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

Qingxia Guo, Saiya Karamali, Lindsay Skinner, and **Gladys Wang**
University of Washington
{qg07, karamali, skinnel, qinyanw}@uw.edu

Abstract

Distinctions between Syntax and Semantics are not readily agreed upon. We seek to explore how representations of these two information sets manifest in BERT embeddings. Specifically we investigate the degree of the linear separability of syntactic and semantic information in BERT embeddings, as well as quantify how important the linear component corresponding to one information set is to solving a classification task that targets the other information set. We use Iterative Nullspace Projection to decompose word-level BERT embeddings into syntactic, non-syntactic, semantic and non-semantic components to be used in syntactic and semantic classification tasks. Our results show that there is significant overlap between the syntactic and semantic components in BERT embeddings to the degree that they are not linearly separable. Our 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 construct a linear probing system for a task and then employ Iterative Nullspace Projection (INLP from here on) (Ravfogel et al., 2020) to generate a new embedding devoid of information learned from the probing task. We then measure the performance of this new embedding on downstream syntactic and semantic classification tasks. The design of our probing procedure follows Elazar et al., 2020, which employs INLP to investigate whether BERT uses part-of-speech (POS) information when solving language modeling (LM) tasks.

A novel method 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 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.

To evaluate the separability of syntax and semantics, we use two tasks: one task for the probing system and INLP procedure, and one task for evaluating performance on embeddings before and after INLP. We choose Combinatory Categorical Grammar (CCG from here on) tagging (Hockenmaier and Steedman, 2007) as the syntactic task and semantic category labeling (Bonial et al., 2014) as the semantic task.

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 Iterative Nullspace Projection method to syntactic (CCG) and semantic labeling 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 labeling from word-level BERT embeddings, and test how the removal of these information sets impacts the embeddings' performance on these tasks.

3 Methods

We construct two separate probing tasks to isolate the syntactic and semantic information in word-level BERT embeddings. The embeddings are separated into syntactic and non-syntactic, and semantic and non-semantic components via INLP which is described in section 3.1. These embedding components are then combined to form new embeddings, which are evaluated on the same tasks that were used for probing.

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. In other words, Wx defines a distribution over the set of categories C and we assign x to the class $c \in C$ which is allotted the greatest probability by Wx. Note that the classifier's accuracy must be greater than that achieved by guessing the majority category, otherwise x contains no linear information relevant for the categorization task and thus no guarding function is needed. 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.

However, this process does not necessarily remove all of the linear information in x that could be used to predict c. For example, x may contain redundant information and W may have only used one set of this information for its prediction. In this case, the redundant information would still be present in $\hat{x}$. Thus, we must repeat the above process, defining a new linear classifier $\hat{W}$ that uses $\hat{x}$ to predict c. If $\hat{W}$ is still able to predict c with a greater than majority class guess accuracy, then we know that $\hat{x}$ contained linear information about c. As above, we project $\hat{x}$ onto the null-space of $\hat{W}$ via the projection function P_2 and define a new $\hat{x} = P_2(P_1(x))$.

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. The final $\hat{x} = P_n(P_{n-1}(\dots P_1(x)))$ contains no linear information about the categories in C and we call $P(x) = P_n(P_{n-1}(\dots P_1(x)))$ the guarding function.

We will pair the INLP method with the probing tasks described in sections 3.3 and 3.4 in order to create two guarding functions that will enable us to isolate the linear components of BERT embeddings that contain syntax-specific and semantics-specific information.

3.2 Data

We use the English Parallel Meaning Bank v4.0 (Abzianidze and Bos, 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.

The original dataset does not include CCG tags, however Abzianidze and Bos, 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 the syntactic and semantic probing tasks as desired.

3.3 Syntactic probing task

The syntactic probing task involves training a linear classifier on the final layer BERT embeddings in order to predict the CCG 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 CCG labeling task. For a given embedding, v_{orig}, the projection that results from applying this guarding function, P_{syn}, 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, $v_{nosyn} = P_{syn} v_{orig}$. We can then determine the "syntactic component" of the embedding by taking the difference of the embedding vector with the non-syntactic component, $v_{syn} = v_{orig} - v_{nosyn}$.

3.4 Semantic probing task

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 section, we use the resulting guarding function to decompose the original embedding into a "non-semantic component", $v_{nosem} = P_{sem} v_{orig}$, and a "semantic component", $v_{sem} = v_{orig} - v_{nosem}$.

3.5 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 embeddings and the embeddings created by randomly removing directional information on the CCG and semantic labeling tasks that were used in the probes.

3.6 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

4.1 Final layer evaluation

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% testing accuracy), 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.

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
$v_{sem} = v_{orig} - v_{nosem}$	Semantic representation
$v_{syn} = v_{orig} - v_{nosyn}$	Syntactic representation
Rand(v, n)	embeddings v with n random directions removed
Subscript i	Objects relating to the i-th layer of the BERT architecture rather than the final layer

Table 1: Description of Embeddings

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. For the embeddings derived from projection matrices, we record the number of removed directions using $Null(P)$, where P is the projection matrix. [1] Then we create embeddings with the same number of directions randomly removed and train the linear classifiers on these embeddings. The testing accuracies from our experiments can be found in table 2.

[1] For embeddings not derived from matrix multiplication, we obtain the number from $Null(M)$ where M is the embedding matrix with size (768,instance number). This is not necessarily equivalent to the number of direction removed. $Null(M) \geq Null(P)$ for the embedding matrix M that corresponds to projection P, but in practice the numbers are very close. With this approach we err on the side of removing more random directions from our random embeddings, resulting in a more conservative comparison when assessing our derived embeddings. Also note that we decided not to include these types of embeddings in our final assessment due to concerning behavior we witnessed after performing the linear transformation to obtain the embeddings. Specifically, the rank of the resulting embedding matrix did not match our expectations. This may be due to a rounding or an issue with the orthonormalization process, but we were unable to confirm. So we decided to remove all embeddings derived in this manner from our evaluation process until we are able to determine the source of this behavior. This means that v_{syn} and v_{sem} will not be included in our final evaluation.

Embedding	Directions Removed	CCG Tagging	Semantic Tagging
v_{orig}	0	84.75%	88.56%
Rand(v_{orig}, $\|v_{nosem}\|$)	77	84.06%	87.71%
Rand(v_{orig}, $\|v_{nosyn}\|$)	159	82.26%	87.57%
v_{nosem}	77	27.93%	17.28%
v_{nosyn}	159	23.76%	49.43%

Table 2: Experiment Result of Different Embeddings

4.2 Intermediate Layer Evaluation

Linear classifiers are generally able to achieve a test accuracy greater than 90% for both CCG tagging and semantics tagging. However, after applying the INLP algorithm and projecting the layer-wise embeddings to the nullspaces of the information sets, we observe that we get the same results across all layers for each evaluation task, with the accuracy being the majority class of the evaluation task. Evaluations of Rand(v_i, $|v_{nosyn_i}|$) and Rand(v_i, $|v_{nosem_i}|$) result in the same majority class accuracy.

Upon a close inspection of the INLP process and the projections of the original intermediate layer embeddings, v_{nosem_i} and v_{nosyn_i}, 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 some cases, the process eventually zeroes out the embeddings, which explains the identical yet trivial result we get from the evaluation task.

5 Discussion

The INLP method successfully guards the information used in the probing task. When performing CCG tagging, the model accuracy drops significantly when comparing the v_{nosyn} embeddings with the Rand($|v_{nosyn}|$) embeddings (from 82.26% to 23.76%), and similarly for semantic tagging (from 87.71% to 17.28%). In contrast, when the directions are randomly removed, the performance remains relatively the similar to the classifiers' performances on the original BERT embeddings for both tasks (84.75% and 88.56%, respectively). This suggests that the syntactic and semantic information contained in the original BERT embeddings is not highly concentrated and that removing a small amount of one information set will not have a significant impact on the classification task.

To measure the importance of semantic information for the syntactic task, we compare the performance of the CCG tagging classifier on v_{nosem} with that of Rand(v_orig, $|v_{nosem}|$) and see a performance drop of 56.13%. Similarly, we can measure the importance of syntactic information for the semantic task by comparing the performance of the semantic tagging classifier on v_{nosyn} with that of Rand(v_{orig}, $|v_{nosyn}|$) and see a performance drop of 38.14%. We can see that the removal of each information set has a significant impact on the performance of a linear classifier trained on either classification task. This suggests that the syntactic and semantic information in BERT embeddings is not easily disentangled.

What is surprising is that the loss of semantic information has a more significant impact on the syntactic classification task than the loss of syntactic information does on the semantic tagging task. These results suggest that the semantic task is less dependent on syntactic information than the opposite direction, which challenges the assumption that syntactic information is more pertinent to semantic comprehension than vice versa.

As an unexpected finding of our experiment, we are unable to fully remove the syntactic/semantic information from the intermediate embedding by training the linear classifier to make prediction that is no better than the majority, without removing more ranks than BERT's hidden size, on all intermediate layers of BERT. 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 seems to reveal that, on the intermediate layers of BERT, the target information is not linearly separable from the original embedding. 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 removing too many ranks for non-trivial results.

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 and Bos, 2017) with a fairly high rate of success. We then employed INLP and successfully guarded 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 linear syntactic and semantic information essential for their respective classification tasks, these information sets are also very important for the opposing classification tasks as well. Thus the two information sets are not easily separated. Somewhat surprising, we determined that the semantic information was more influential on the success of the syntactic classifier than the other way around.

By performing a similar analysis on embeddings derived from different layers of BERT architecture, we discovered that the syntactic information set and semantic information set are not linearly separable from the original intermediate layer embeddings. Attempting to completely remove these information sets will remove too much that the embeddings become degenerate.

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. Reproduing 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 and Johan Bos. 2017. Towards universal semantic tagging. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS 2017) – Short Papers*, pages 1–6, Montpellier, France.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

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. CCGbank: 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*.

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.

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.