Proposal for Analyzing Language Models: Separatablity of Syntax and Semantics

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

1 Introduction

The boundary between semantics and syntax has always been a hotly debated topic in linguistics since generative syntax, but do NLP models make such a distinction? The objective of this project is to explore BERT's reliance on certain syntactic information when handling a semantic task, and vice versa. To achieve our goal, we will construct a linear probing system for a task and then employ Iterative Nullspace Projection (INLP) (Ravfogel et al., 2020) to generate a new embedding devoid of information learned from probing task. Then we will measure the performance of another task on this new embedding.

The design of our probing procedure follows Elazar et al., 2020, who employed Iterative Nullspace Projection (INLP from here) to investigate whether BERT uses part-of-speech (PoS) information when solving language modeling (LM) tasks. A novel method for removing information from a embedding, INLP iteratively trains a linear model on a specific task, and projects the input on the nullspace of the linear model. 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. Then we can compare the performance of a semantic task in whole space and only in the non-syntactic space to see if BERT is using syntactic information when performing the semantic task. Conversely, we can also first probe a semantic task and then measure with a syntactic task.

To evaluate the separability of syntax and semantics, we will need two task, one task for the probing system and INLP procedure, one task for evaluating performance on embeddings before and after INLP. We choose Combinatory Categorical Grammar (CCG) tagging as the syntactic task and Semantic Role Labeling (SRL) as the semantic task. Section 2 provides description of the tasks and the construction of the experiment pipeline. Section 3 provides possible results and interpretations. Finally, section 4 provides project management details including timeline and division of labor.

2 Methods

We construct two separate probing tasks to isolate the syntactic and semantic information in word-level BERT embeddings(Devlin et al., 2019). The embeddings are separated into syntactic and non-syntactic, and semantic and non-semantic components via Iterative Null-Space Projection (INLP), which is described in section 2.1. These embedding components are then combined to form new embeddings, which are evaluated on the same tasks that were used for probing. If time permits, we will also evaluate how these new embeddings perform on the language modeling task.

2.1 The Iterative Null-Space Projection method

The Iterative Null-Space Projection method (INLP from here on) 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 its affiliated category, $c \in C$. 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 random chance, 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 random chance 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 random chance. 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 2.3 and 2.4 in order to create two guarding functions that will enable us to isolate components of BERT embeddings that contain syntax-specific and semantics-specific information.

2.2 Data

We will use the English V4 subset of the CoNLL 2012 shared task data(Pradhan et al., 2012) . We must perform two pre-processing steps for this data to be used for our probing and evaluation tasks. The first is to apply Hockenmaier and Steedman (2007)'s CCG Derivation algorithm to the parse tree field in the dataset, in order to create the CCG tags for each word. If this proves to be too time-consuming or computationally expensive then we shall change the syntactic probing task to utilize the POS tags available in the dataset, in place of CCG tags. The second task is to use the SRL frames in the dataset to generate (verb, BIO-argument-tag) pairs that will act as the categories for the semantic probing task.

2.3 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, the projection that results from applying this guarding function 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. We can then determine the "syntactic component" of the embedding by taking the difference of the embedding vector with the non-syntactic component.

2.4 Semantic probing task

Similar to the above, the semantic probing tasks involves training a linear classifier on the final layer BERT embeddings in order to predict the semantic role tag (described in the data section) associated with each word. This classifier is used in the INLP algorithm in order to create a guarding function for the information necessary to complete the SRL labeling task. For this particular task, it is possible that a single word will have multiple SRL tags associated with it. In this case, we treat each (vector, SRL tag) pair as a single example in the dataset, in order to create a guarding function that works across all of the SRL tags affiliated with a particular word. As described in the Syntactic probing task section, we shall use the resulting guarding function to decompose the original embedding into a "semantic component" and a "non-semantic component".

2.5 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.

The new embeddings to be tested include the syntactic component, the non-syntactic component, the semantic component and the non-semantic component derived from the probing tasks. Additionally,

we can create an embedding that contains syntactic information and removes semantic information by linearly projecting the syntactic component onto the non-semantic component. Using a similar process, we can create an embedding that contains semantic information and removes the syntactic information present. Finally, we can create an embedding that contains the semantic information captured by the syntactic component, by linearly projecting the syntactic component onto the semantic component. Similarly, we can create an embedding that contains the syntactic information captured by the semantic component.

We will assess each of these embedding types and the original BERT embeddings on the CCG and SRL labeling tasks that were used in the probes. We have also hypothesized several additional assessment tasks that we would like to undertake, if time permits, or relegate to future work. The first of these tasks is to assess how each embedding type performs on the language modeling task. We would also like to perform the evaluation classification task using a feed-forward neural network with a single hidden layer that contains 10 nodes, in order to determine if there is any task-relevant non-linear information present in the embeddings. If time permits, we would also like to look for patterns in the performance of different embeddings, e.g. explore if a particular embedding type tends to perform better/worse on one of the evaluation tasks for words of a particular POS compared to others. Finally, if time permits we would like to repeat the above procedure to explore the embeddings output by different layers of the BERT model.

3 Possible Results

In this section, we consider what each evaluation task tells us about the embeddings that are being probed. When we isolate the syntactic component and run it on the syntactic and semantic tasks, we learn how successfully the component responsible for CCG tagging has been isolated, and we also learn how effective the syntactic component alone is on the semantic task. Similarly, running the semantic component on the evaluation tasks tells us how well we've isolated the semantic component and how effective it is on the syntactic task. If the removing of syntactic information leads to a insignificant decrease in the model's performance in a semantic task (or removing semantic information leads to insignificant decrease in performance in a syntactic task), we can conclude that syntactic and semantic information in BERT's representation is linearly separable. Conversely, an significant decrease in performance will indicate low separability between syntactic and semantic information in BERT 's representation.Finally, running the non-syntactic and non-semantic components on the evaluation tasks tells us whether any information not identified by INLP is at all useful for the evaluation tasks.

Next, we consider the potential results of the various projected word embeddings on the evaluation tasks. Each of these tell us how much overlap there is between the syntactic and semantic components of the contextual word embeddings. Projecting the syntactic component onto the non-semantic component removes semantic information from the syntactic component, and projecting the semantic component onto the non-syntactic component removes syntactic information from the semantic component. Projecting the syntactic component onto the semantic component gives us the semantic information that is also part of the syntactic component, and projecting the semantic component onto the syntactic component gives us the syntactic information that is also part of the semantic component. Running all of these embeddings un the semantic and syntactic tasks tells us how separated the semantic and syntactic components are, and how important the overlapping portions are to each task.

4 Divsion of Labor and Timeline

There are three main parts of this project, coding the experiment, presenting related topics in class and writing the final paper. Each part has detailed requirements and been assigned to a member to take charge of as presented in table 1. It is hard to anticipate the time consumed and difficulty of each task. We have set up a regular weekly meeting to update the progress of each part weekly so that everyone is on the same page with the development of the project.

References

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 Tech-*

Duty	Details	Who is in Charge
Coding the experiment	data preprocessing	Qingxia
	building probing system & INLP	Saiya & Qingxia
	building evaluation system	Gladys
	merging subparts together	Saiya
	running different task pairs	Saiya
Presenting related topic	presenting INLP method	Lindsay & Gladys
Writing paper	method section	Lindsay
	result section	Saiya
	discussion/conclusion section	all

Table 1: Role Description

nologies, 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.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

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