Probing Frozen NL Models for Alignment with Human Reasoning

Stanford CS224N Custom Project

Clara MacAvoy
Department of Computer Science
Stanford University
cmacavoy@stanford.edu

Claire (Ge) Cheng
Department of Biomedical Informatics
Stanford University
gecheng@stanford.edu

Abstract

As large language models proliferate through society, it is increasingly desirable to be able to understand how these models align with traditional human reasoning. We propose a framework for probing the frozen hidden states of previously trained language models to generate natural language explanations of their reasoning, allowing us to measure alignment between model and human logic. In this work, we train an LSTM to generate natural language explanations for the SNLI task (Bowman et al., 2015) using the hidden layers of previously trained language models. This framework serves as an analysis tool for determining the ability to extract information encoded in model hidden states about internal model reasoning. Our analysis concludes that these SNLI-trained models have only a weak alignment with human reasoning, with our probing unable to fully recover sufficient information to generate coherent explanations of reasoning.

1 Key Information to include

- Mentor: Jesse Mu (muj@stanford.edu)
- External Collaborators (if you have any): NA
- Sharing project: NA

2 Introduction

This work's contribution to the field is analysing the ability to extract information about reasoning from the hidden states of an already frozen model. Previous work in the eSNLI paper (Camburu et al., 2018) has explored the task of explanation generation for the SNLI dataset, but trained a model for the task of explanation and label prediction given the input phrases without exploring the task of extracting knowledge from a previously trained model. Therefore, this paper establishes a baseline for probing LM models to extract interpretable information about their internal logical states.

Model interpretability is one of the most exciting of modern AI research, with it increasingly important to be able to not just have high task performance but also be able to understand why a model acts in the way that it does. As language models become more and more widely spread within society, so also grows the motivation to be able to evaluate if these language models contain the logic necessary for proper generation of knowledge in line with human thought. Probing an already trained model is a powerful tool for extracting encoded knowledge about how the model works, and in turn can be used in order to develop an understanding of the internal logic being applied by a model. We focus our efforts on investigating models trained on the Stanford Natural Language Inference (SNLI) task (Bowman et al., 2015). The SNLI task consists of pairs of premise and hypothesis statements which are labeled with their logical relationship (entailment, contradiction or neutral). This task has a high accuracy rate, with current state of the art models achieving over 93% accuracy (PapersWithCode).

However, research has shown that this accuracy is often achieved through the discovery of spurious correlations rather than through the logical reasoning (Nie et al., 2020). Therefore, we felt as though this task was an ideal space to apply our probing model in order to investigate the alignment between the internal logic of models trained for the SNLI task and human reasoning.

The eSNLI dataset (Camburu et al., 2018) expands SNLI and consists of pairs of premise and hypothesis sentences labeled based on their logical relationship, and natural language explanations for each label. The eSNLI explanations provide a baseline for human reasoning in natural language format upon which we can base our probing model outputs.

Our work focuses on using a probing model to answer two separate questions about models trained on the SNLI task to form a richer understanding of alignment of model logic with human reasoning. Our first research question pertains to the differences in model encoding between a pretrained and non-pretrained SNLI model. Our initial belief was that non pretrained models may rely more on spurious correlations in comparison to pretrained models, and we explore the differences between the information encoded in these models. A second research question is where reasoning information is encoded throughout a model, which we probe by examining the models across multiple layers.

3 Related Work

This work takes inspiration from the space of probing models when designing our system for understanding alignment with human reasoning. Previous work in this space has explored the extent to which probing can be used as a tool to extract information from models, although probing work has not previously been focused on the field of human reasoning alignment. (Adi et al., 2016) examined the ability to probe different text models for information about inputs to the model, and this work found that the hidden states of a variety of models could be successfully probed to recover information about inputs to the model. This work inspired us to consider different possible approaches to the design of our model, such as probing different embedding dimensions, which was found in the paper to encode different levels of information.

Probing models have been used for extracting knowledge from NL models in the past, such as the work of Hewitt and Manning (2019) which was able to probe models for syntax trees. Additional work in the field of probing which was informative to our work was Hewitt and Liang (2019), which introduced control tasks for probing. These are tasks for which a model with high selectivity will achieve a low performance while still having a high accuracy on probing true tasks. Hewitt and Liang (2019) establishes that models with high selectivity are desirable because this is a strong indication of true knowledge being uncovered by a probing model. From this work also flows that simple models such as LSTMs can have higher selectivity than more complicated models frequently used in probing which have low selectivity. This field of probing NL models is further supplemented by the work of Liu et al. (2019) which implements techniques of training a model on top of a previously frozen NL model, and in which it was found to be possible to train such a probing model to complete a variety of NL tasks using information encoded in hidden states. This work also reveals the marked differences in where and how knowledge is encoded across layers of frozen models.

Prior research in the space of the SNLI dataset informs our research by helping us to understand the SNLI task and how models are trained for this task. The work of Camburu et al. (2018) attempted to broaden approaches to the SNLI dataset by examining not only correct labeling of sentence pairs but also the human language generation of logical explanations for labels. This work inspires us to think about the ways in which humans use language to describe logical reasoning, and in turn provide a baseline for our probing models to attempt to generate similar language.

4 Approach

4.1 Probing Model

Our probing model is an LSTM which is initialized with the hidden states of a frozen model trained for the SNLI task (He et al., 2020) and trains to generate logical explanations from these hidden states, applying teacher forcing on the explanations from eSNLI as true labels of correct human language reasoning. This architecture is seen in Figure 1, whereby our model takes in the frozen hidden states from a model and then uses a LSTM to translate those hidden states into a natural

language explanation of logic. The probing model concludes with a fully-connected layer to return logits of tokens. An LSTM (Hochreiter and Schmidhuber, 1997) is an appropriate model for this task because it includes feedback connections, which are well suited for decoding a sequential explanation, as is necessary for our goal of natural language representations of internal logic. [1]

We probed two different models to answer our research question about SNLI models. The first was a pre-trained model, cross-encoder deBerTa-v3 which had been finetuned for the SNLI task. The second was a model which we wrote that had identical architecture to cross-encoder deBerTa-v3 but without pretraining. This allowed us to make comparisons between a pretrained and non-pretrained model. We created a baseline by training our model using random tensors as the initialized state which acted as a comparison for our probing results on the models. Our model can take any of the 13 hidden layers of the two models, which allows us to compare information encoded across layers.

4.2 Analytical Tools

For analysing our model, we built a classifier which took in an explanation and predicted a label. This classifier is based on a pretrained bert transformer which we finetuned for the task of explanation classification using the eSNLI dataset. This tool allows us to analyse the extent to which correct logical reasoning is being extracted even when correct subject matter is not. This also allows us to compare how different types of information (logical vs subject matter) are being extracted, information not included in methods such as BLEU scores which rely on only n-grams present. This classifier had above 97% accuracy on the test set when trained and tested on eSNLI explanations.

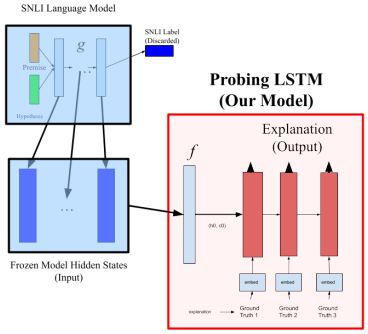


Figure 1: Probing Architecture

5 Experiments

5.1 Data

We use the open-sourced e-SNLI dataset (Camburu et al., 2018), an extension of the Stanford Natural Language Inference (SNLI) task (Bowman et al., 2015). e-SNLI includes text pairs (premises, hypotheses), entailment labels, and natural language explanations of entailment relations. The dataset consists of 549,367 training, 9,842 validation, and 9,824 testing records. We pre-processed the natural language explanations, which are used during teacher forcing, using Huggingface's GPT2Tokenizer (Wolf) prior to training, and this tokenizer is also used when decoding our generated explanations. The GPT2Tokenizer transforms explanations into lists of indices ranging from [0, 50255]. We add index "1" to indicate the beginning and "2" to mark the end of the explanations and pad the lists to be the same length using a large index "50256". We also collected frozen hidden states (Mu, 2023), averaged across their sequence length, of a language model (cross-encoder/nli-deberta-v3-base) trained on the SNLI task. These hidden states, as the input, have shape (N, L, D) where N is the number of records, L is the number of layers (baseline model has 13 layers) and D is the size of the embedding of the hidden state (768 in this case).

5.2 Evaluation method

To measure the similarity between generated and human reasoning, we evaluate based on two criteria: 1) the extent of overlap between the generated reasoning and human reasoning, and 2) the logical coherence and consistency of the generated reasoning in comparison to human reasoning. We assess the performance of our experiments utilizing established metrics within the field of text generation evaluation, such as BLEU scores, loss/perplexity. The BLEU score is calculated using 2-grams. Perplexity measures how well a probability model predicts a sample and is the exponential of cross-entropy loss. We acknowledge that the BLEU score may exhibit limitations in assessing the semantic

[1]We built our architecture from scratch with reference to online LSTM model pipeline setups (pytorch (Robertson, 2020) github: (lkulowski, 2020)). Our mentor Jesse Mu (Mu, 2023) provided the script to collect the frozen model states and also guided us in correctly setting up the model

nuances of generated phrases. Therefore, we have developed a classifier to investigate whether the explanations correspond to the appropriate label types, namely entailment, contradiction, and neutral. This evaluation method enables us to assess the probing model's capability of extracting reasoning information from hidden states, even in cases where specific subject details cannot be retrieved. Since it is a challenging task to learn the structure of language and how to predict logical relationships at the same time from the hidden states, our primary objective is not to optimize these metrics. Rather, we focus on comparing the metrics across different experimental conditions to examine the relative performance of each variant.

5.3 Experimental details

Our probing model runs on a batch size of 64 and uses the Adam optimizer and cross-entropy loss. We evaluated the model and computed metrics on the validation set every 2 epochs, with a maximum training limit of 100 epochs. To prevent over-fitting, we set an early stopping mechanism that terminates training if the BLEU score fails to improve for 10 epochs after the peak. During prediction, the model generates tokens sequentially to form an explanation, halting only when the end index is reached. In the same setup, we investigate the following variants:
Hyperparameter exploration of the probing model: We want to find a reasonable set of hyperparameters before exploring the variants and comparing the results achieved with each. Thus, we explored learning rate among {$3e^{-5}$, 0.0001, 0.0003, 0.0005, 0.001, 0.005} and LSTM input size among {192, 384, 768, 1152, 1536}.
Hidden states comparison: We want to examine the presence of reasoning information in the hidden states of the pre-trained NLI DeBerTa model. We established our own baseline by using tensors randomly generated from the normal distribution as input. We also evaluate the performance of the DeBerTa model against the results obtained from our own model that was trained from scratch. This comparative analysis enables us to determine the extent to which the DeBerTa model can capture reasoning information in its hidden states.
Input layers experiment: We are interested in learning where human aligned reasoning is located if it is present in the hidden states and we run our experiments across 13 layers to compare the explanation quality. By understanding the specific location of reasoning information within the hidden states, we can gain a better understanding of how the DeBerTa model processes and represents complex language-based reasoning tasks.
Multi-layer experiment: We posit that increasing the number of layers in a neural network could provide more diverse and informative representations that may exhibit interactive effects across multiple layers. To test this, we collapsed several layers of the input hidden states into a single layer and probed whether there were significant interactive effects across those layers.
Extended probing model structure: Considering the limited information available to the probing model, we conjecture that a more sophisticated model might be required to decode additional information and potentially improve the quality of the generated results. Therefore, we investigated a more powerful multi layer LSTM probing model
Google Flan T5 exploration: Finally, we expand our investigation by examining datasets that require multi-step reasoning to determine whether the answer can still generate predictive representations of the intermediate computational "steps". We employ the same probing strategies on the hidden states of the state-of-the-art Google Flan-T5 model (Ling et al., 2017a) that was trained using the AQUA-RAT dataset (Ling et al., 2017b).

5.4 Results

5.4.1 Hyperparameters explorations of the probing model

We conducted a series of experiments by running 5 iterations per setting in order to obtain a range of results for LSTM input size, as indicated in Figure 2. Our observations reveal that the BLEU score, despite its utility as a performance metric, exhibits lower stability and a wider range of values for the same setting. Notably, larger embedding sizes tend to yield higher BLEU scores and lower perplexity due to their capacity to encapsulate more information. In our search for the learning rate, we found that smaller rates had slow convergence but superior performance. It is worth noting that excessively small rates may result in the model becoming stuck in a suboptimal solution, without exhibiting the double descent phenomenon. Larger rates have a tendency to overshoot and may cause the model to

miss the optimal solution, as indicated by an observed increase in evaluation loss (Appendix Figure 4). Based on our experimentation, we determined that a learning rate of 0.0005 is optimal. We conducted subsequent experiments using a hidden size of 1568 and a learning rate of 0.0005 which strikes the balance between performance and stability.

5.4.2 Hidden states comparison

Results (Figure 3) from pre-trained and no-pretrained hidden states are both better than random tensors which matches our hypothesis and indicates that representations do contain extractable information about NLI tasks. Comparing pre-trained and non-pretrained representations, we observed that pre-trained representations outperformed the latter in terms of both BLEU score and classifier accuracy. This outcome is expected, given that pre-trained models begin from a point of knowledge containing language structure and/or logical relationships. We did not observe a strong correlation between the loss pattern and BLEU score/classifier accuracy. This is because even though pre-trained models may have a stronger ability to predict the correct token, the probabilities may be skewed towards certain tokens, which can lead to a higher loss.

5.4.3 Input layers experiment

Our initial hypothesis was that the layers of the neural network closer to the input data, would produce better results. However, during our experiments, we observed that both pre-trained and non-pretrained models showed better performance metrics (Figure 3) in the middle to output layers (layer 13). We interpret this result as indicating that the higher layers require more time to extract more specific and fine-grained features from the inputs, gradually stabilizing to produce optimal results. We also noted that the non-pretrained model demonstrated a more consistent pattern towards better metrics from the input layer to the output layer because the model was able to focus on learning information gradually from a single e-SNLI data source. In contrast, the pre-trained model used the Multi-Genre Natural Language Inference (MultiNLI) corpus in addition to the SNLI dataset, which covers a wider range of spoken and written text genres and supports cross-genre generalization evaluations. The use of a more diverse dataset may have made it more difficult for the pre-trained model to learn information evenly across all layers. In addition, our analysis revealed that layer 13 of the pre-trained model exhibited outstanding results, as highlighted in the heatmap. Upon closer examination, we discovered that the last layer of the DeBerTa model is a linear layer that concatenates all the information from the previous layer and makes a prediction based on one of three classes.

5.4.4 Multi-layer experiment

Our findings, as presented in Table 1, indicate that combining multiple layers to create a flattened representation can improve the performance of the probing model relative to using a single layer. Notably, layer 5 exhibited particularly poor performance, which impacted the overall performance of the flattened layer from layers 5 to 8, leading to worse results compared to other layers. This suggests that care must be taken when selecting layers to flatten and input into the probing model to avoid negative impacts on overall performance.

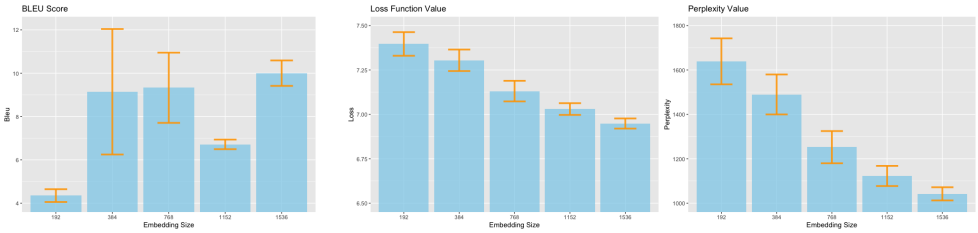
5.4.5 Extended probing model structure

Although our quantitative results show patterns and improvement across the experiments, our qualitative results did not exhibit significant differences across each variant and failed to recover the underlying reasoning information. We hypothesize that the limitations of the one-layer LSTM structure may not be sufficient to support both language construction and logical relationship building, which led us to explore more complex models, such as multi-layer LSTM, to test this hypothesis.

Our experimentation with multi-layer LSTM models did, in fact, result in a significant improvement in the BLEU score, which showed a strong correlation with the number of layers used (Appendix 5). Intuitively, this makes sense as more layers provide more opportunities to study the representations. However, upon closer examination of the results, we discovered that the multi-layer LSTM falsely predicted more popular words such as "is" and "a", which inaccurately increased the BLEU score. This indicates that a more complicated structure may not necessarily be effective in generating machine reasoning that aligns with human reasoning, particularly if the representation does not encode sufficient information.

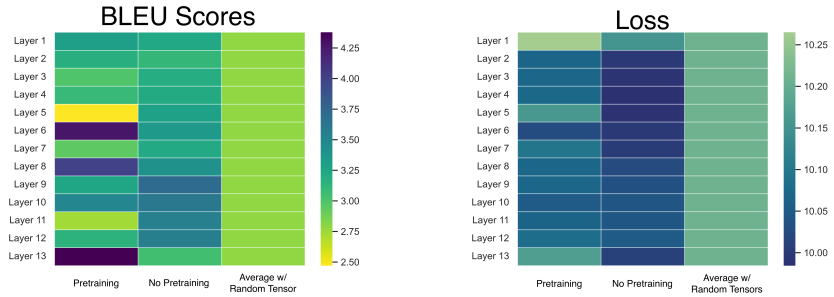
5.4.6 Google flan T5 exploration

We were also curious about the applicability of our model to other tasks, and we chose to explore this using the Google Flan dataset. Our preliminary results indicate that probing Google Flan T5 was able to achieve a BLEU score of 2.996 and a perplexity of 1.6e3 (Appendix Figure 8). Qualitative analysis revealed that the model was able to recover some popular math notation, such as the equal sign and parentheses, and even "pretend" to calculate certain equations (See appendix for example). However, despite these positive findings, the model still fell far short of reproducing the true rationale behind the given problem. This outcome was not unexpected, as the task at hand involves complex multi-step reasoning, and the limited information encoded in the hidden states data may not be sufficient to capture all the necessary reasoning steps. However, we do consider this a positive result for our model's applicability to probing models trained for tasks other than SNLI.



(a) BLEU Score (b) Loss Values (c) Perplexity Values

Figure 2: LSTM input embedding size Experiments: The yellow interval represents the value range from 5 runs. Larger embedding size tends to have better and more stable results



(a) BLEU Score (b) Loss Values

(c) Perplexity (d) Classifier Accuracy

Figure 3: Model type experiments across layers. Lighter color means a weaker result and darker color means a stronger result. Both SNLI models perform better than random. section 5.4.3 analyzes the result difference among layers.

6 Analysis

A representative example of our generated explanation when probing the pretrained model is included below. The premise / hypothesis inputs to the pretrained model are included for reference, as is the

Experiment	BLEU score	Perplexity	Loss	Classifier Accuracy
Best Layer 13	4.378	26107.582	10.170	**38.19%**
Flatten layers 1 to 4	5.519	**2839.321**	**7.951**	33.2%
Flatten layers 5 to 8	4.385	2874.873	7.964	33.1%
Flatten layers 9 to 12	5.636	2853.512	7.956	33.3%
Flatten layers 1 to 8	5.780	6850.272	8.832	33.06%
Flatten all layers	**5.801**	3313.2578	8.106	34.62%

Table 1: Metrics comparison between best layer and multi layers for Pre-trained Model

"true explanation" from the eSNLI dataset. In this example, relatively little information is extracted from the model that matches the true explanation, as highlighted, and the generated explanation is clearly not natural, human generated language. This is typical of the quality of many of our generated explanations, expressing only small aspects or glimpses of human reasoning.

- **Premise**: A woman is holding a sign that says honk to indict bush.
- **Hypothesis**: The woman is swimming in her pool.
- **True Label**: 2 (contradiction)
- **eSNLI Explanation**: A woman cannot swim while holding a sign .
- **Generative Text**: The cannot The women cannot be be on

During our model development, we found our model maximized BLEU scores through the production of sentence fragments, phrase repetition and nonsense sentences. When manually examining generated sentences it becomes clear that our model was only somewhat successful at the task of extracting knowledge into natural language. Although the explanations contain some information, they do not form coherent expressions of human reasoning.

Our classifier's poor performance on the generated explanations is evidence of the difference between the language of the generated explanations and the language of human authored explanations. The classifier had very high (>95%) accuracy on the eSNLI generated explanations but much lower accuracy on results from the probing model, as seen in Fig. 3. We believe this indicates that although reasoning information can be extracted from our model, it is not sufficiently rich to generate a coherent explanation in natural language which matches that of language used by humans to describe reasoning.

Through our qualitative analysis, we come away with the understanding that under our current probing model, we were unable to extract the information necessary to generate an explanation. This suggests that these models may not be fully aligned with human reasoning. However, using the same probing model on multiple SNLI models allowed us to compare the amount and type of information encoded in each, expanding our abilities to understand relative differences in reasoning used between models. This analysis ultimately supported our initial beliefs that pretrained models were able to more thoroughly encode human reasoning than non-pretrained models on the same task.

6.1 Additional probing model findings

Due to the originality of this research problem, we have gained several insights from our exploration of this specific probing method that may be useful to the broader explainable AI community. Our key learnings include:

1. During training, we observed the double descent phenomena (9) and found that adding dropout before inputting to the LSTM can help mitigate this problem without negatively affecting performance.
2. Similar to other text generation tasks, we also experienced repetition problem. To address this, we experimented with three different sampling strategies, including sampling all words from the top two predictions, sampling the first half of words from the top three predictions, and sampling the second half of words from the top three predictions. We found that the second option worked best (6).
3. While normalization can help improve training speed, it does not fundamentally assist the probing model in decoding more information. This is because normalization only scales the values and does not change the distribution of the data.
4. We also explored the potential benefits of adding a convolution layer to scan information between

hidden states. However, our findings suggest that this approach did not significantly improve the probing model's ability to decode more information. This may be due to the fact that hidden states are independent units in the network and have low correlation.

7 Conclusion

7.1 Overall Findings

Our results show that there is extractable knowledge about reasoning from the SNLI model hidden states, as seen by our improved BLEU scores for a probing model initialized with model states as opposed to random tensors. This is a promising result for the ability to use natural language to provide insight into model labels and outputs. However, we also conclude that these models are only partially aligned with human reasoning, as demonstrated by the weak qualitative performance of probing model at creating full explanations.

7.2 Limitations

Although we endeavoured to be thorough in our analysis, we are aware of limitations of our work:

- **Choice of Model:** We focused our efforts on building an LSTM model. Although this was a choice motivated by a desire for a simple model to prevent the "hallucination" of knowledge by our probing model, it would be informative to explore other choices of architecture for the model to see if other models are more effective at extracting model.
- **Pretraining vs Not Pretraining Probing Model:** Because our probing model was not pretrained, it essentially had to learn both the task and natural language at once. We purposely chose a non-pretrained model in order to prioritize a high selectivity model (Hewitt and Liang, 2019), but as a potential result had less natural generated language, whereas a pretrained model may have had more ease generating fluent sentences.
- **Need for Explanations:** Our model is constrained by one definition of human reasoning, based on the examples of human language explanations from the eSNLI dataset. This could limit ease of adapting to models trained on other tasks where no such explanations exist.

7.3 Future Work

We believe that this work inspires a wide variety of potential paths for future exploration in the space of using probing models to improve model interpretability. We can see several different avenues for this work, both going more in depth in the space of probing SNLI models and treating our model as a foundation upon which to build similar probing models for probing models built for different tasks.

- **Probing Additional Models:** Both models we probed were trained on the SNLI task, which has a known weakness of being subject to spurious correlations. Models for other tasks could be analysed using our probing model to compare how different model types do or do not encode human reasoning. This would create a rich field of model analysis as having logically sound models is a desirable quality for many applications. This would be easiest to accomplish in other spaces where there are established corpora of examples of human reasoning for the task which can be used to train the model.
- **Improved Analysis Tools:** Our classifier had limited utility because the generated explanations were too distant from the original language of the eSNLI dataset. An improved classifier could be created with a more powerful ability to classify the logical relation of the sentence fragments generated by models. Additionally, we could design a modified BLEU score which analyses only subjects of sentences to analyse the ability to extract subject material across layers. This would allow us to perform a deeper analysis of how information is encoded differently across layers or between models.
- **Pretrained Probing Model:** There is some controversy about the efficacy of pretrained models (Hewitt and Liang, 2019), as they can "hallucinate" knowledge due to their understanding of natural language, but we do believe it could be informative to explore a pretrained probing model. This could allow for more fluent natural language to be produced, potentially expanded the range of knowledge that could be extracted from models.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *CoRR*, abs/1812.01193.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *CoRR*, abs/2006.03654.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *CoRR*, abs/1909.03368.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017a. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017b. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations.

lkulowski. 2020. Lstm encoder decoder. `https://github.com/lkulowski/LSTM_encoder_decoder`.

Jesse Mu. 2023. Nl probing. `https://github.com/jayelm/nl-probing`.

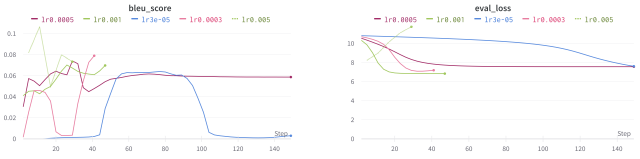
Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding.

PapersWithCode. Natural language inference on snli. `https://paperswithcode.com/sota/natural-language-inference-on-snli`.

Sean Robertson. 2020. Pytorch tutorial. `https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html#training-the-model`.

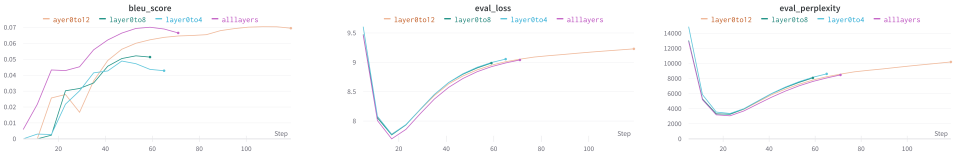
Thom Wolf. Gpt2tokenizer. `https://huggingface.co/docs/transformers/model_doc/gpt2`.

A Appendix



(a) BLEU Score (b) Loss Values

Figure 4: Explored learning rates as part of hyperparameter tuning, presenting best-performing rate. Learning rate search was done before addressing double descent and repetition issues.



(a) BLEU Score (b) Loss Values (c) Perplexity Values

Figure 5: Extended LSTM experiment results: stacking more layers lead to better result

(a) BLEU Score (b) Loss Values (c) Perplexity Values

Figure 6: Sample Strategy comparison results: sampling from top k in first half outperforms

A.1 Google Flan Generations

- Question: Bill's shop sells candy bars by the full case. 80% of a full case is added to 96 candy bars already in the case to fill it. How many candy bars are in a full case?
- Options: A)500 candy bars, B)450 candy bars, C)510 candy bars, D)375 candy bars, E)480 candy bars
- Rationale: (80/100) * X + 96 = X.80*X +96= X96 = .20 * XX = 480 Answer: E
- correct: E
- Generated text: IM O 1) = (5 (k x + = x Answer = (x Answer Answer is

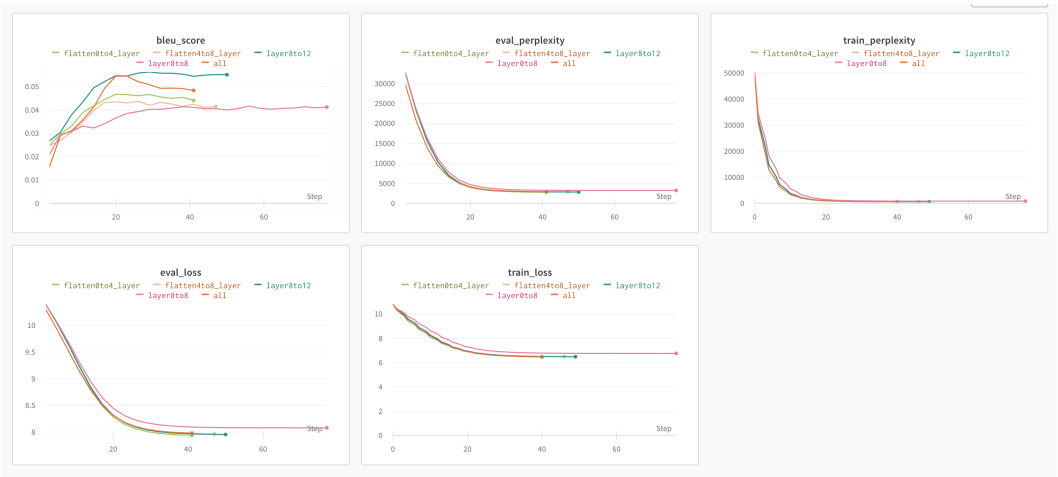


Figure 7: Flatten layer experiment: more representations are better in general

Figure 8: Flan explore T5 best parameter

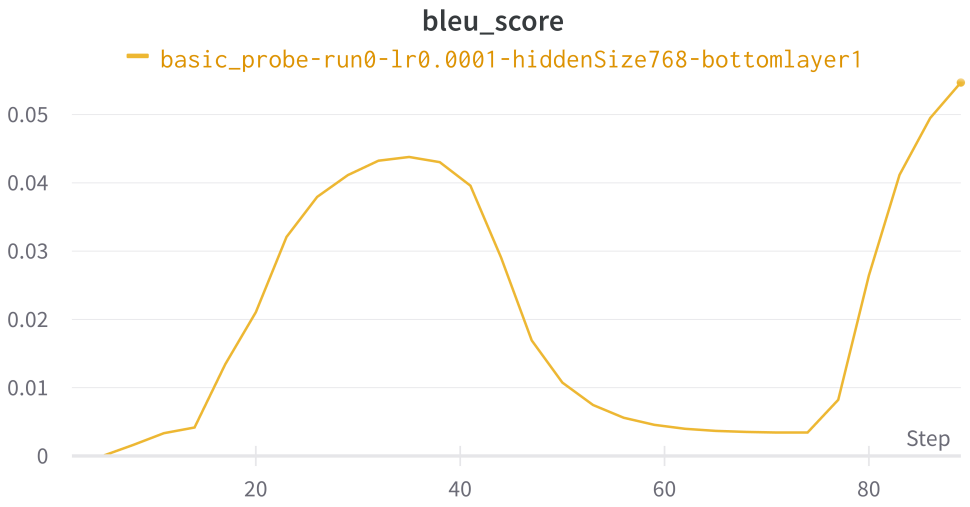


Figure 9: Double descent phenomenon