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

Abstract

This document contains the instructions for preparing a manuscript for the proceedings of ACL 2020. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the directions reported in this document.

1 Introduction

The following instructions are directed to authors of papers submitted to ACL 2020 or accepted for publication in its proceedings. All authors are required to adhere to these specifications. Authors are required to provide a Portable Document Format (PDF) version of their papers. **The proceedings are designed for printing on A4 paper.**

2 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 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 func-

tion.

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. We must perform two preprocessing steps for this data to be used for our probing and evaluation tasks. The first is to apply Hockenmaier and Steedman'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. 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

As reviewing will be double-blind, papers submitted for review should not include any author information (such as names or affiliations). Furthermore, self-references that reveal the author's identity, *e.g.*,

> We previously showed (**?**) ...

should be avoided. Instead, use citations such as

> **?** previously showed...

Please do not use anonymous citations and do not include acknowledgements. **Papers that do not conform to these requirements may be rejected without review.**

Any preliminary non-archival versions of submitted papers should be listed in the submission form but not in the review version of the paper. Reviewers are generally aware that authors may present preliminary versions of their work in other venues, but will not be provided the list of previous presentations from the submission form.

Once a paper has been accepted to the conference, the camera-ready version of the paper should include the author's names and affiliations, and is allowed to use self-references.

LaTeX-specific details: For an anonymized submission, ensure that `\aclfinalcopy` at the top of this document is commented out, and that you have filled in the paper ID number (assigned during the submission process on softconf) where `***` appears in the `\def\aclpaperid{***}` definition at the top of this document. For a camera-ready submission, ensure that `\aclfinalcopy` at the top of this document is not commented out.

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

A Appendices

Appendices are material that can be read, and include lemmas, formulas, proofs, and tables that are not critical to the reading and understanding of the paper. Appendices should be **uploaded as supplementary material** when submitting the paper for review. Upon acceptance, the appendices come after the references, as shown here.

LaTeX-specific details: Use `\appendix` before any appendix section to switch the section numbering over to letters.

B Supplemental Material

Submissions may include non-readable supplementary material used in the work and described in the

paper. Any accompanying software and/or data should include licenses and documentation of research review as appropriate. Supplementary material may report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.

Nonetheless, supplementary material should be supplementary (rather than central) to the paper. **Submissions that misuse the supplementary material may be rejected without review.** Supplementary material may include explanations or details of proofs or derivations that do not fit into the paper, lists of features or feature templates, sample inputs and outputs for a system, pseudo-code or source code, and data. (Source code and data should be separate uploads, rather than part of the paper).

The paper should not rely on the supplementary material: while the paper may refer to and cite the supplementary material and the supplementary material will be available to the reviewers, they will not be asked to review the supplementary material.