What does BERT know about distributivity?

Stanford CS224N Custom Project

J. Adolfo Hermosillo
Department of Linguistics
Stanford University
jadolfoh@stanford.edu

Jiayi Lu
Department of Linguistics
Stanford University
jiayi.lu@stanford.edu

Leyla Kursat
Symbolic Systems Program
Stanford University
lkursat@stanford.edu

Abstract

Our goal is to test whether transformer models are able to encode event structural information in their representations of sentences. Using [1]'s event classification dataset, we seek to evaluate whether and how BERT may encode information about distributivity. We train classifiers to predict whether a predicate is distributive or collective using each of BERT's hidden layers contextualized representations of a complete predicate, a predicate span, and an argument span. In line with previous research that suggest that semantic information is encoded in the top layers, we find that BERT may encode information about distributivity in later layers but it can still find a signal as early as layer 1. We also report that while all three types of representation perform in a similar fashion, argument span representations may better encode information about distributivity.

1 Key Information to include

- Mentor: Benjamin Newman
- External Collaborators: None
- Sharing project: No

2 Introduction

As deep pre-trained language models continue to set new state-of-the-art results on NLP benchmarks, questions regarding their potential to encode linguistic knowledge have been raised. In this project, we investigate whether BERT [2] encodes information about the structure of events a using a probing task. A probe consists of a simple classification task that seeks to reveal information about a linguistic phenomenon using as features the learned parameters of an external model trained on a different task[3]. Probes are trained to predict properties from representations of language and are used to investigate whether learned representations of language encode information about a particular feature of language [4]. A successful classifier may suggest that the external model stores information about the linguistic phenomenon of interest. We focus on examining whether the contextual representations for the plural or conjoined arguments of certain predicates encode information about distributivity. A predicate is interpreted distributively if an event is individually true of each participant or collectively otherwise [5, 6, 7, 8]. For example, sentence (1a) describes *multiple* laughing events where each child is the sole participant of their own laughing. In contrast, sentence (1b) describes *one* meeting event that involves multiple children.

(1) a. The children laughed,
b. The children met.

The laughing event described in (1a) has the property of being distributive: each participant in the subject noun phrase *the children* engages in their own laughing event separately, the event is true of

each participant. On the other hand, the meeting event described in (1b) has the property of being collective: all participants described by the subject noun phrase *children* mandatorily engage in the meeting event collectively, true only of the collective. Human language users, even without being explicitly taught what distributivity means, can usually spot the event-structural difference between two events described by these sentences in (1) [1]. We examine whether and how BERT is able to discern information about this semantic feature through a series of experiments.

3 Related Work

Previous work in using probes to explore language representations have focused on finding signs for morphology [9], part-of-speech [10], sentence length [11], and syntax [12]. Recent work by [13] is a great example of the success of growing research in using probing as a method to evaluate whether neural networks preserve certain linguistic properties in their representations of sentences. This paper offers a novel structural probe to test whether deep contextual models embed syntactic parsed trees in their learned representations. Using a linear transformation of the learned vector space, they test whether, when squared, the L2 distance between two word vectors corresponds to the number of edges between these words in their parse tree. They evaluate representations from BERT (BASE and LARGE) and ELMo at different layers on how well the predicted distances between pairs of words resemble gold parse trees distance metrics. When it comes to syntax, $BERT_{LARGE}$ performs better than $BERT_{BASE}$, which performs better than ELMo. Specific to BERT, layers after the middle tend to encode information about how words are placed with respect to others, which in this case is taken as proxy for syntactic knowledge. They also find, for the best performing models, that when the size of k in $B \in \mathbb{R}^{k \times m}$ is larger than 64, parsing accuracy converges for all models.

Another interesting paper that focuses on finding an encoding of language structure is the work by [14] that uses probing tasks to assess individual BERT layers in their ability to capture different types of linguistic features. They find that surface information such as sentence length are embedded in lower levels, syntactic information such as depth of syntactic tree are found in middle layers and that semantic information such as tense and subject number are embedded in top layers of BERT. This paper shows the hierarchy with which linguistic information is encoded in layers of contextualized language models. Despite the abundance of past linguistics literature on event classification [15, 16, 17], there has not been an attempt to test whether deep language models are able to encode event structural information in their representations of sentences.

4 Approach

To examine whether and how BERT encodes information about distributivity, we build a classifier to predict whether a predicate has the distributive feature (see Eq. 1). Our classifier consists of a fully connected feed-forward network with one hidden layer (*h*) of size 128 with a ReLU activation in between. The classifier takes as input a contextual embedding of size *d* and predicts whether the predicate represented with the vector is distributive or collective.

$$Distributivity_{classifier}(x) = Softmax(Linear(ReLU(Linear(x))) \tag{1}$$

5 Experiments

5.1 Data

We use a subset of the Event Structure dataset [1] to train our classifiers. The Event Structure is part of the Universal Decompositional Semantics (USD) by Decompositional Semantics Initiative. As the largest collection of annotation of event structure, this dataset covers the English Web Treebank and includes annotations for event structural distinctions such as (a) the substructure of an event, (b) superstructure in which an event takes part and (c) the relationship between an event and it's participants as well as other related properties such as dynamicity, telicity and durativity. To answer whether BERT encodes information about distributivity, we focus on the event-entity subset of the data. This subset of the data consists of predicate-argument pairs with plural or conjoined arguments from the English Web Tree Bank. Plural arguments were identified using the attribute NUMBER

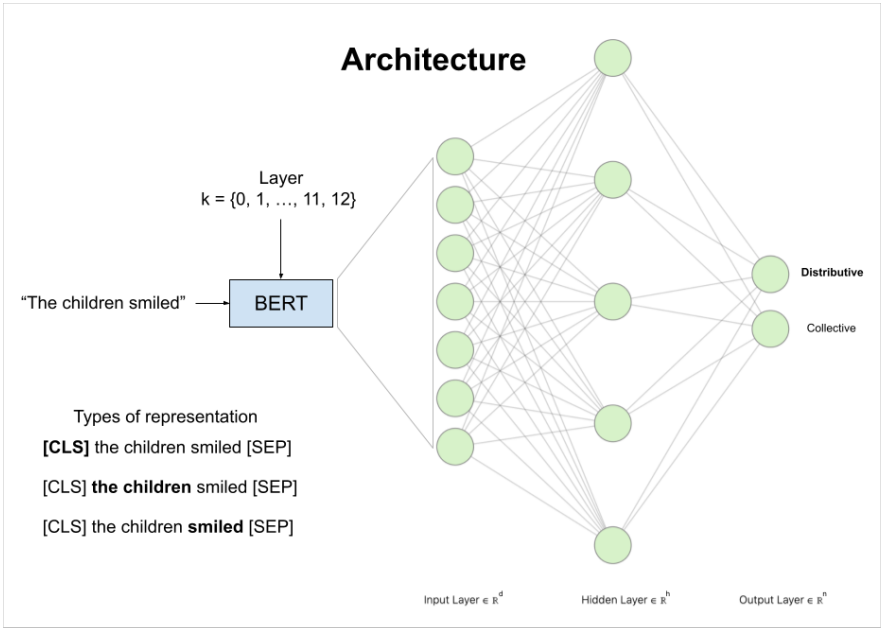

Figure 1: Probing protocol: A predicate is fed to BERT, then a layer and a type of representation are chosen, the output is then an input to a fully connected FFN for binary classification.

Split	Distributive	Collective	Total
Train	4341	4537	8878
Dev	589	415	1004
Test	542	448	990
Total	5472	5400	10872

Table 1: Number of predicates per dataset split

from the Universal Dependencies. Conjoined arguments were identified using the `conj` dependency between a head and a noun. Only arguments with one of the following universal dependencies: `nsubj`, `nsubjpass`, `dobj`, and `iobj` were considered for either type of predicate-argument pair.

5.2 Evaluation method

The development and test sets for the distributive-collective dataset from [1] were annotated three times, for our purposes, we take the majority label in our experiments. We used three metrics metrics to compare our models: F1 score, accuracy and weighted F1 score. We were interested in seeing how well our models identify true cases (accuracy), and we also wanted to get a predictive metric of precision and recall combined (F1 score). Our final evaluation metric was weighted F1 score: we had different number of distributive and collective samples and we calculated the F1 score separately for each group, and then weighted the score by its support.

5.3 Experimental details

In order to investigate how BERT encodes information about distributivity, we run three main experiments that vary the type of input representation. The motivation behind is that the embeddings associated with a token change depending on their context. We thus train classifiers with a predicate representation, and contextualized predicate and arguments spans in the predicate (see figure 1). In experiment 1, the representation for a predicate consists of the classification embedding computed using BERT's self-attention mechanism ([CLS]), the *Contextual* model. In experiment 2, the representation

for a predicate consists of the averaged contextualized embeddings for the tokens that correspond to the argument span in the predicate, the *Argument*$_{span}$ model. In experiment 3, we follow the same logic in experiment 2 but with predicate spans, the *Predicate*$_{span}$ model. For each experiment, we train 13 classifiers, one for each of BERT layers (including the token layer). This set up will allows to investigate 1) where in a predicate and 2) where in BERT is information about distributivity encoded. For training, we use Adam optimization with default parameters and an optimal learning rate of 0.0001 (we experimented with 0.1, 0.01, 0.001, 0.0001, 0.00001), an effective batch size of 128, and minimize cross entropy loss. We run the models for 30 epochs and use early stopping after 5 validation steps. Each classifier is run 5 times, we report the best performing run. We compare these models against a baseline which consists of a randomly initialized word model, where the input is a sentence embedding vector, which is computed by averaging the vectors for each word in a predicate.

5.4 Results

Our results suggest that our classifier is able to classify contextualized embeddings as distributive or collective relatively well (see Table 2). When compared to our baseline, we see that BERT-based models outperformed the baseline. Our best performing model was $Argument_{span}$ model, however, differences between BERT-base models were not that striking (see figure 2). In general, performance increased for later layers with some model specific variations (see Appendix). The $Contextual$ model shows a relatively stable upward trend, where performance appears to increase as we move into later layers. A similar patters appears to be occurring with the $Predicate_{span}$ model, however, we see a peak at around layer 7, and then performance slightly drops. The $Argument_{span}$ model shows a slightly more volatile behavior. It shows two peaks at around layer 4 and 8 and then again continues to increases in performance in the last two layers. We see that in general, the $Argument_{span}$ model performs the best, followed by the $Contextual$ model, which is in turn followed by the $Predicate_{span}$ model. Finally, a striking difference between BERT-based models lied in layer 0. In this layer, $Contextual$ performed at par with the $Baseline$.

Model Performance			
Representation	F1	Accuracy	Weighted F1
$Baseline$	70.76	54.75	38.74
$Contextual$	77.85	74.04	73.93
$Argument_{span}$	78.28	74.34	74.37
$Predicate_{span}$	77.46	73.17	73.13

Table 2: Best Performing Model per representation

6 Analysis

We found that probe performance was high even in the first layer of the network and improved in deeper layers, which suggests that BERT manage to learn some distributivity information as early as the first layer. Best performance was reached in the 8^{th} layer, which follows previous findings that suggest semantic information is usually encoded in later BERT layers [14]. Regardless of the type of input, the representation stores some information about whether an event is performed by participants collectively or separately by each participant, which may be due to the contextual information present in each token vector. The three models performed similarly, and better than the baseline across the three evaluation metrics: F1, Accuracy and Weighted F1. The contrast between the baseline and the critical conditions is the biggest along the weighted F1 metric. This is probably because the baseline simply treats all sentences as distributive regardless of their actual distributivity, which leads to a high unweighted F1 and accuracy score due to the condition imbalance in the data set. When weighted F1 score is used, this confound is controlled for. The representation that performed the best was the $Argument_{span}$ model. This suggests that the argument span may encode the most information about distributivity, although distributivity may also be encoded in the predicate span or the complete predicate made up of the predicate and argument spans.

Figure 2: Weighted F1 scores for baseline and each BERT based model

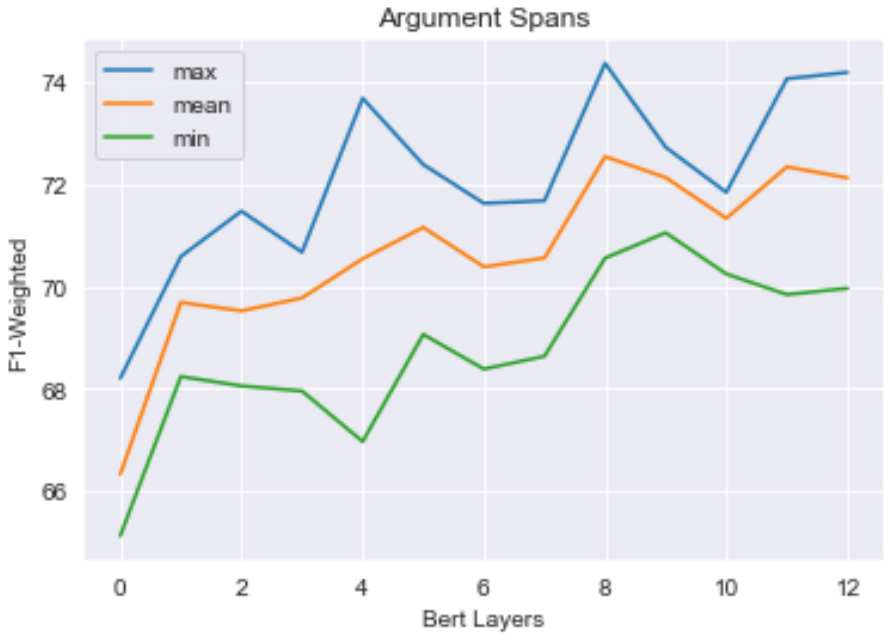

Figure 3: Maximum, mean, and minimum Weighted F1 scores from the 5 runs for argument span model across layers

7 Conclusion

In this study, we aimed to test whether we can recover distributivity information from the outputs of BERT. By creating a binary classifier probe trained in the distributivity dataset in [6], we evaluate the probe performance when it takes in outputs from BERT. We observed that BERT embeddings for argument span, verb span, and contextualized sentence embeddings all lead to better probe performance than the randomly initialized baseline. The best performance is achieved at BERT's

8^{th} layer. Overall, our results suggest that BERT can encode distributivity information in its outputs, more so in later layers. We also find that the signal about distributivity can be recovered slightly better from the arguments in a predicate than from the predicate span (which may include verbs or adjectives) or the complete predicate (which includes the predicate and argument spans).

Given the binary nature of our inputs, we were unable to tease apart how ambiguous predicates are handled. For example, in "Mary and John opened the window", how are Mary and John involved in the opening event? Do they both, collectively, participate, or do they each do their part? these questions we leave for future work. Further, we would like to exploring whether deep language models can learn other information about event structure, such as event-event relations or event subevent relations. The dataset we used in our project includes other event structural information such as these and we can create corresponding probes in similar ways to the current study. This work can also be extended to probing the output of other deep language models like RoBERTa and T5.

References

[1] William Gantt, Lelia Glass, and Aaron Steven White. Decomposing and recomposing event structure. *Transactions of the Association for Computational Linguistics*, 10:17–34, 2022.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[4] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[5] Godehard Link et al. The logical analysis of plurals and mass terms: A lattice-theoretical approach. *Formal semantics: The essential readings*, pages 127–146, 1983.

[6] David Dowty et al. Collective predicates, distributive predicates, and all. In *Proceedings of the 3rd ESCOL*, pages 97–115. (Eastern States Conference on Linguistics), Ohio State University Ohio, 1987.

[7] Yoad Winter. Distributivity and dependency. *Natural language semantics*, 8(1):27–69, 2000.

[8] Lelia Montague Glass. *Distributivity, lexical semantics, and world knowledge*. Stanford University, 2018.

[9] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*, 2018.

[10] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*, 2017.

[11] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

[12] Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1526–1534, 2016.

[13] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word 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 4129–4138, 2019.

[14] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[15] Zeno Vendler. Verbs and times. *The philosophical review*, 66(2):143–160, 1957.

[16] George Philip Lakoff. *On the nature of syntactic irregularity*. Indiana University, 1966.

[17] Emmon Bach. The algebra of events. *Linguistics and philosophy*, pages 5–16, 1986.

A Acknowledgments

We would like to thank our mentor Ben Newman for his valuable feedback through the development of this project.

B Appendix (optional)

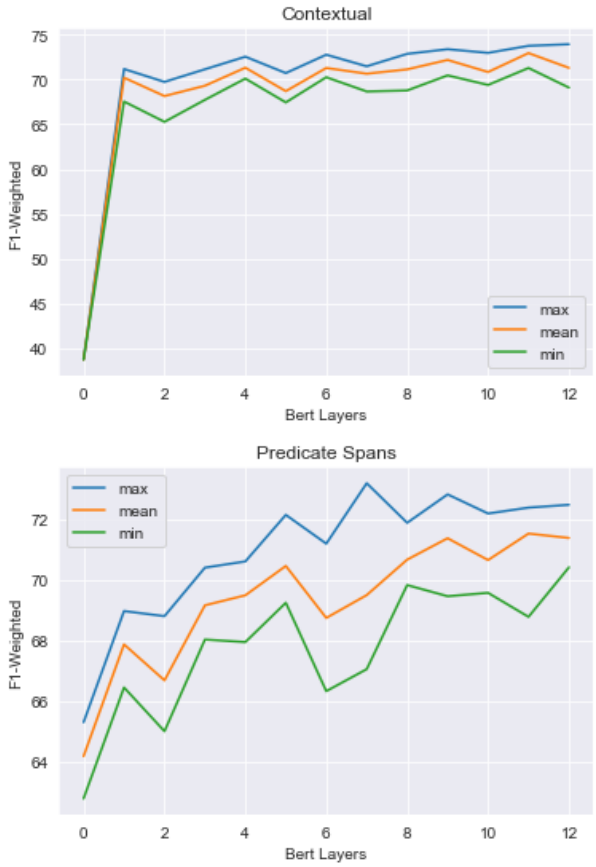

Figure 4: Maximum, mean, and minimum Weighted F1 scores from the 5 runs for each the BERT-based models across layers