it is important that objectives for generative training and task specific training are unrelated. For instance, consider a setting where we attempt to naively replicate “cued recognition” from (Slamecka & Graf, 1978). Generative training, as discussed previously, would train the model to output the second word in the pair. With task-specific training/evaluation, the model would predict the second word, given the first (cue) word. In this scenario, because of generative training, the model would learn to capture correlations between words in a pair. During evaluation, we expect the model to choose those target words that are related to the cue words in a manner similar to word pairs observed during generative training. This does not capture any notion of better memory of those pairs.

## Experiments: Slamecka & Graf

### I. Details of experiments from Slamecka & Graf (1978)

In the first experiment detailed in (Slamecka & Graf, 1978), about 24 introductory psychology students were tested individually for two cases: Generate and Read. For the Generate condition, 20 cue cards per rule with a stimulus and the first letter of the response for that particular rule were shown to the subjects, e.g., *rapid-f* for Synonym. For the Read condition, the cards contained both the stimulus and the response, e.g., *rapid-fast* for Synonym. Five such rules were tested, Category, Associate, Synonym, Antonym, and Rhyme. The subjects were informed of the rule before looking at the 20 cue cards corresponding to that rule and the process was repeated for all five rules. The stimulus-response word pairs were also meant to be uttered out aloud once. A recognition test was then conducted by providing the all subjects with a test sheet with 100 reordered sets of three alternatives where each set consisted of a target response and two extra words added. The subjects were also informed, in prior, about the recognition test. The subjects were asked to circle the right response word for each stimulus by following a fixed direction. No skipping or retracing was allowed. The subjects also rated their confidence levels (ranging from 1 or low to 5 or high confidence) for each choice made.

## II. Dataset

The dataset consisted of norms for the five rules from (Slamecka & Graf, 1978). The Association norms were derived from Appendix A of the Association Norms database (Nelson et al., 1998). About 721k (Stimulus, Response) pairs were extracted. For the Category rule,  about 2k norms were derived from (Overschelde et al., 2004). The list of synonyms (500k) and antonyms (6k) were derived from WordNet. Finally, for the Rhymes rule, the rhyming pairs were obtained by extracting words which had matching three or more ending letters from Appendix B of the Association Norms database (Nelson et al., 1998). About 5k rhyming pairs were obtained. Since the dataset was skewed in terms of number of entries in each rule category, downsampling was performed to balance out the dataset by discarding irrelevant data. Entries with only a word rather than a phrase were considered and about 400 (Stimulus, Response) pairs were randomly sampled from each rule category to form the final dataset.

## III. Input encoding to BERT

The input to the language model BERT is encoded in a slightly different manner when compared to the (Slamecka & Graf, 1978) experiments. Instead of predicting a response to the given stimulus based on the rule, the neural network can be trained to learn the relationship between a given pair of words i.e. if the two words are related by association, represent categories, are synonyms or antonyms of each other or rhyming pairs. However, instead of generating the relationship between the two words as output, the training task remains to learn to generate the response. Essentially, by generating the second word, the network is able to better reason about the relationship.

Thus, to model this approach, for training, the word pair, e.g., *apple-fruit* is encoded as "apple is ________ of fruit" where the blank is filled in by the rule that the word pair falls under (which for this example would be Category). For the control group, the language model (BERT) is not modified. For the generation group, on the other hand, a fine-tuned version of BERT is used to predict the last token, e.g., "apple is example of _____" → "fruit". For testing the predictive model, a stimulus-response pair is given as input and the relationship between the words is predicted, e.g., "apple is ______ of fruit" → "example".

**Table 1. Examples for train and test input encoding**

| ***Rule*** | **Train Task** | **Evaluate Task** |
|---|---|---|
| *associate* | surgery is *associate* of [MASK] | surgery is [MASK] with hospital |
| *category* | tree is *example* of [MASK] | tree is [MASK] of christmas |
| *antonym* | far is *opposite* of [MASK] | far is [MASK] of near |
| *synonym* | refer is *synonym* of [MASK] | refer is [MASK] of cite |
| *rhyme* | prick is *rhyme* of [MASK] | prick is [MASK] of slick |

If in the testing phase, the network's relationship prediction accuracy is good, it means that it can "reason" well for the items in our dataset. If it is higher for the generation group as compared to the control group, then we conclude that by learning to generate the target word from the relationship, it has memorized the relationship between them better.

## Results

Training loss decreased over the 100 training epochs, showing that appending the predicted examples to the generative model improved performance at the evaluation task of determining the rule. This loss can be seen below in Figure 2.

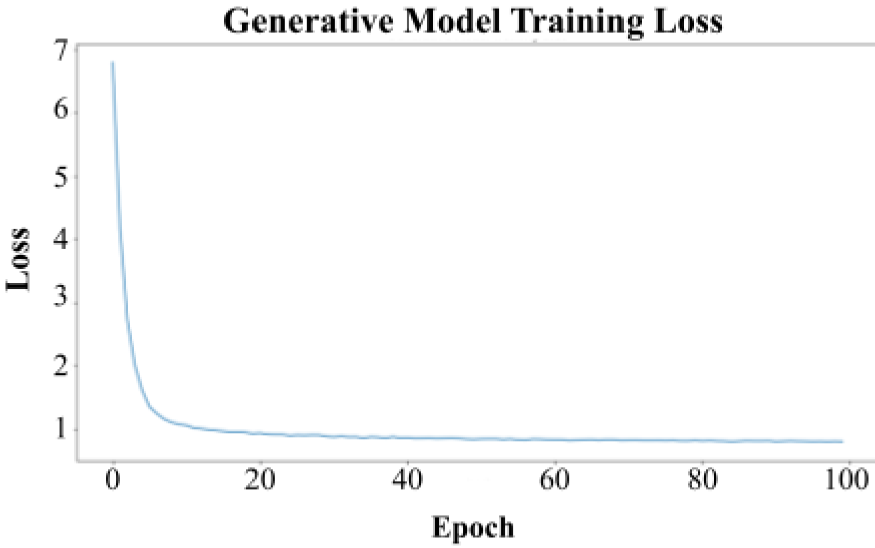

**Figure 2. Plot of training loss vs. number of epochs for the generative model training process.**

A balanced test set of 443 examples consisting of 88 or 89 examples from each rule was withheld for evaluation of the reading-only language model and the generative language model. Only the evaluation task of predicting the relationship was performed on the test set for both models. The generative model achieved a 0.22577 accuracy of predicting the correct rule on the unseen data, compared to 0.18738 accuracy of the control model. The class-wise accuracies of the two models are shown in Figure 3 (left). The generative performed best at predicting associate, category, and rhyme, but failed to improve on the control model for synonym and antonym.

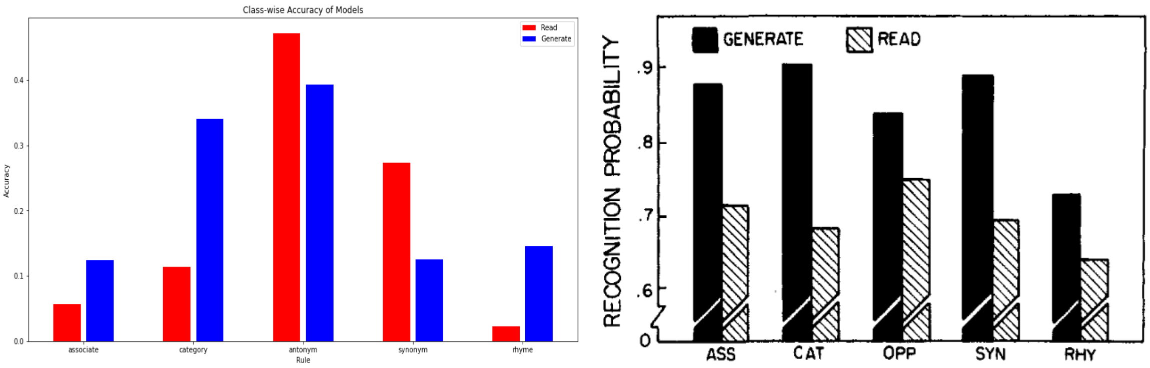

**Figure 3. The graph on the left shows the accuracy of rule prediction of our models. The graph on the right shows the recognition probability of words under each rule.**

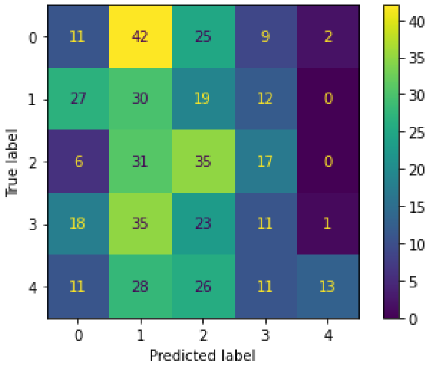

**Figure 4. Confusion matrix of the rule prediction by our models.**

From the confusion matrix shown in Figure 4, antonym rules were most often predicted as associate and synonym rules were most often predicted as category. A McNemar test was performed class-wise between the two models, finding a statistically significant difference of the control model and generative model at p<0.005 for all classes, shown in Table 2.

**Table 2. Class-wise accuracy for each model with the McNemar p-value shows that the distributions of the model for each class were different.**

| ***Accuracy*** | *associate* | *category* | *antonym* | *synonym* | *rhyme* |
|---|---|---|---|---|---|
| **Reading-only** | 0.05618 | 0.11364 | 0.47191 | 0.27273 | 0.02247 |
| **Generative** | 0.12360 | 0.34091 | 0.39326 | 0.12500 | 0.14607 |
| **Significance Threshold** | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | $p < 0.005$ |

The results demonstrate that for three of the five rules, augmenting generations shows improvement on predicting the rule for unseen data. The model is learning more richly the context of the rule and updating the embedding during the training phase when generating a word to match the given rule. The decrease in performance for antonym and synonym pairs may be due to the semantic nature of the examples, where the antonym and synonym word pairs may not be closely-related enough but can still be associated with each other. The generative model did not achieve good accuracy on class prediction, with only two classes able to be correctly predicted more than by random chance.

The obtained results for our methodology are dissimilar to the experimental results from (Slamecka & Graf, 1978) (as shown side-by-side in Figure 3). It should be noted that rule-prediction accuracy and recognition probability are not strict analogues. However, given that we evaluate the generation effect as a reasoning phenomenon (as explained under Sec. Methodology), rule-prediction accuracy is the closest available analogue. For rules *associate*, *category* and *rhyme*, with our methodology, we observe a 10%, 25%, 15% points improvement between read and generate conditions resp. Whereas, with human subjects, the improvement is 15%, 20% and 10%. While our model actually becomes worse for *synonyms* and *antonyms*, human subjects observe the least amount of improvement in *antonyms*. Overall, the improvements of our model do not capture the improvements observed in human subjects.

## Experiments and Results: Corpus of Linguistic Acceptability (CoLA)

### I. Dataset

To test the effect of generative processes on a more complex task, the architecture was modified for testing on the CoLA dataset (Warstadt et al., 2019) to classify whether the sentence is grammatically correct or not. The dataset includes both raw and tokenized versions, from which the raw data was used for the experiment. The data used consists of the following files:

a. **In_domain_train.tsv:** 8.5k (sentence, labels) tuples for training

b. **In_domain_dev:** 527 tuples for development/validation

**c. Out_of_domain_dev:** 516 tuples for testing purposes

**II. Methodology:**

1. *Experiment 1:* Train in_domain_train.csv on the generative processing part and use out_of_domain.tsv to test on the classification pipeline. Through this experiment we want to understand whether generative processing on the language model leads to performance improvement for a related task (classification) on a different holdout dataset.
2. *Experiment 2:* Train in_domain_dev on the generative processing part and use out_of_domain.tsv to test it on the classification pipeline. This experiment aims to gauge whether generative processing with a small dataset can lead to improvement in performance on a related task on the holdout dataset.

**III. Results:**

In both experiments it was observed that generative processing does not offer marked improvement in the performance of the classification pipeline when compared to training directly on the complete dataset. Sometimes it even performed worse.

## Conclusion and Future directions

From our results, it can be seen that generative processing does not offer improved accuracy across rules. Additionally, the obtained improvements are not similar to those seen in humans. We conclude that masked language models do not provide a plausible mechanistic account of proceduralism in the human mind.

We also tested the effect of generative processing on popular abstract natural language tasks such as Grammar Verification but it was observed that generative processing failed to induce a marked difference when tested across tasks. It was observed that it still needed to be trained on the classification task in order to produce an improved accuracy. A possible explanation for this result could be that the architecture of the language model used for the experiments i.e. BERT already uses an internal masking and generation method to achieve its state of the art performance. Therefore meaningful results might still be achievable for this application with a different architecture.

A positive result to this study can have many implications. If generative processing possibly improves classification accuracy, it would mean that a generative model could be used to classify noisy data as well! With most of the real-world data available today being noisy, this would save us a significant amount of cost, time and effort that would have otherwise gone into data preprocessing. However, in order to study the similarity between the generation effect on humans and neural networks on tasks like these, some substantial data derived from human user studies is necessary, which can be another possible future direction.

**References**

(Slamecka & Graf, 1978) Slamecka, N. J., & Graf, P. (1978). The generation effect: Delineation of a phenomenon. *Journal of experimental Psychology: Human learning and Memory*, *4*(6), 592.

(Craik & Lockhart, 1972) Craik, F.I.M., & Lockhart, R.S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior*, *11*, 671–684.

(Van der Ven et al., 2020) van de Ven, G. M., Siegelmann, H. T., & Tolias, A. S. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, *11*(1), 1-14.

(Devlin et al., 2019) Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv, abs/1810.04805*.

(Craik & Tulving, 1975) Craik, F.I.M., & Tulving, E. (1975). Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology: General*, *104*, 268–294.

(Carpenter & Grossberg, 1988) Carpenter, G.A. & Grossberg, S. (1988) The ART of adaptive pattern recognition by a self-organising neural network. *Computer,* **21,** 77± 88.

(Robins, 1995) Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, *7*(2), 123-146.

(Mirza & Osindero, 2014) Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

(Nelson et al., 1998) Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). The University of South Florida word association, rhyme, and word fragment norms. http://www.usf.edu/FreeAssociation/.

(Overschelde et al., 2004) Van Overschelde, J. P., Rawson, K. A., & Dunlosky, J. (2004). Category norms: An updated and expanded version of the norms. Journal of memory and language, 50(3), 289-335.

(Warstadt et al., 2019) Warstadt, A., Singh, A., & Bowman, S. R. (2019). Cola: The corpus of linguistic acceptability (with added annotations).

Sukhov, S., Leontev, M., Miheev, A., & Sviatov, K. (2020). Prevention of catastrophic interference and imposing active forgetting with generative methods. Neurocomputing, 400, 73-85.

Iii, H. L. R., Gallo, D. A., & Geraci, L. (2002). Processing approaches to cognition: The impetus from the levels-of-processing framework. *Memory*, *10*(5-6), 319-332.

Friston, K., Kilner, J., & Harrison, L. (2006). A free energy principle for the brain. *Journal of physiology-Paris*, *100*(1-3), 70-87.

Heilbron, M., Armeni, K., Schoffelen, J. M., Hagoort, P., & De Lange, F. P. (2022). A hierarchy of linguistic predictions during natural language comprehension. Proceedings of the National Academy of Sciences, 119(32).

Jaeger, T. F. (2010). Redundancy and reduction: Speakers manage syntactic information density. Cognitive psychology, 61(1), 23-62.

Fellbaum, C. (2010). WordNet. In Theory and applications of ontology: computer applications (pp. 231-243). Springer, Dordrecht.