Exploring the linear separablity of syntactic and semantic information in BERT embeddings

Anonymous ACL submission

Abstract

Relations between syntax and semantics are not readily agreed upon. We seek to explore how representations of syntax and semantic information sets manifest in BERT embeddings, particularly the degree of the linear separability of each other in BERT embeddings by applying Iterative Nullspace Projection (INLP) to decompose BERT embeddings into syntactic and semantic subspaces. We also investigate how important the linear component corresponding to one information set is to solving a classification task that targets the other information set. Our results show that both syntactic and semantics informations are not linearly represented in BERT embeddings. Therefore INLP fails separate syntactic and semantic space from BERT embeddings and does not provide interpretable results. The results also indicate a factor of consideration when applying INLP, regarding the rank of the projection matrix.

1 Introduction

The boundary between semantics and syntax has been hotly debated, but do language model embeddings present this information in a way that is easily separated and recognized by humans? The objective of this project is to explore BERT's (Devlin et al., 2019) reliance on certain syntactic information when handling a semantic task, and vice versa. Specifically, we seek to quantify the importance of linearly-separable syntactic or semantic information when performing semantic or syntactic classification, respectively.

To achieve our goal, we apply a novel method Iterative Nullspace Projection (INLP from here) (Ravfogel et al., 2020)for removing information from an embedding. INLP iteratively trains linear models on a specific classification task, and projects the input on the intersection of the nullspaces of those linear models.

Our experiment scheme follows Elazar et al., 2020, which employs INLP to investigate whether BERT uses part-of-speech (POS) information when solving language modeling (LM) tasks. Similarly, we construct a linear probing system for a task and then employ INLP to generate a new embedding devoid of information learned from the probing task. We then evaluate the performance of this new embedding on another downstream task. Then we will perform the same procedure but switch the probing task and downstream evaluating task. To evaluate the separability of syntactic and semantic representation, we need two tasks that could extract those information on word level. Hence, we choose Combinatory Categorical Grammar (CCG from here on) tagging (Hockenmaier and Steedman, 2007) as the syntactic task and semantic tagging (Abzianidze and Bos, 2017) as the semantic task.

Our objective is that, by applying the INLP procedure to a syntactic task, we are able to separate the representation into a syntactic space and a non-syntactic space. We then compare the performance of a linear classifier for semantic labels using the original BERT embeddings with an otherwise identical model trained on embeddings projected onto the non-syntactic space. Conversely, we can define a semantic and non-semantic space by probing a semantic task, and then investigate the performance of embeddings projected onto those spaces when performing a syntactic classification task. The performance of these embeddings on their opposing classification tasks will give us an indication of how linearly separable the two information sets are.

The remainder of the paper proceeds as follows: Section 2 explores previous work related to our experiment. Section 3 provides a description of the probing and evaluation tasks and gives an overview of the experiment pipeline. Section 4 reviews our experiments and affiliated results. Section 5 discusses the implications of those results. Finally, section 6 gives an overview of the entire process and outlines possible next steps.

2 Related Work

The separation and overlap between syntax and semantics has been of interest to linguists for years. More recently, with the growing popularity of large language models, computational linguists have begun to explore how large language models deal with the boundaries of these information sets.

Huang et al., 2021 use paraphrase pairs and new target syntax to train a semantic encoder, syntactic encoder and decoder to learn separate representations of the semantic and syntactic information contained in BART embeddings, in order to create semantically equivalent paraphrases with the new syntactic structure. Alongside the encoders they also train an adversarial syntax discriminator to try and predict the source syntax from the semantic embeddings, thus encouraging the disentanglement of the semantic and syntactic information by training the semantic embedder to remove as much syntactic information as possible. Their results show that disentanglement of some information is possible. Though they do not achieve perfect separation of the two information sets. Other non-linear approaches to syntactic-semantic information disentanglement have been carried out in Chen et al., 2019

Unlike the aforementioned studies, we seek to explore the linear separability of syntactic and semantic information in large language model embeddings at the word level. To accomplish this task we apply the INLP method to syntactic (CCG) and semantic tasks in order to define the syntactic and semantic components of BERT embeddings that will be used in our downstream classification tasks.

INLP, introduced in Ravfogel et al., 2020, is a method to define a linear guarding function that masks all the linear information in a word embedding that may be used for a downstream classification task. In the original paper the authors use this method to remove gender bias from BERT embeddings of biographical descriptions and then measure how easy it is to determine an individual's gender from the guarded embedding by using various downstream classification methods. Beyond this example, the authors hypothesize several additional use cases for this procedure, including information disentanglement.

The authors of Elazar et al., 2020 use INLP for exactly this task. They use INLP to separate and guard certain linguistic information sets from BERT embeddings in order to better understand what information is being used by large language models, and not just what is encoded. The main premise behind this paper is that if a particular property is used to solve a task, then the removal of that property should negatively influence the model's ability to solve that task. Specifically, Elazar et al., 2020 seeks to quantify the importance of the information sets used for part-of-speech tagging, syntactic dependency labeling, named entity recognition and syntactic constituency boundaries on BERT's ability to perform the language modeling task.

We take a similar approach to Elazar et al., 2020 by separating the information sets used for CCG tagging and semantic tagging from word-level BERT embeddings, and test how the removal of these information sets impacts the embeddings' performance on these tasks.

3 Experiment

To isolate the syntactic and semantic information from word-level BERT embeddings efficiently, we implement INLP using method described in section 3.1. CCG tagging and semantics tagging are probing tasks for INLP to extract relevant information from embeddings, which are described in section 3.2,3.3 . We also conduct experiments using BERT embeddings from different layers to see which layer might contain more syntactic or semantics information, as described in 3.5.

3.1 The Iterative Null-Space Projection method

The INLP method first introduced in Ravfogel et al., 2020, is used to create a guarding function that masks all the linear information contained in a set of vectors, X, that can be used map each vector to $c \in C$, where C is the set of all categories. This is accomplished by training a linear classifier, a matrix W, that is applied to each $x \in X$ in order to predict the correct category c with the greatest possible accuracy. Once W is determined, for any $x \in X$ we can remove the information that W uses to predict c by projecting x onto the null-space of W, $N(W) = \{x|Wx = 0\}$. Call this projection function P_1 and let $\hat{x} = P_1(x)$. This removes all of the linear information in x that W used to predict the category c.

We iteratively apply this process until no linear information remains in $\hat{x}$, i.e. a linear classifier is unable to predict the correct category c with any probability greater than that achieved by guessing the majority class[1]. The final $\hat{x} = P_n(P_{n-1}(\ldots P_1(x)))$ contains no linear information about the categories in C and we call $P(x) = P_n(P_{n-1}(\ldots P_1(x)))$ the guarding function.

The projection matrix P derived by matrix multipcliaitons $P_n \cdot P_{n-1} \ldots P_1$ can be susceptible to numerical errors, therefore Ravfogel et al., 2020 utilized the following formula using *rowspace projection*[2] proposed by Ben-Israel, 2015 to compute the intersection of null spaces of weight matrix. Then projection matrix P is derived from null space projection of the intersection, $P = P_{\cap_{i=1}^{n} N(W_i)}$, instead. Our experiment follow the same computation.

$$\cap_{i=1}^{n} N(W_i) = N(\textstyle\sum_{i=1}^{n}(P_R(W_i)))$$

3.2 Data

We use the English Parallel Meaning Bank v4.0 (Abzianidze et al., 2017) to test the linear separability of the semantic and syntactic information in word-level BERT embeddings. This dataset consists of gold standard and silver standard word-level semantic tags. The gold standard contains 5,438 sentences with annotations that are manually verified while the silver standard contains 62,739 sentences with autogenerated annotations. All of our experiments are conducted on gold standard data with standard train/dev/test split[3].

The original dataset does not include CCG tags, however Abzianidze et al., 2017 utilized a CCG parser to produce CCG tags. We follow a similar procedure and apply a CCG parser (Yoshikawa et al., 2017) to develop word-level CCG tags. Once we obtain both CCG tags and semantic tags for the dataset, we can perform word-level syntactic and semantic probing tasks as desired. The total number of labels in CCG tags and Semantics tags are 159 and 72 respectively.

3.3 Probing tasks

The probing task involves training a linear classifier[4] on the final layer BERT embeddings in order to predict the CCG tag or semantic tag associated with each word. We will use this classifier in the INLP algorithm in order to create a guarding function for the information that is necessary to complete the task. Take CCG tag as an example: for a given embedding, v_{orig}, the projection that results from applying this guarding function, P_{syn} or P_{sem}, to the embedding will represent the non-syntactic information contained in the embedding and will from now on be referred to as the "non-syntactic component" of the embedding, $v_{nosyn} = P_{syn} v_{orig}$.

Similar to the above, the semantic probing task involves training a linear classifier on the final layer BERT embeddings in order to predict the semantic tag associated with each word. This classifier is used in the INLP algorithm in order to create a guarding function, P_{sem}, for the information necessary to complete the semantic tag labeling task. As described in the syntactic probing task, we use the resulting guarding function to compute a "non-semantic embedding", $v_{nosem} = P_{sem} v_{orig}$.

3.4 Evaluation tasks

Our goal is to determine which information sets captured in the BERT embeddings are relevant for our evaluation tasks. We thus use the components derived from the probing tasks to create new embeddings that isolate specific types of information. These embeddings are then evaluated on the syntactic and semantic tasks that were used for probing, and their performance is compared to that of the original embeddings. We also compare the performance of each model trained on one of these embeddings with another trained on new embeddings that are created by randomly removing the same number of dimensions from the original embeddings as are removed by the INLP guarding function. In doing so we can test the extent to which the loss of the particular information set of interest is responsible for the drop in performance, as opposed to a general loss of information.

We will assess each of the non-syntactic and non-semantic embedding types, the original BERT em-

[1] The stopping criterion follows Elazar et al., 2020, iterations will stop if the linear classifier achieve within one point above majority accuracy on development set.

[2] $P_R(W_i)$ in the formula means row space projection of weight matrix W.

[3] Gold standard dataset contains total of 34706 words, with 80% of training and dev data, and 20% of testing data

[4] The linear classifier will use Adam as optimizer (Kingma and Ba, 2014) implemented in torch (Paszke et al., 2017). Therefore the total number of parameters will be dimensions of BERT embeddings · number of labels, which will be 122112 for syntactic task and 55296 for semantics.

beddings and the embeddings created by randomly removing directional information on the CCG and semantic labeling tasks that were used in the probes. All the embeddings are listed in table 1.

Expression	Description
v_{orig}	Original BERT embeddings
$v_{nosem} = P_{sem}v_{orig}$	Gained after INLP with semantic task
$v_{nosyn} = P_{syn}v_{orig}$	Gained after INLP with syntactic task
Rand(v, n)	Embeddings v with n random directions removed

Table 1: Description of Embeddings

3.5 Layer-wise evaluation

In addition to the final layer BERT embeddings, we perform a similar analysis on the embeddings derived from different layers of the BERT architecture, in order to determine the separability of these information sets at each layer. For embedding v_{orig_i} from layer i, a linear classifier is trained for each probing task to acquire guarding functions P_{syn_i} and P_{sem_i}, respectively. Applying these projection functions, we are able to acquire v_{nosyn_i} and v_{nosem_i}. Subtracting them from the original embedding, we get the semantic representation v_{sem_i} and the syntatic representation v_{syn_i}. We also randomly remove the same number of dimensions in the original embedding for comparison.

By comparing the experiment results across different information sets and different layers, we hope to better understand how BERT processes different types of linguistic information throughout the encoding process.

4 Results

We first evaluate our two tasks on the original embeddings, and determine that linear classifiers can successfully predict both CCG tags and semantic tags (around 85% and 89% testing accuracy, respectively), as shown in table 2. We then apply the INLP method to derive the guarding matrices P_{syn} and P_{sem}, which are used to project the original embeddings onto the complements of the syntactic information sub-space and the semantic information sub-space. By applying linear transformations to the original embeddings and their projections, we are able to extract the embeddings described in table 1.

To ensure a fair assessment of the impact of the information loss, we conduct experiments for which we start with the original BERT embeddings and randomly remove the same number of directions that our derived embeddings lost, and train the linear classifiers on these embeddings. The testing accuracies from our experiments can be found in table 2. Curiously, our linear classifiers for evaluation tasks cannot do bettet than majority class.

On the intermediate layers, linear classifiers are generally able to achieve a test accuracy greater than 85% for both CCG tagging and semantics tagging. However, we observe the same majority case accuracy across all layers for each evaluation task. Evaluations of Rand($v_i, |v_{nosyn_i}|$) and Rand($v_i, |v_{nosem_i}|$) result in the same majority class accuracy.

5 Discussion

We are surprised to find out that we are unable to fully remove the syntactic/semantic information from the embeddings by training the linear classifier to make prediction that is no better than the majority, without removing more ranks than BERT's hidden size. However, removing more ranks than BERT's hidden size, whether through the INLP algorithm or randomly, results in a degenerate embedding where every element is reduced to an extremely small magnitude that the linear probe on the evaluation task will only reach the majority class accuracy. This is true on all layers of BERT. This seems to reveal that, the target information is not linearly separable from the original embeddings.

Upon a close inspection of the INLP process and the projections of the original embeddings, v_{nosem} and v_{nosyn}, we realize that, the INLP process continues to run even if it already removes more ranks than BERT's hidden size, which is 768 in our case, because the desired dev accuracy is still not met. Once the rank of the projection matrix reaches the limit, the INLP process simply reduces the magnitude of each elements in the embeddings. In most cases, the process eventually zeroes out the embeddings, which explains the identical yet trivial result we get from the evaluation tasks across all layers.

Embedding	Directions Removed	CCG Tagging	Semantic Tagging
v_{orig}	0	84.75%	88.56%
Majority Guess	N/A	16.57%	22.93%
Rand(v_{orig}, $\|v_{nosem}\|$)	792	16.57%	22.93%
Rand(v_{orig}, $\|v_{nosyn}\|$)	795	16.57%	22.93%
v_{nosem}	792	16.57%	22.93%
v_{nosyn}	795	16.57%	22.93%

Table 2: Experiment Result of Different Embeddings

6 Conclusion

It has been established that linear classifiers are successful in various linguistics probing tasks (Liu et al., 2019). Our experiment has confirmed that linear classifiers can perform CCG tagging and semantic tagging on the Parallel Meaning Bank data set (Abzianidze et al., 2017) with a fairly high rate of success. We then employed INLP to guard the information contained in BERT embeddings that linear classifiers use to perform the aforementioned classification tasks.

Using the INLP-derived guarding functions we were able to explore the importance and separability of the syntactic and semantic information contained in BERT embeddings. We evaluated the classification tasks on various derived embeddings and concluded that not only is the syntactic and semantic information essential for their respective classification tasks, these information sets are also very crucial for the opposing classification tasks as well. Thus the two information sets are not linearly separable from the original embeddings. Attempts to remove the information sets by INLP will result in projection matrices whose ranks are higher than the rank of embeddings. Applying the projection matrices will result in degenerate embeddings where all information is removed.

Our results indicate that besides using the majority class accuracy as the stopping condition, researchers hoping to use INLP to guard information from BERT embeddings should also make sure the loop stops before too many ranks are removed. If the rank of the projection matrix P is higher than the rank of the embedding matrix, only trivial results will be achieved.

Though INLP successfully produces interesting results on various tasks, it is worth noting that our dataset is relatively small compared to the number of parameters in the linear classifier. Reproducing this experiment at a larger scale will be helpful in further validating the experiment results. Additionally, the variety of training and evaluation tasks can be increased for a broader understanding of how syntactic and semantic information is encoded in BERT embeddings.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Adi Ben-Israel. 2015. Projectors on intersections of subspaces. *Contemporary Mathematics*, page 41–50.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence 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 2453–2464, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. Amnesic probing: Behavioral explanation with amnesic counterfactuals.

Julia Hockenmaier and Mark Steedman. 2007. CCG-bank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. 2021. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. In *NAACL*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. *CoRR*, abs/1903.08855.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* ccg parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287. Association for Computational Linguistics.