Won't You Be My Neighbor? Probing Informational Spread in Contextual Representations of Natural Language

Stanford CS224N Custom Project

Hagop Chinchinian
Department of Electrical Engineering
Stanford University
hagop@stanford.edu

Avi Gupta
Department of Political Science
Stanford University
avigupta@stanford.edu

Sevahn Vorperian
Department of Chemical Engineering, Sarafan ChEM-H
Stanford University
sevahn@stanford.edu

Abstract

Masked language models produce contextualized representations, where token embeddings encode information about its role and function in the overall sequence. However, the extent to which these representations reflect neighboring word identity at different offsets remains to be explored. In this project, we apply probing to extract mutual redundancy in contextual BERT word encodings. We first construct custom datasets of paired token embeddings over various layer-offset combinations from 10.4K documents from the HuggingFace Wikipedia dataset. We then fit several variations of linear models (using different initialization schema and dimensionalities) to predict context word identities from a center contextualized word encoding. In Experiment 1, we combine a 768 x 768 linear classifier (initialized to the identity matrix + Gaussian noise) with a the frozen weights of a 768 x 30522 classification matrix extracted from the BERT model. In Experiment 2, we fit a 768 x 30522 matrix created with Xavier initialization. In Experiment 3, we initialize our linear classifier using the matrix extracted from BERT, but do not freeze the weights. Across all experiments, we find that contextualized word encodings are substantially redundant, with simple linear models achieving relatively high prediction accuracy of neighboring word identities. The prediction accuracy is distributed asymmetrically, as neighboring tokens that come *before* the center word are predicted much more accurately. Moreover, the ability to predict more distant neighbors increases throughout the layers of BERT, with deeper layers enabling the predictions of further away words with higher accuracy. These preliminary findings are one of the first steps towards exploring interpretability of BERT encodings. Moreover, these results suggest that future work on redundancy in contextualized BERT embeddings could facilitate reduction of model complexity and word embedding size.

1 Key Information to include

- Custom Project
- Mentor: John Hewitt
- External Collaborator/Mentor: Ethan Chi
- This project is not shared with other courses.

2 Introduction

A wide variety of traditional NLP tasks, such as part-of-speech tagging, numeracy, dependency mapping (just to name a few examples), seek to model the intuitive ways in which humans understand the semantics of natural language. These factors, among others, contribute to the meaning of words as they are understood in the context of sentences. Pre-trained encoders presently offer the highest-performance for state-of-the-art NLP tasks (e.g. ELMo, BERT, RoBERTa, GPT, etc), relative to earlier static word embedding models, which are computed from simple co-occurence statistics at the corpus levels (e.g. GloVe, word2vec, n-gram). However, the extent to which human-perceived semantic features are reflected in high-performing pre-trained sentence encoder embeddings remains unclear. As the general success of these models would suggest, studies indicate that word meaning is accurately captured in contextual word encodings (Wallace et al., 2019). Moreover, it has been demonstrated through ablation studies that long-range word relationships are reflected in a given word embedding (Khandelwal et al., 2018). Most profoundly, contextual word encodings capture syntactic and semantic details of the sentence from which they were generated. (Tenney et al., 2019).

In this project, we seek to expand on the existing literature on contextual word encodings by attempting to directly measure the degree to which the specific context in which a contextual word encoding was generated can be readily extracted from the encoding vector itself. Using encodings generated by BERT (Devlin et al., 2018), we demonstrate that a simple linear model can predict the identities of neighboring context tokens with reasonable accuracy given a center word's contextual encoding. Moreover, we show that prediction accuracy for more distant words increases with the layer depth in a BERT model at which a contextual encoding was generated, which corresponds to the development of richer semantic meaning. Our findings provide greater interpretability of the contextual encodings generated by BERT. These results also suggest a possible direction to explore in reducing model complexity, improving encoder efficiency, and shrinking the size of word encodings.

3 Related Work

Tenney et al. (2019) demonstrates that the contextual word encodings developed at each layer of a BERT encoder model vary in the degree to which they capture syntactic and semantic meaning. In particular, the contextual word encodings at different layers of the BERT model encode information that is applicable to a variety of classical NLP tasks, such as part-of-speech tagging, parsing, named entity recognition, semantic roles, and coreference (Tenney et al., 2019).

In addition to probing the contextual encoding vectors generated by BERT, Tenney et al. (2019) provides several useful metrics for evaluating performance of language models at high-level language tasks; these include (1) a measure of the individual importance of each layer in accomplishing a particular task; (2) the average layer at which a particular task is adequately achieved; and (3) a measure of the contribution of each additional layer relative to all prior layers in accomplishing a particular task. We extend Tenney et al. (2019) by applying similar linear proving methods to word identity prediction, a topic that has not been previously explored but nevertheless has important implications for the interpretability of contextual word encodings.

Pimentel et al. (2020) provides an analysis of the value of BERT-generated embeddings for high-level language tasks, suggesting that these embeddings provide no useful information for the language task that is not already provided by the individual word embedding. This is demonstrated by estimating gain (essentially, the amount of information obtained for a particular task relative to some control) for the purpose of part of speech tagging with BERT. This paper also provides a mathematically rigorous analysis (from an information theoretic perspective) of what probing approaches to encoder network analysis are actually trying to measure.

One critique of Pimentel et al. (2020) involves the use of a neural network for the purpose of probing. Such a model may be too complex and consequently prone to overfitting. Additionally, the paper only considers the task of part of speech tagging, not considering the task of context word identity prediction. While it is reasonable that a task like part of speech tagging would not benefit from the additional layers of an encoder, as further corroborated by Tenney et al. (2019), the same may not be true of word identity prediction, the task attempted in this project.

4 Approach

In this project, we use word identity as a proxy for the broader idea for information: we claim that representation i holds information about position j if it is possible to extract information about token j by applying a linear classifier over word identity to representation i. Our selection of a linear classifier differs from (Pimentel et al., 2020), who specifically argue for the selection of more complex probes. However, we employ a linear model to demonstrate that the context token word identity is readily extractable from a contextual word embedding itself and is not a property of a more complex model, and to minimize the potential for overfitting.

Focusing on masked language models, we treat the prediction of context word identity as an instance of a probing task (see previous literature review). In particular, for a particular (layer, offset) pair l, m, we fit the following linear classification problem:

$$\textbf{min}_{M \in \mathcal{M}} \Sigma_{s \in \mathcal{S}} \Sigma_{i \in 1..|s|} \textbf{Cross-Entropy Loss}((M\mathbf{r}_{l,i}), \textbf{token}_{i+k})$$

where V is the vocabulary size, d is the hidden dimension size, $\mathcal{M}$ is the set of $V \times d$ matrices, $\mathcal{S}$ is the set of all sentences in the corpus, $|s|$ is the length of sentence s, and $\mathbf{r}_{l,i}$ signifies representations drawn from layer l and token i, and $\textbf{token}_{i+k}$ is the one-hot representation of the word at position $i+k$ in the sentence. In other words, we are using the contextual encoding of a center word to predict a softmax distribution over the one-hot identity vector of a neighboring word.

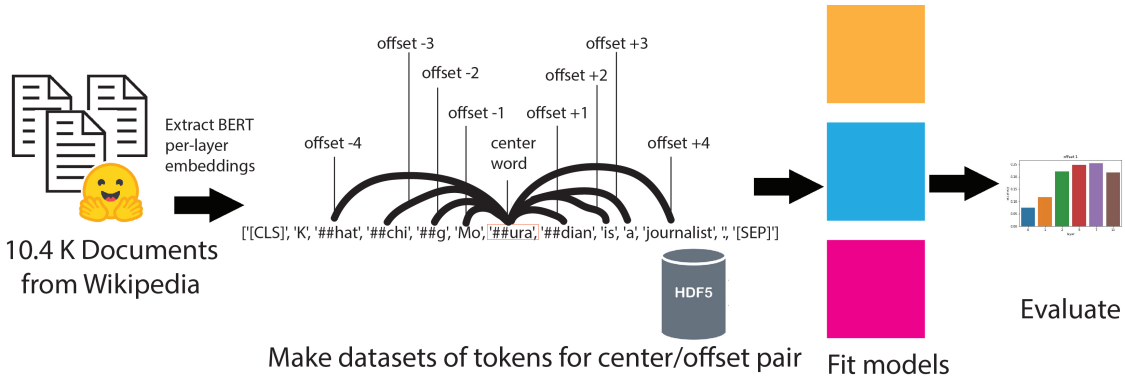


Figure 1: Summary of Data Processing and Model Training Pipeline

We fit $2K + 1$ linear models per layer, where K is the maximum offset that we would like to investigate. For the experiments presented in this report, K was 10. For each (layer, offset) pair, we fit a single model across all sentences in the corpus.

Broadly, we can think of this process as performing the masked language modelling objective—word prediction—both "prematurely" (i.e. at an earlier layer) and "offset" (at a different position than expected). Equipped with this framework, we can then ask the following analysis questions:

- **Speed of informational spreading**: How quickly does contextual information spread across the sequence? Starting from zero contextualization at layer 0, does contextualization increase linearly, superlinearly, sublinearly, etc.? In terms of our word identity probe, we ask to what extent the accuracy of a probe at layer l changes with l.
- **Distance:** How far does information spread? In terms of our word identity probe, we ask to what extent the accuracy of a probe with offset k changes with k.
- **Model complexity:** How do encoder model characteristics (parameter count, model class) affect the speed of informational spreading? This will be more extensively studied in future work on different models for generating contextual word encoding (other versions of BERT, RoBERTa, ELMo, etc.).

We intentionally fit relatively simple linear models in order to demonstrate that contextual information is 'easily' extractable from the contextual encodings themselves. Adopting mutual information as a

target metric, Pimentel et al. (2020) argue for the selection of arbitrarily complex probes in order to achieve the best possible results. However, because mutual information is representation-invariant, estimating mutual information using probing would not provide any information about the underlying properties of the contextual encodings (Hewitt et al., 2021).

Therefore, we present results from three simple linear models with slightly different structures and initialization schema. We fit these probes to contextual word encodings at selected layer/offset combinations. All of these linear models map a 768-dimensional encoding vector representing the center word (extracted from the given layer of BERT) to a 30522-dimensional probability distribution over the possible neighboring word identities for the particular offset at issue. In Experiment 1, we represent this linear model by initializing a 768 x 768 linear probe by adding random noise to the identity matrix. We then pass these results through the 'softmax matrix', a 768 x 30522 classification matrix extracted from the BERT model itself. We employ the softmax matrix on the theory that has residual connections that enable it to map from BERT vector space to BERT vocabulary space. To minimize the parameter space, we freeze the weights of the softmax matrix and optimize only over the 768 x 768 linear probe matrix. However, in analyzing the results of this experiment, we were puzzled by the extremely high accuracy achieved at offset -1 for all layers (see Figure 3 below) and wondered whether this was attributable to some property of the softmax matrix. (We also, of course, thoroughly examined our data loader code for bugs, but found that everything was working as expected.) In order to test our hypothesis, we naively initialize a 768 x 30522 matrix using Xavier initialization in Experiment 2 on a small subset of the layer/offset combinations from Experiment 1. Finally, in Experiment 3, we initialize a 768 x 30522 matrix to the weights given by the softmax matrix, but do not freeze the parameters on the same subset of layer/offset pairs. Due to computational constraints (we ran out of AWS and Google Cloud credits partway through training), we were unable to finish training some of our models, and therefore can only present partial results for some experiments.

5 Cosine Similarity Analysis

As a preliminary step, we mapped the cosine similarities of the contextual word encoding vectors at different layers and offsets. We observed that token embeddings closest to the center token (i.e., those with the lowest absolute offset value) exhibited the highest cosine similarity across all layers. However, as the layer number increased, farther away neighboring tokens exhibited higher cosine similarity (Fig. 2).

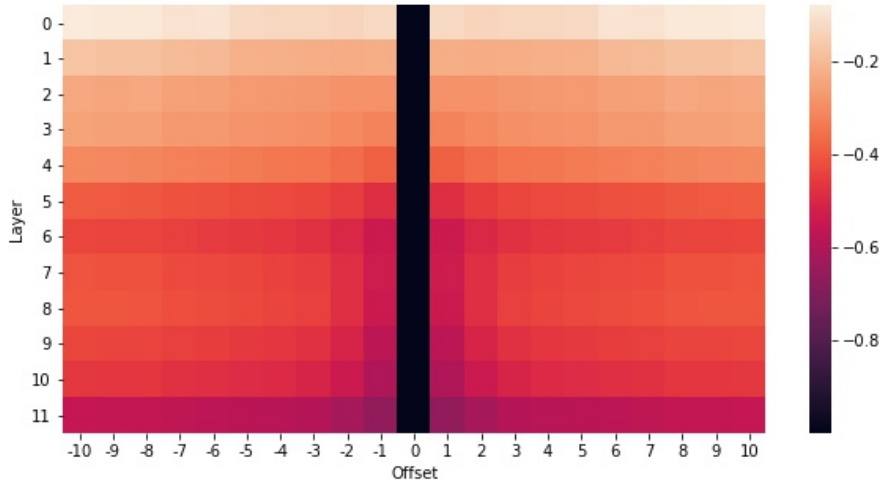


Figure 2: Cosine Similarities of Contextual Word Encoding Vectors by Layer and Offset

.

6 Experiments

6.1 Data

We begin from a corpus of English-language Wikipedia articles made publicly available on the HuggingFace platform (Foundation). We then truncate and tokenize individual items from a subset of the corpus using HuggingFace's implementation of the BERTTokenizer (Wolf et al., 2019). We then dump the center token and offset token identities and word encodings to a separate hdf5 file for each layer/offset pair using h5py (Collette, 2013). We use these hdf5 files to train a separate linear model for each layer/offset pair. Due to a lack of computational resources (we exhausted both our allotted AWS credits and our personal Google Cloud credits), we were only able to use a subset of 10,408 Wikipedia documents for training. We validated our models on a different set of approximately 3,500 Wikipedia documents.

6.2 Evaluation method

We employ a variety of metrics to assess the degree to which our linear probes predict context word identity. We assess the prediction accuracy of our models on both training and test sets by measuring the number of word identities that are correctly predicted (assigned the highest probability by the softmax output, computed using argmax over the softmax vector). We also measure the running and average cross-entropy loss on training and test sets, and graph loss curves to measure training convergence.

6.3 Experimental details

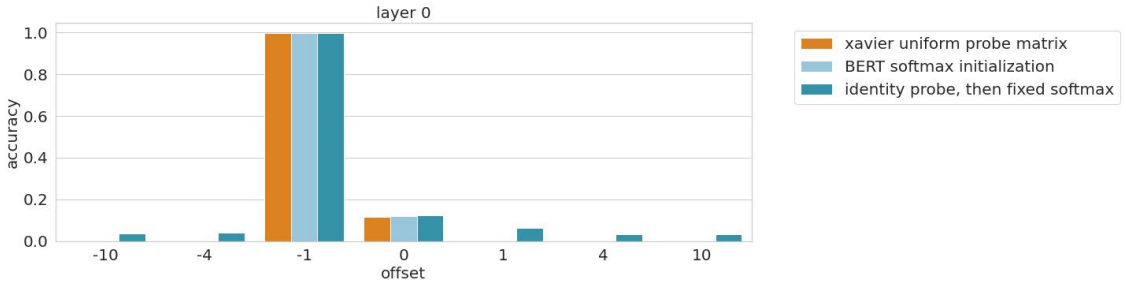
We train all models using an AdamW optimizer with learning rate 0.001. In Experiment 1, we train for 30 epochs (with the exception of the layer 0/offset 0 model, which was trained for 10 epochs). In Experiments 2 and 3, we train for 10 epochs. Which takes approximately 5-10 minutes per epoch using our current computational infrastructure (due to AWS and Google Cloud credit exhaustion, we were forced to train on FarmShare's relatively limited GPU resources). Models were implemented in PyTorch. We used HuggingFace's implementations of cased BERT tokenizers and models.

6.4 Results

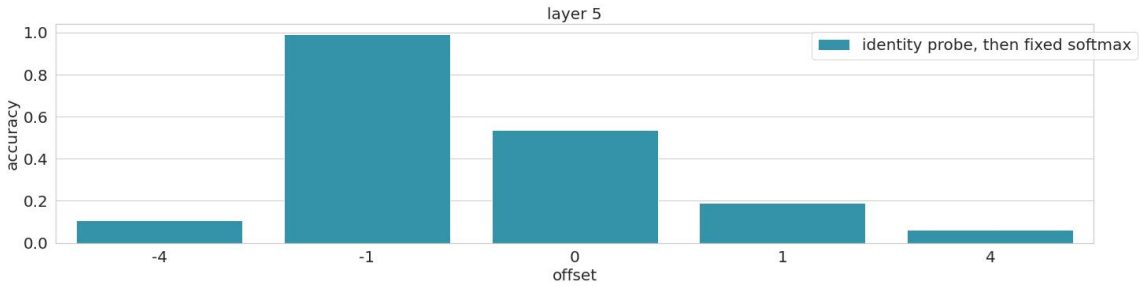
In this section, we present a subset of the graphs generated to justify our key findings. Additional data can be found in the Appendix. In also bears noting that in addition to measuring accuracy, we also measured the cross-entropy loss associated with our predictions. For brevity, plots depicting average and total loss were omitted from this report, but we found that losses were essentially perfectly negatively correlated with prediction accuracy.

In Figure 3, we observe that accuracy generally improves from layer 0 (representing the static word embeddings with which BERT is initialized) to layer 5, indicating that the context accrued by BERT's encoding scheme improves the probe's ability to predict neighboring words. Interestingly enough, context also enriches the probes ability to predict its own word identity, as demonstrated by the increased accuracy for offset 0. Moreover, we see that all probes perform roughly similarly for instances in which we have matched data, indicating that our results are not an artifact of the initialization scheme but rather of fundamental properties of the underlying contextual word encoding vectors. We also observe substantial asymmetry, as the probe achieves much higher accuracy on prior words (negative offset values) than future words. Given the left-to-right nature of the English language, there is some logic to this finding, since prior words are likely to contain important context that BERT would 'want' to encode.

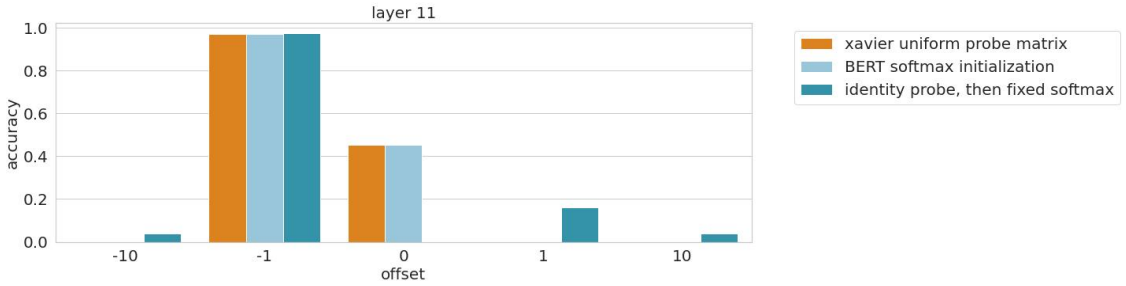
In Figures 4 and 5, we hold the offset constant to evaluate how performance at a particular prediction task evolves across layers. The models generally achieve higher prediction accuracy on closer neighbors (smaller absolute offset values), which is to be expected. In general, performance generally improves with deeper layers (see Figs. 5 and 4). However, at higher layers for smaller absolute offset values, performance worsens relative to middle layers, which may indicate a possible tradeoff between the breadth of context stored across the sentence as a whole and the amount of information stored about a particular neighboring word.



(a) Validation accuracy by offset for layer 0



(b) Validation accuracy by offset for layer 5



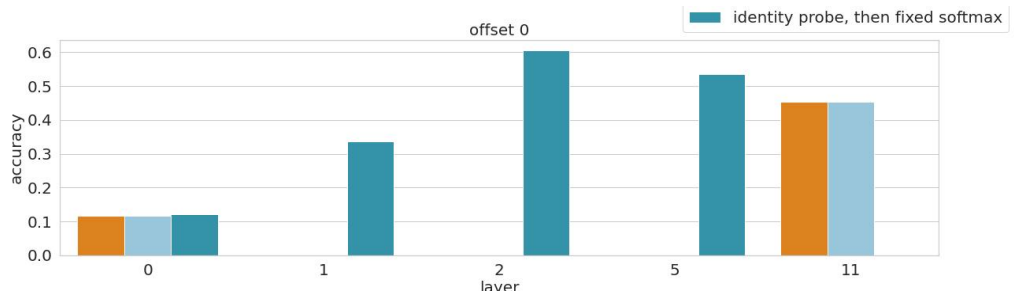
(c) Validation accuracy by offset for layer 11

Figure 3: Validation accuracy by offset for layers 0, 5, 11 for all experiments. "Identity probe, then fixed softmax" corresponds to Experiment 1, "xavier uniform probe matrix" corresponds to Experiment 2, "BERT softmax initialization" corresponds to Experiment 3.

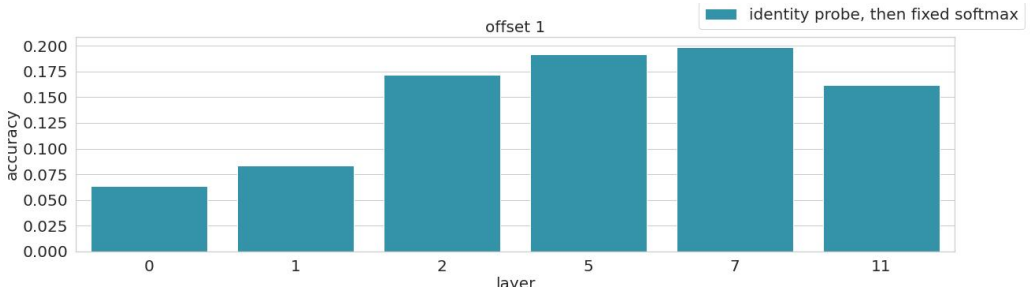
7 Analysis

In general, our preliminary results are in line with our expectations, and supported our qualitative hypotheses. Among the most interesting phenomena we observed in our experimentation occurred at offset -1. Across all three experiments we observed that a token immediately to the left of the center word (offset -1) had the highest prediction performance (Fig. 3). It is notable that this performance substantially exceeds even the baseline suggested by the cosine similarity (Fig. 2), which would suggest that the prediction accuracy values for offsets -1 and +1 should be roughly similar. Further testing will be required to determine the exact cause, but we hypothesize that it may be due to the key/query attention operation performed within the BERT transformer's encoder architecture, which is a linear function of the center word's immediate predecessor. Moreover, as a matter of general language modeling, we would reasonably expect that the immediately preceding neighbor would contain the most relevant contextual information that would be incorporated into the center vector. However, it is somewhat surprising that the accuracy for predicting offset -1 was higher than offset 0.

We also notice that prediction generally increases with deeper BERT layers, a finding consistent with our hypothesis that deeper layers of BERT "enrich" the contextual representation contained within the center word vector.

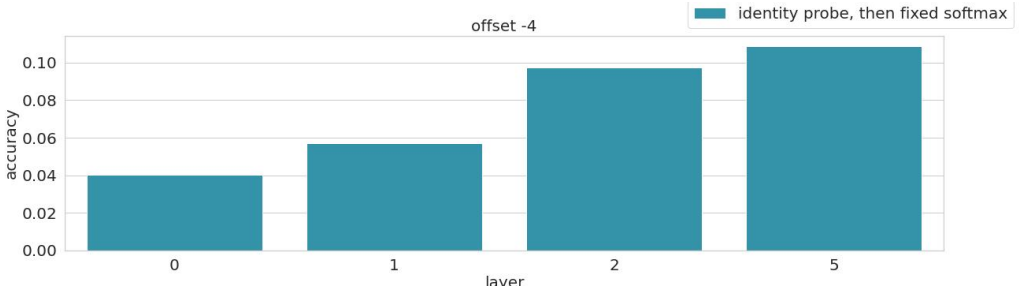


(a) Validation accuracy by layer for offset 0



(b) Validation accuracy by layer for offset 1

Figure 4: Validation accuracy by layer for offsets 0, 1



(a) Validation accuracy by layer for offset -4

(b) Validation accuracy by layer for offset 4

Figure 5: Validation accuracy by layer for offsets -4, 4

8 Conclusion

In this report, we present early results from fitting various simple linear probes to predict word identity from contextual word encodings generated by different layers of BERT. We find that context word identities are readily (i.e., linearly) extractable from BERT vectors using a linear probing scheme. This finding of redundancy in contextual word encodings suggests that word vectors contain substantial information about their neighbors. Moreover, we show that although it is easier to predict closer words than farther away words, deeper layers of BERT enrich the context available about farther neighbors, making more accurate predictions possible. We also observe substantial asymmetry, as prior context words are much more readily extractable than future words, with immediate predecessor words (offset -1) being predicted with near-perfect accuracy. One limitation of these preliminary findings is that they are only based on one encoding model (BERT) and one corpus (Wikipedia).

In future work, we intend to leverage greater computing resources in order to fully cover the range of layers and offsets using all of our models. We will also expand the size of our Wikipedia subset and to use different datasets, such as OpenWebText (Gokaslan et al., 2019), CNN/DailyMail (See et al., 2017), and other publicly available datasets. We also intend to test our approach on other cutting-edge NLP architectures, such as ELMo, RoBERTa, and GPT.

We also intend to make concrete improvements to our current methodology. We will optimize hyperparameters such as the learning rate, optimizer type, and weight initialization schemes. Each of the three experiments we present in this report is predicated on certain assumptions about the information encoded by the "softmax" matrix we extract from BERT's linear classifier, which we plan to study further. We will also explore similarity metrics besides cosine similarity.

References

Andrew Collette. 2013. *Python and HDF5*. O'Reilly.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Wikimedia Foundation. Wikimedia downloads.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie. Tellex. 2019. Openwebtext corpus.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context. In *arXiv*.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-Theoretic Probing for Linguistic Structure. In *arXiv*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *arXiv*.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *arXiv*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

A Appendix

In this appendix, we present additional plots depicting prediction accuracy, average model loss during training, and model training.

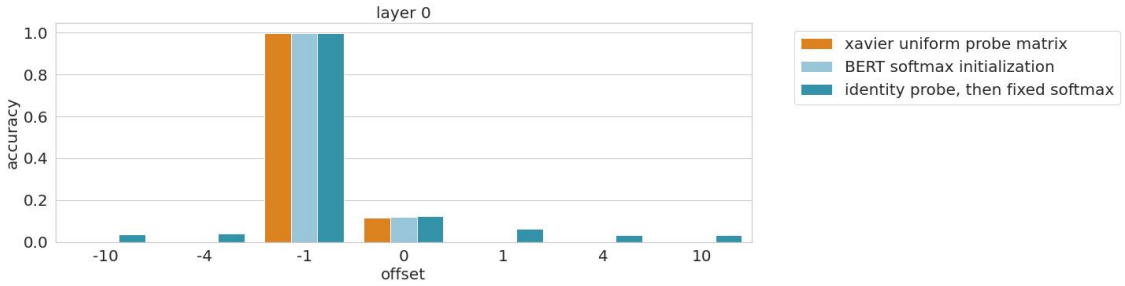


Figure 6: Validation accuracy at layer 0

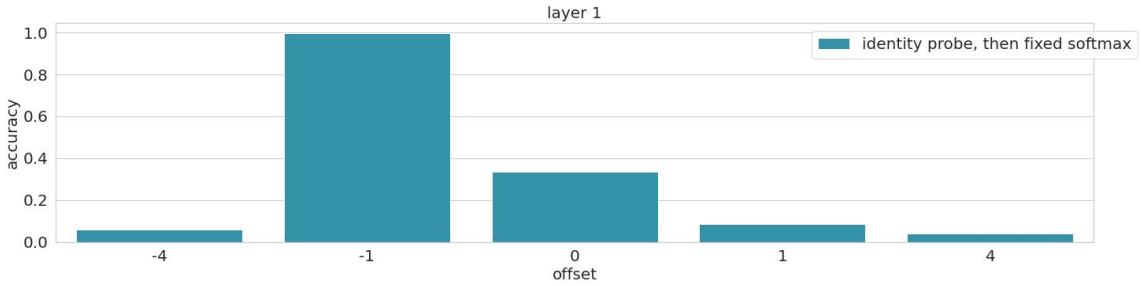


Figure 7: Validation accuracy at layer 1

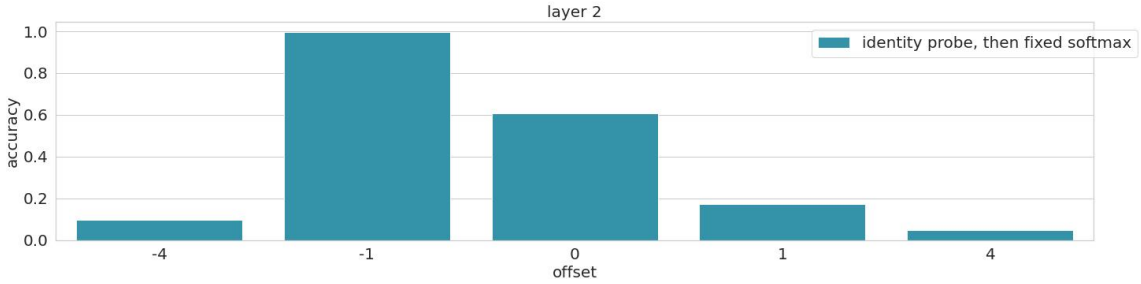


Figure 8: Validation accuracy at layer 2

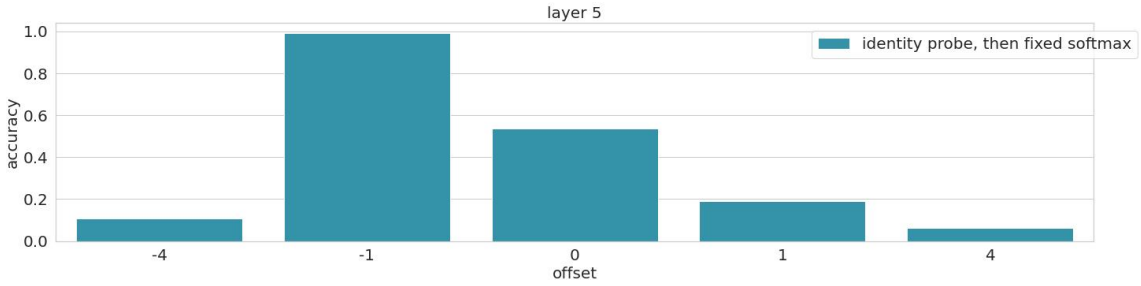


Figure 9: Validation accuracy at layer 5

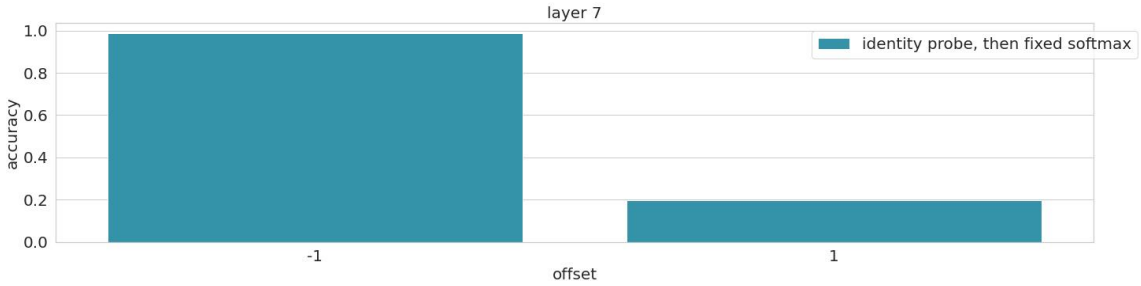


Figure 10: Validation accuracy at layer 7

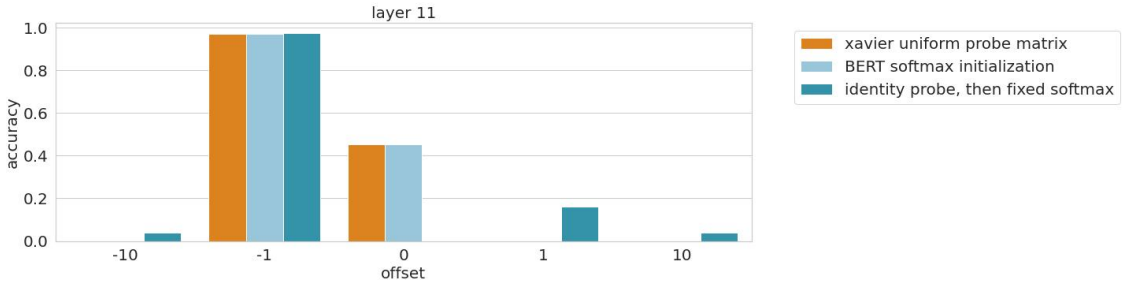


Figure 11: Validation accuracy at layer 11

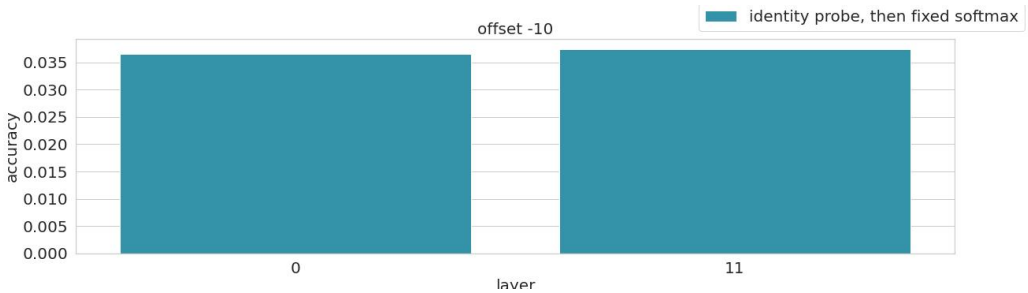


Figure 12: Validation accuracy at offset -10

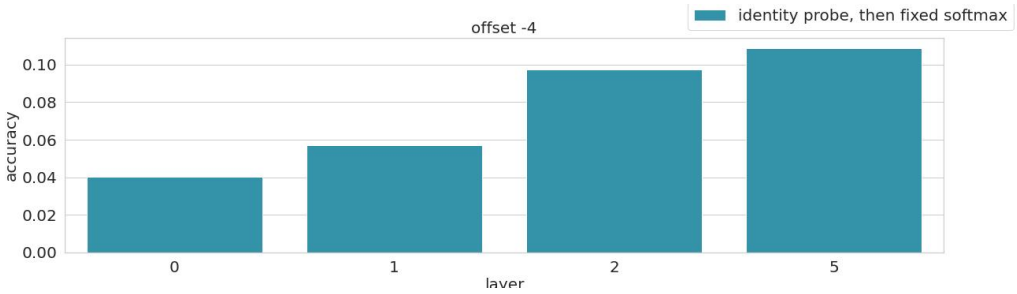


Figure 13: Validation accuracy at offset -4

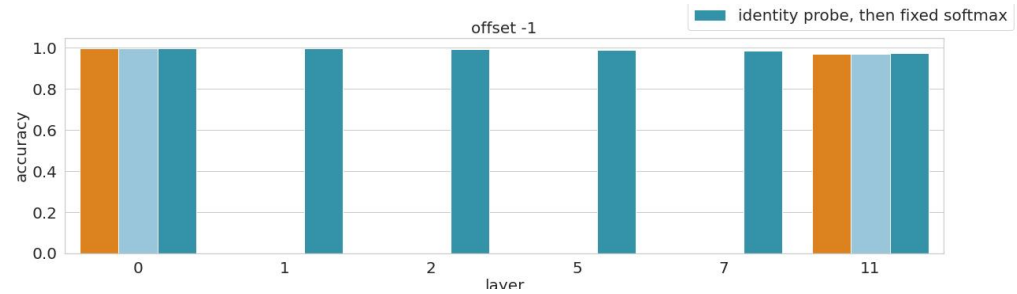


Figure 14: Validation accuracy at offset -1

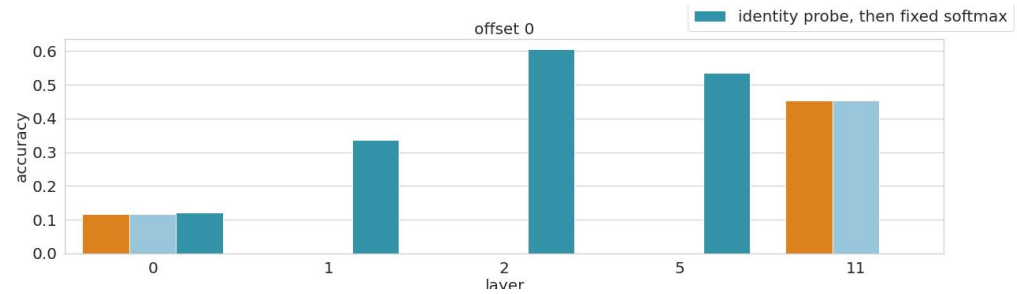


Figure 15: Validation accuracy at offset 0

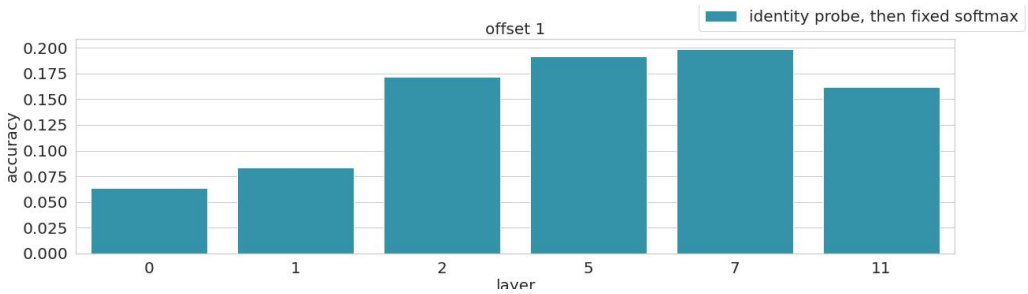


Figure 16: Validation accuracy at offset 1

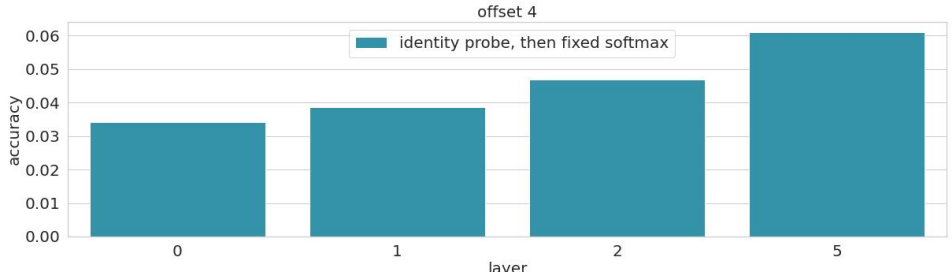


Figure 17: Validation accuracy at offset 4

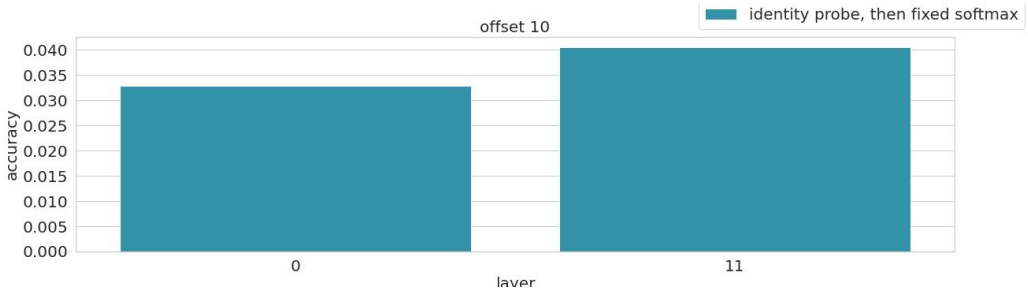
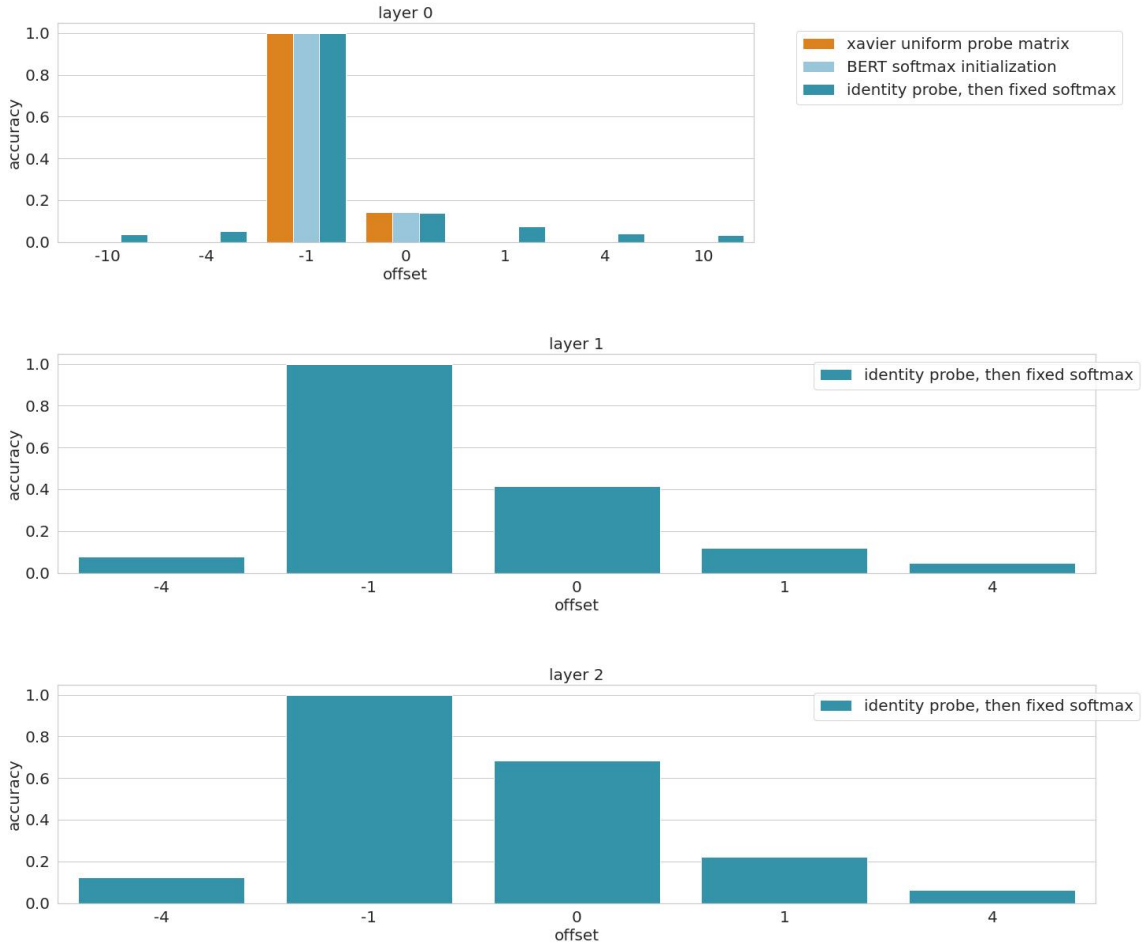


Figure 18: Validation accuracy at offset 10



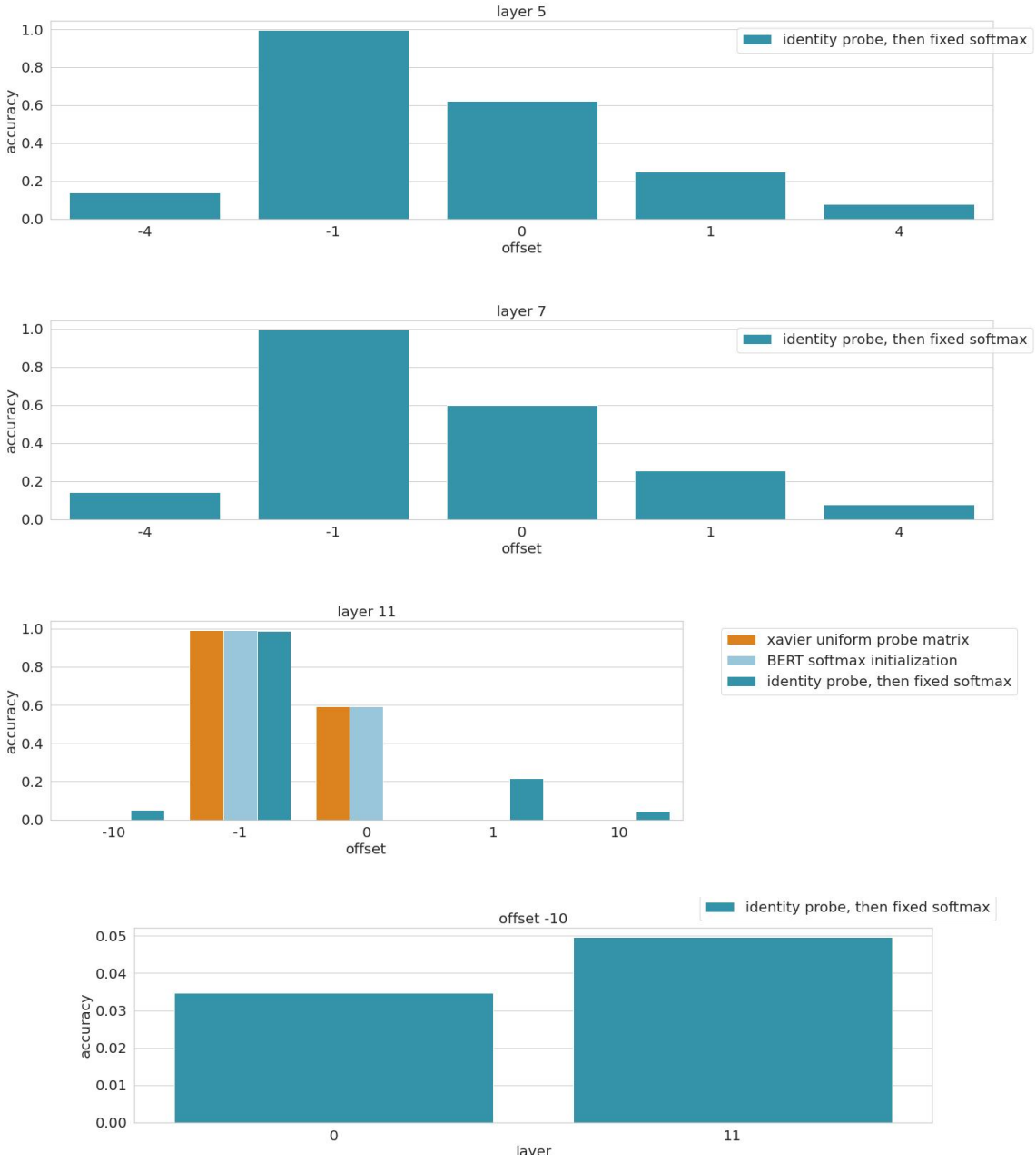


Figure 19: Training accuracy at offset -10

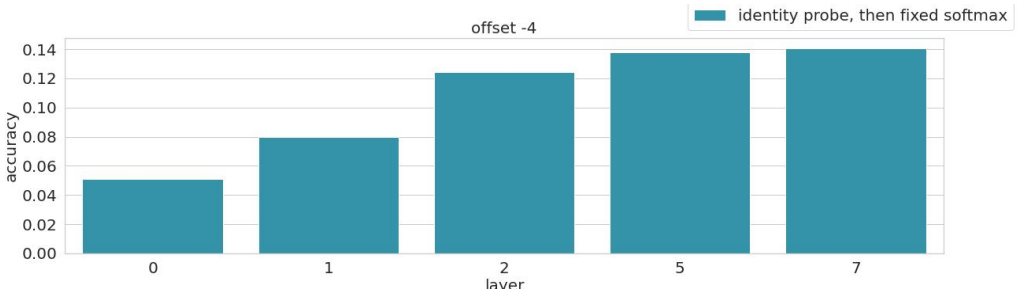


Figure 20: Training accuracy at offset -4

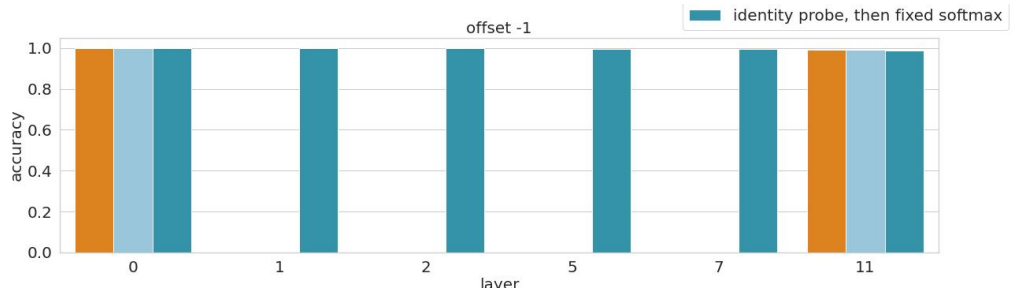


Figure 21: Training accuracy at offset -1

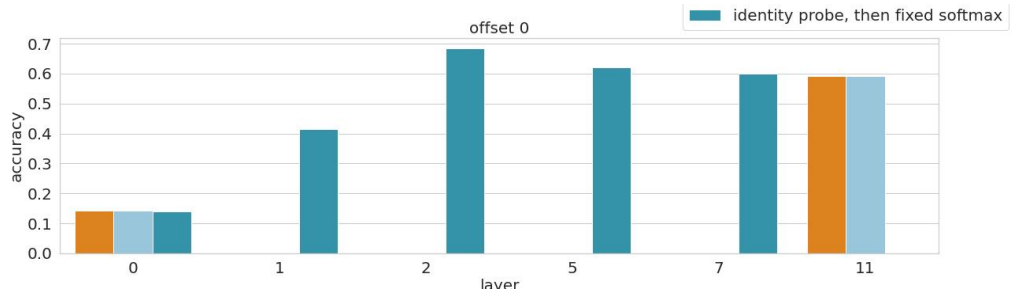


Figure 22: Training accuracy at offset 0

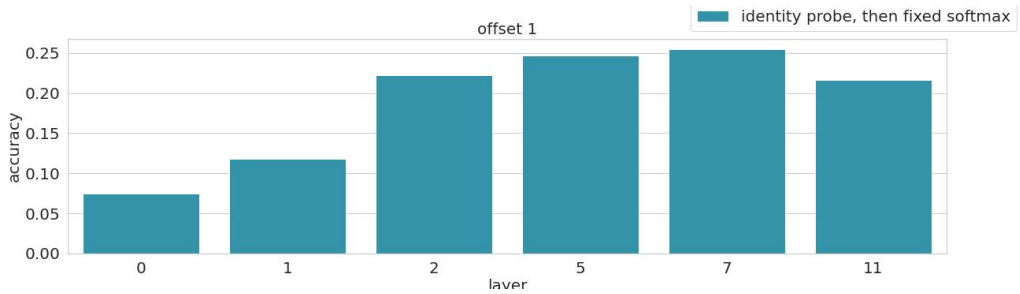


Figure 23: Training accuracy at offset 1

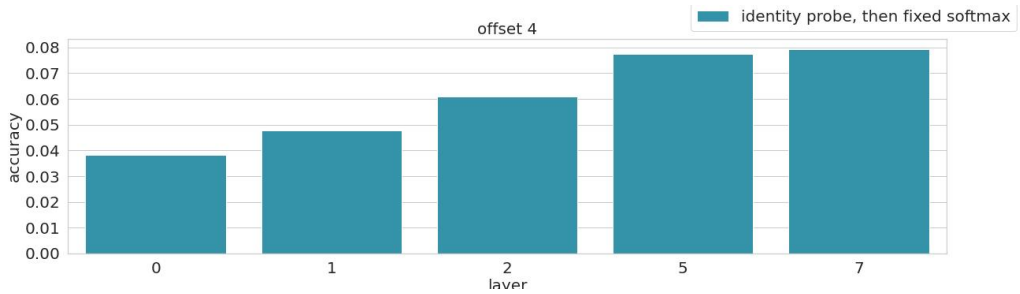


Figure 24: Training accuracy at offset 4

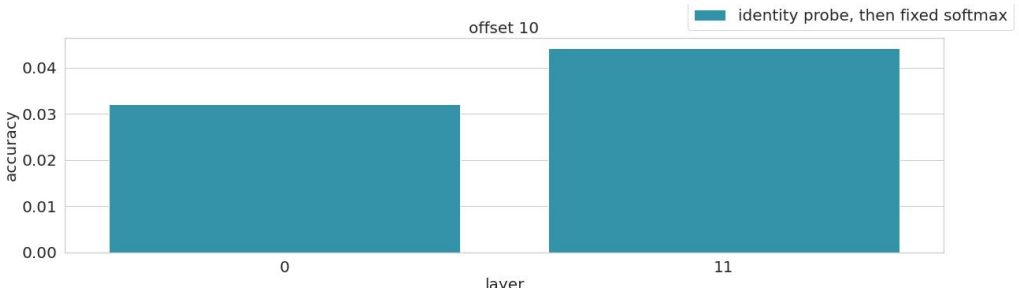


Figure 25: Training accuracy at offset 10

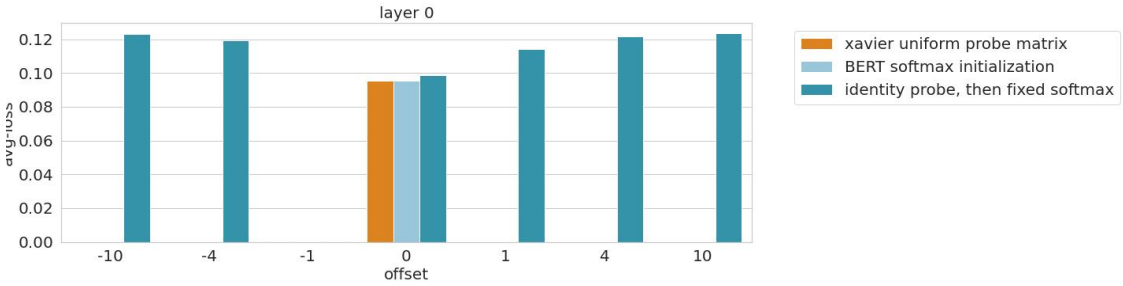


Figure 26: Average training loss at layer 0

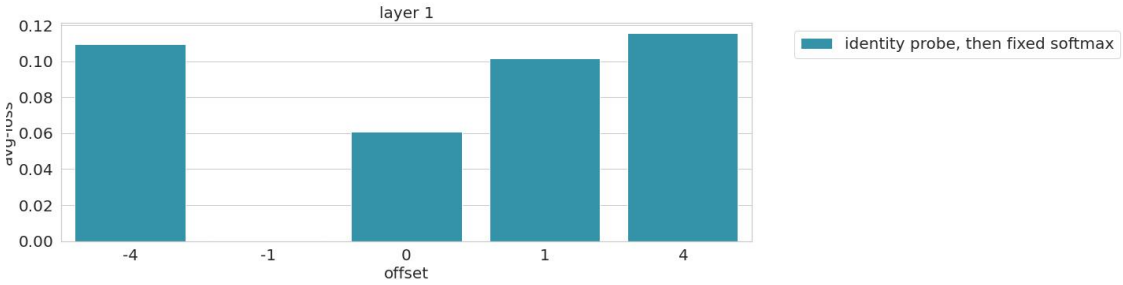


Figure 27: Average training loss at layer 1

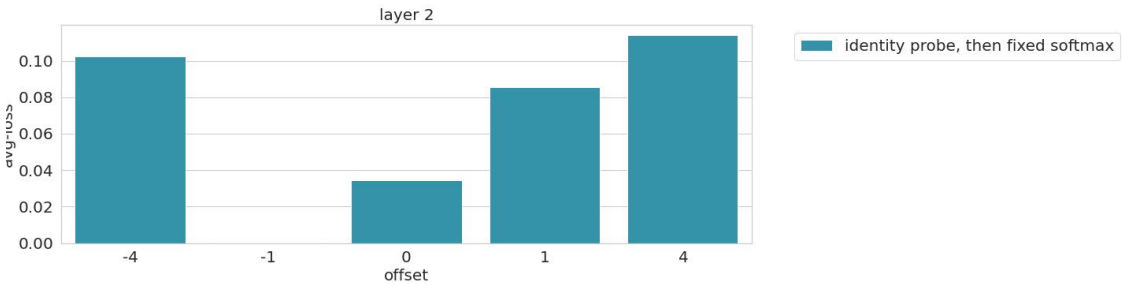


Figure 28: Average training loss at layer 2

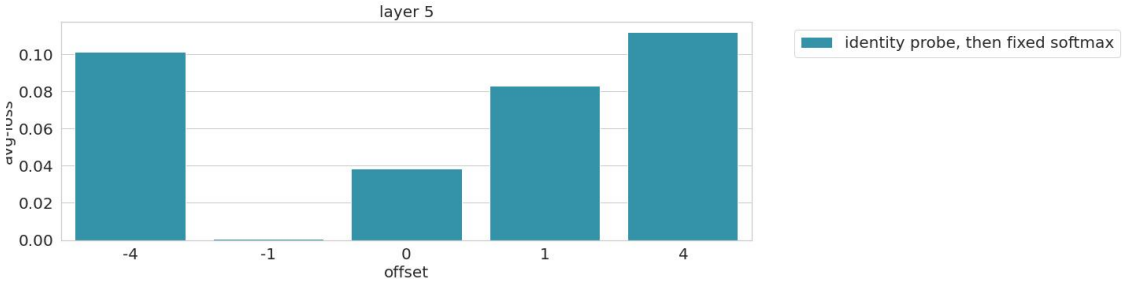


Figure 29: Average training loss at layer 5

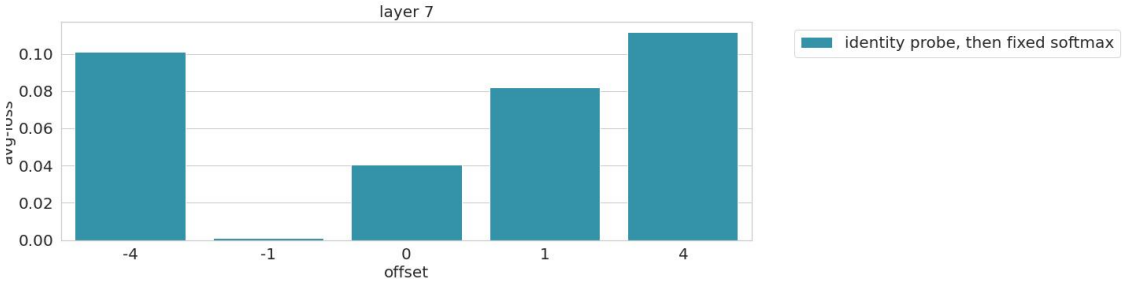


Figure 30: Average training loss at layer 7

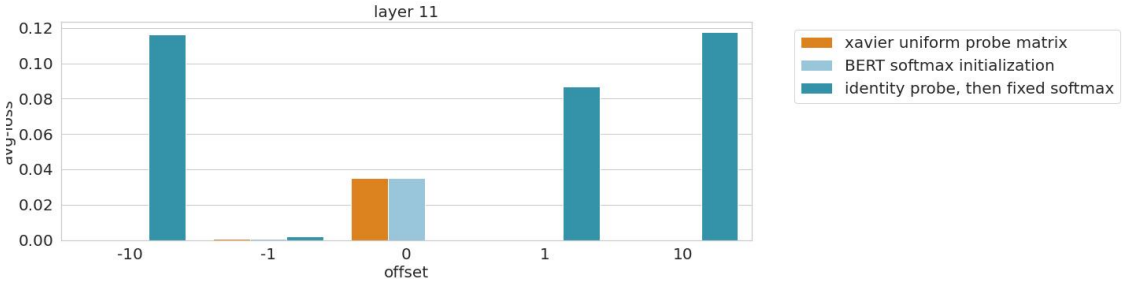


Figure 31: Average training loss at layer 11

Figure 32: Training loss during model fitting, all models