

Quantifying the Contextualization of Word Representations with Semantic Class Probing

Mengjie Zhao[†], Philipp Dufter[†], Yadollah Yaghoobzadeh[‡], Hinrich Schütze[†]

[†] CIS, LMU Munich, Germany [‡] Microsoft Turing, Montréal, Canada

mzhao@cis.lmu.de

Abstract

Pretrained language models achieve state-of-the-art results on many NLP tasks, but there are still many open questions about how and why they work so well. We investigate the contextualization of words in BERT. We quantify the amount of contextualization, i.e., how well words are interpreted in context, by studying the extent to which *semantic classes* of a word can be inferred from its contextualized embedding. Quantifying contextualization helps in understanding and utilizing pretrained language models. We show that the top layer representations support highly accurate inference of semantic classes; that the strongest contextualization effects occur in the lower layers; that local context is mostly sufficient for contextualizing words; and that top layer representations are more task-specific after finetuning while lower layer representations are more transferable. Finetuning uncovers task-related features, but pretrained knowledge about contextualization is still well preserved.

1 Introduction

Pretrained language models like ELMo (Peters et al., 2018a), BERT (Devlin et al., 2019), and XLNet (Yang et al., 2019) are top performers in NLP because they learn contextualized representations, i.e., representations that reflect the interpretation of a word in context as opposed to its general meaning, which is less helpful in solving NLP tasks. As stated, pretrained language models contextualize words, is clear *qualitatively*; there has been little work on investigating contextualization, i.e., to which extent a word can be interpreted in context, *quantitatively*.

We use BERT (Devlin et al., 2019) as our pretrained language model and quantify contextualization by investigating how well BERT infers **semantic classes** (s-classes) of a word in context, e.g., the s-class *organization* for “Apple” in “Apple

stock rises” vs. the s-class *food* in “Apple juice is healthy”. We use s-class inference as a proxy for contextualization since accurate s-class inference reflects a successful contextualization of a word: an effective interpretation of the word in context.

We adopt the methodology of probing (Adi et al., 2016; Shi et al., 2016; Belinkov et al., 2017; Liu et al., 2019; Tenney et al., 2019b; Belinkov and Glass, 2019; Hewitt and Liang, 2019; Yaghoobzadeh et al., 2019): diagnostic classifiers are applied to pretrained language model embeddings to determine whether they encode desired syntactic or semantic features.

By probing for s-classes we quantify directly where and how contextualization happens in BERT. E.g., we find that the strongest contextual interpretation effects occur in the lower layers and that the top two layers contribute little to contextualization. We also investigate how the amount of context available affects contextualization.

In addition, since pretrained language models in practice need to be finetuned on downstream tasks (Devlin et al., 2019; Peters et al., 2019), we further investigate the interactions between finetuning and contextualization. We show that the pretrained knowledge about contextualization is well preserved in finetuned models.

We make the following **contributions**: (i) We investigate how accurately BERT interprets words in context. We find that BERT’s performance is high (almost 85% F_1), but that there is still room for improvement. (ii) We quantify how much each additional layer in BERT contributes to contextualization. We find that the strongest contextual interpretation effects occur in the lower layers. The top two layers seem to be optimized only for the pre-training objective of predicting masked words (Devlin et al., 2019) and only add small increments to contextualization. (iii) We investigate the amount of context BERT needs to exploit for interpreting a

GloVe	BERT
suits	suits
lawsuit	suited
filed	lawsuit
lawsuits	##suit
sued	lawsuits
complaint	slacks
jacket	47th

Table 1: Nearest neighbors of “suit” in GloVe and in BERT (BERT-base-uncased) wordpiece embeddings

word and find that BERT effectively integrates local context up to five words to the left and to the right (a 10-word context window). (iv) We investigate the dynamics of BERT’s representations in finetuning. We find that finetuning has little effect on lower layers, suggesting that they are more easily transferable across tasks. Higher layers are strongly changed for word-level tasks like part-of-speech tagging, but less noticeably for sentence-level tasks like paraphrase classification. Finetuning uncovers task-related features, but the knowledge captured in pretraining is well preserved. We quantify these effects by s-class inference performance.

2 Motivation and Methodology

The key benefit of pretrained language models (McCann et al., 2017; Peters et al., 2018a; Radford et al., 2019; Devlin et al., 2019) is that they produce contextualized embeddings that are useful in NLP. The top layer contextualized word representations from pretrained language models are widely utilized; however, the fact that pretrained language models implement a *process of* contextualization – starting with a completely uncontextualized layer of wordpieces at the bottom – is not well studied. Table 1 gives an example: BERT’s wordpiece embedding of “suit” is not contextualized: it contains several meanings of the word, including “to suit” (“be convenient”), lawsuit, and garment (“slacks”). Thus, there is no difference in this respect between BERT’s wordpiece embeddings and uncontextualized word embeddings like GloVe (Pennington et al., 2014). Pretrained language models start out with an uncontextualized representation at the lowest layer, then gradually contextualize it. This is the process we analyze in this paper.

For investigating the contextualization process, one possibility is to use word senses and to tap resources like the WordNet (WN) (Fellbaum, 1998) based word sense disambiguation benchmarks of the Senseval series (Edmonds and Cotton, 2001;

	words	comb’s	contexts
train	35,399	62,184	2,178,895
dev	8,850	15,437	542,938
test	44,250	77,706	2,722,893

Table 2: Number of words, word-s-class combinations, and contexts per split in our probing dataset. Appendix §A.6 shows the 34 s-classes and statistics per class.

Snyder and Palmer, 2004; Raganato et al., 2017). However, the abstraction level in WN sense inventories has been criticized as too fine-grained (Izquierdo et al., 2009), providing limited information to applications requiring higher level abstraction. Various levels of granularity of abstraction have been explored such as WN domains (Magnini and Cavaglià, 2000), supersenses (Ciarmita and Johnson, 2003; Levine et al., 2019) and basic level concepts (Beviá et al., 2007). In this paper, we use semantic classes (s-classes) (Yarowsky, 1992; Resnik, 1993; Kohomban and Lee, 2005; Yaghoobzadeh et al., 2019) as the proxy for the meaning contents of words to study the contextualization capability of BERT. Specifically, we use the Wikipedia-based resource for Probing Semantics in Word Embeddings (Wiki-PSE) (Yaghoobzadeh et al., 2019) which is detailed in §3.1.

3 Probing Dataset and Task

3.1 Probing dataset

For s-class probing, we use the s-class labeled corpus Wiki-PSE (Yaghoobzadeh et al., 2019). It consists of a set of 34 s-classes, an inventory of word→s-class mappings and an English Wikipedia text corpus in which words in context are labeled with the 34 s-classes. For example, contexts of “Apple” that refer to the company are labeled with “organization”. We refer to a word labeled with an s-class as a word-s-class combination, e.g., “@apple@-organization”.¹

The Wiki-PSE text corpus contains >550 million tokens, >17 million of which are annotated with an s-class. Working on the entire Wiki-PSE with BERT is not feasible, e.g., the word-s-class combination “@france@-location” has 98,582 contexts. Processing all these contexts by BERT consumes significant amounts of energy (Strubell et al., 2019; Schwartz et al., 2019) and time. Hence for each word-s-class combination, we sample a maximum of 100 contexts to speed up our experiments.

¹In Wiki-PSE, s-class-labeled occurrences are enclosed with “@”, e.g., “@apple@”.

Algorithm 1 Train a classifier with type-level embeddings

```

1: procedure TYPESECLSTRAINER(Dict: word2vec, Dict:
   word2sclass, sclass:  $\mathcal{S}$ , List: TrainWords):
2:   PosVecs, NegVecs = [], []
3:   for word  $\in$  TrainWords do
4:     vector = word2vec.get(word)
5:     sclasses = word2sclass.get(word)
6:     if  $\mathcal{S} \in$  sclasses then
7:       PosVecs.append(vector)
8:     else
9:       NegVecs.append(vector)
10:  classifier = Classifier()
11:  classifier.train(PosVecs, NegVecs)
12:  return classifier

```

Figure 1: Training a diagnostic classifier with uncontextualized word representations for an s-class \mathcal{S} .

Wiki-PSE provides a balanced train/test split; we use 20% of the training set as our development set. Table 2 gives statistics of our dataset.

3.2 Probing for semantic classes

For each of the 34 s-classes in Wiki-PSE, we train a binary classifier to diagnose if an input embedding encodes information for inferring the s-class.

3.2.1 Probing uncontextualized embeddings

We make a distinction in this paper between two different factors that contribute to BERT’s performance: (i) a powerful learning architecture that gives rise to high-quality representations and (ii) contextualization in applications, i.e., words are represented as contextualized embeddings for solving NLP tasks. Here, we adopt Schuster et al. (2019)’s method of computing uncontextualized BERT embeddings (AVG-BERT- ℓ , see §4.2.1) and show that (i) alone already has a strong positive effect on performance when compared to other uncontextualized embeddings. So BERT’s representation learning yields high performance, even when used in a completely uncontextualized setting.

We adopt the setup in Yaghoobzadeh et al. (2019) to probe uncontextualized embeddings – for each of the 34 s-classes, we train a binary classifier as shown in Figure 1. Table 2, column *words* shows the sizes of train/dev/test. The evaluation measure is micro F_1 over all decisions of the 34 binary classifiers.

3.2.2 Probing contextualized embeddings

We probe BERT with the same setup: a binary classifier is trained for each of the 34 s-classes; each BERT layer is probed individually.

For uncontextualized embeddings, a word has

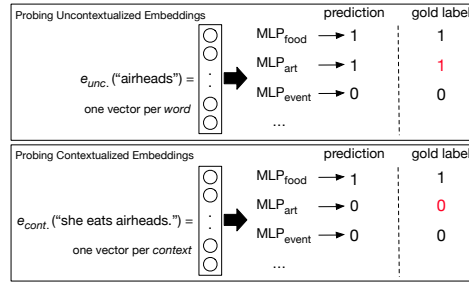


Figure 2: Setups for probing uncontextualized and contextualized embeddings. For BERT, we input a context sentence to extract the contextualized embedding of a word, e.g., “airheads”; “food” is the correct s-class label for this context.

a single vector, which is either a positive or negative example for an s-class. For contextualized embeddings, the contexts of a word will typically be mixed; for example, “food” contexts (a candy) of “@airheads@” are positive but “art” contexts (a film) of “@airheads@” are negative examples for the classifier of “food”. Table 2, column *contexts* shows the sizes of train/dev/test when probing BERT. Figure 2 compares our two probing setups.

In evaluation, we weight frequent word-s-class combinations (those having 100 contexts in our dataset) and the much larger number of less frequent word-s-class combinations equally. To this end, we aggregate the decisions for the contexts of a word-s-class combination. We stipulate that at least half of the contexts must be correctly classified. For example, “@airheads@-art” occurs 47 times, so we evaluate the “art” classifier as accurate for “@airheads@-art” if it classifies at least 24 contexts correctly. The final evaluation measure is micro F_1 over all 15,437 (for dev) and 77,706 (for test) decisions (see Table 2) of the 34 classifiers for the word-s-class combinations.

4 Experiments and Results

4.1 Data preprocessing

BERT uses wordpieces (Wu et al., 2016) to represent text and infrequent words are tokenized to several wordpieces. For example, “infrequent” is tokenized to “in”, “##ir”, “##e”, and “##quent”. Following He and Choi (2020), we average word-piece embeddings to get a single vector representation of a word.²

²Some “words” in Wiki-PSE are in reality multiword phrases. Again, we average in these cases to get a single vector representation.

We limit the maximum sequence length of the context sentence input to BERT to 128. Consistent with the probing literature, we use a simple probing classifier: a 1-layer multilayer perceptron (MLP) with 1024 hidden dimensions and ReLU.

4.2 Quantifying contextualization

4.2.1 Representation learners

Six **uncontextualized embedding spaces** are evaluated: (i) PSE. A 300-dimensional embedding space computed by running skipgram with negative sampling (Mikolov et al., 2013) on the Wiki-PSE text corpus. Yaghoobzadeh et al. (2019) show that PSE outperforms other embedding spaces. (ii) Rand. An embedding space with the same vocabulary and dimension size as PSE. Vectors are drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{300})$. Rand is used to confirm that word representations indeed encode valid meaning contents that can be identified by diagnostic MLPs rather than random weights. (iii) The 300-dimensional fastText (Bojanowski et al., 2017) embeddings. (iv) GloVe. The 300-dimensional space trained on 6 billion tokens (Pennington et al., 2014). Out-of-vocabulary (OOV) words are associated with vectors drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{300})$. (v) BERTw. The 768-dimensional wordpiece embeddings in BERT. We tokenize a word with the BERT tokenizer then average its wordpiece embeddings. (vi) AVG-BERT- ℓ .³ For an annotated word in Wiki-PSE, we average all of its contextualized embeddings from BERT layer ℓ in the Wiki-PSE text corpus. Comparing AVG-BERT- ℓ with others brings a new insight: to which extent does this “uncontextualized” variant of BERT outperform others in encoding different s-classes of a word?

Four **contextualized embedding models** are considered: (i) BERT. We use the PyTorch (Paszke et al., 2019; Wolf et al., 2019) implementation of the 12-layer BERT-base-uncased model (Wiki-PSE is uncased). (ii) P-BERT. A bag-of-words model that “contextualizes” the wordpiece embedding of an annotated word by averaging the embeddings of wordpieces of the sentence it occurs in. Comparing BERT with P-BERT reveals to which extent the self attention mechanism outperforms an average pooling practice when contextualizing words. (iii) P-fastText. Similar to P-BERT, but we use fastText word embeddings. Comparing BERT with

³BERTw and AVG-BERT- ℓ have more dimensions. But Yaghoobzadeh et al. (2019) showed that different dimensionalities have a negligible impact on relative performance when probing for s-classes using MLPs as diagnostic classifiers.

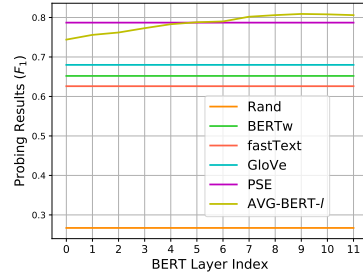


Figure 3: S-class probing results for **uncontextualized** embeddings. Results are micro F_1 on Wiki-PSE test set. Numerical values are in Table 5 in Appendix.

P-fastText indicates to which extent BERT outperforms uncontextualized embedding spaces when they also have access to contextual information. (iv) P-Rand. Similar to P-BERT, but we draw word embeddings from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{300})$. Wieting and Kiela (2019) show that a random baseline has good performance in tasks like sentence classification.

4.2.2 S-class inference results

Figure 3 shows **uncontextualized embedding** probing results. Comparing with random weights, all embedding spaces encode informative features helping s-class inference. BERTw delivers results similar to GloVe and fastText, demonstrating our earlier point (cf. the qualitative example in Table 1) that the lowest embedding layer of BERT is uncontextualized; several meanings of a word are conflated into a single vector.

PSE performs strongly, consistent with observations in Yaghoobzadeh et al. (2019). AVG-BERT-10 performs best among all spaces. Thus for a given word, averaging its contextualized embeddings from BERT yields a high quality type-level embedding vector, similar to “anchor words” in cross-lingual alignment (Schuster et al., 2019).

As expected, the top AVG-BERT layers outperform lower layers, given the deep architecture of BERT. Additionally, AVG-BERT-0 significantly outperforms BERTw, evidencing the importance of position embeddings and the self attention mechanism (Vaswani et al., 2017) when composing the wordpieces of a word.

Figure 4 shows **contextualized embedding** probing results. Comparing BERT layers, a clear trend can be identified: s-class inference performance increases monotonically with higher layers. This increase levels off in the top layers. Thus, the features from deeper layers improve word

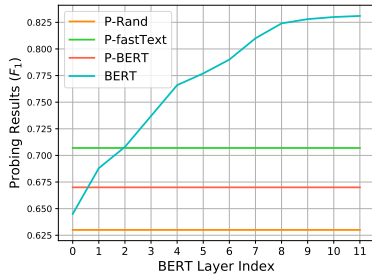


Figure 4: S-class probing results for **contextualized** embedding models. Results are micro F_1 on Wiki-PSE test set. Numerical values are in Table 6 in Appendix.

contextualization, advancing s-class inference. It also verifies previous findings: semantic tasks are mainly solved at higher layers (Liu et al., 2019; Tenney et al., 2019a). We can also observe that the strongest contextualization occurs early at lower layers – going up to layer 1 from layer 0 brings a 4% (absolute) improvement.

The very limited contextualization improvement brought by the top two layers may explain why representations from the top layers of BERT can deliver suboptimal performance on NLP tasks (Liu et al., 2019): the top layers are optimized for the pretraining objective, i.e., predicting masked words (Voita et al., 2019), not for the contextualization of words that is helpful for NLP tasks.

BERT layer 0 performs slightly worse than P-BERT, which may be due to the fact that some attention heads in lower layers of BERT attend broadly in the sentence, producing “bag-of-vector-like” representations (Clark et al., 2019), which is in fact close to the setup of P-BERT. However, starting from layer 1, BERT gradually improves and surpasses P-BERT, achieving a maximum gain of 0.16 in F_1 in layer 11. Thus, BERT knows how to better interpret the word in context, i.e., contextualize the word, when progressively going to deeper (higher) layers.

P-Rand performs strongly, but is noticeably worse than P-fastText and P-BERT. P-fastText outperforms P-BERT and BERT layers 0 and 1. We hypothesize that this may be due to the fact that fastText learns embeddings directly for words; P-BERT and BERT have to compose subwords to understand the meaning of a word, which is more challenging. Starting from layer 2, BERT outperforms P-fastText and P-BERT, illustrating the effectiveness of self attention in better integrating the information from the context into contextualized

word embeddings than the average pooling practice in bag-of-word models.

Figure 3 and Figure 4 jointly illustrate the high quality of word representations computed by BERT. The BERT-derived uncontextualized AVG-BERT- ℓ representations – modeled as Schuster et al. (2019)’s anchor words – show superior capability in inferring s-classes of a word, performing best among all uncontextualized embeddings. This suggests that BERT’s powerful learning architecture may be the main reason for BERT’s high performance, not contextualization proper, i.e., the representation of words as contextualized embeddings on the highest layer when BERT is applied to NLP tasks. This offers intriguing possibility for creating (or distilling) strongly performing uncontextualized BERT-derived models that are more compact and more efficiently deployable.

4.2.3 Qualitative analysis

§4.2.2 quantitatively shows that BERT performs strongly in contextualizing words, thanks to its deep integration of information from the entire input sentence in each contextualized embedding. But there are scenarios where BERT fails. We identify two such cases in which the contextual information does not help s-class inference.

(i) **Tokenization.** In some domains, the annotated word and/or its context words are tokenized into several wordpieces due to their low frequency in the pretraining corpora. As a result, BERT may not be able to derive the correct composed meaning. Then the MLPs cannot identify the correct s-class from the noisy input. Consider the tokenized results of “@glutamate@-biology” and one of its contexts:

“three ne ##uro ##tra ##ns ##mit ##ters that play important roles in adolescent brain development are g ##lu ##tama ##te ...”

Though “brain development” hints at a context related to “biology”, this signal could be swamped by the noise in embeddings of other – especially short – wordpieces. Schick and Schütze (2020) propose a mimicking approach (Pinter et al., 2017) to help BERT understand rare words.

(ii) **Uninformative contexts.** Some contexts do not provide sufficient information related to the s-class. For example, according to probing results on BERTw, the wordpiece embedding of “goodfellas” does not encode the meaning of s-class “art” (i.e., movies); the context “Chase also said he wanted Imperoli because he had been in Goodfellas” of

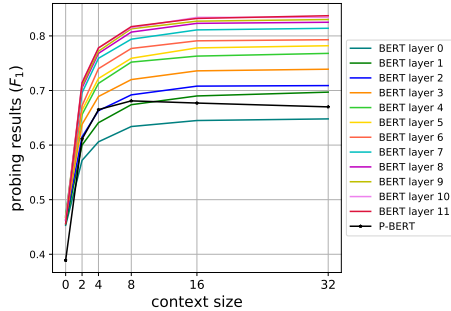


Figure 5: Probing results on the dev set with different context sizes. For BERT, performance increases with context size. Large context sizes like 16 and 32 slightly hurt performance of P-BERT.

word-s-class combination “@goodfellas@-art” is not informative enough for inferring an “art” context, yielding incorrect predictions in higher layers.

4.3 Context size

We now quantify the amount of context required by BERT for properly contextualizing words to produce accurate s-class inference results.

When probing for the s-class of word w , we define *context size* as the number of words surrounding w (left and right) in a sentence before wordpiece tokenization. For example, a context size of 5 means 5 words left, 5 words right. The context size seems to be picked heuristically in other work. Yarowsky (1992) and Gale et al. (1992) use 50 while Black (1988) uses 3–6. We experiment with a range of context sizes then compare s-class inference results. We also enclose P-BERT for comparison. Note that this experiment is different from edge probing (Tenney et al., 2019b), which takes the full sentence as input. We only make input words within the context window available to BERT and P-BERT.

4.3.1 Probing results

We report micro F_1 on Wiki-PSE dev, with context size $\in \{0, 2, 4, 8, 16, 32\}$. Context size 0 means that the input consists only of the wordpiece embeddings of the input word. Figure 5 shows results.

Comparing context sizes. Larger context sizes have higher performance for all BERT layers. Improvements are most prominent for small context sizes, e.g., 2 and 4, meaning that often local features are sufficient to contextualize words and infer s-classes, supporting Black (1988)’s design choice of 3–6. Further increasing the context size im-

proves contextualization only marginally.

A qualitative example showing informative local features is “The Azande speak Zande, which they call Pa-Zande.” In this context, the gold s-class of “Zande” is “language” (instead of “people-ethnicity”, i.e., the Zande people). The MLPs for BERTw and for context size 0 for BERT fail to identify s-class “language”. But the BERT MLP for context size 2 predicts “language” correctly since it includes the strong signal “speak”. This context is a case of selectional restrictions (Resnik, 1993; Jurafsky and Martin, 2009), in this case possible objects of “speak”.

As small context sizes already contain noticeable information contextualizing the words, we hypothesize that it may not be necessary to exploit the full context in cases where the quadratic complexity of full-sentence self attention is problematic, e.g., on edge devices. Initial results on part-of-speech tagging with the Penn Treebank (Marcus et al., 1993) in Appendix §C confirm our hypothesis. We leave more experiments to future work.

P-BERT shows a similar pattern when varying the context sizes. However, large context sizes such as 16 and 32 hurt contextualization, meaning that averaging too many embeddings results in a bag of words not specific to a particular token.

Comparing BERT layers. Higher layers of BERT yield better contextualized word embeddings. This phenomenon is more noticeable for large context sizes such as 8, 16 and 32. However for small context sizes, e.g., 0, embeddings from all layers perform similarly and badly. This means that without context information, simply passing the wordpiece embedding of a word through BERT layers does not help, suggesting that contextualization is the key ability of BERT yielding impressive performance across NLP tasks.

Again, P-BERT only outperforms layer 0 of BERT with most context sizes, suggesting that BERT layers, especially the top layers, contextualize words with abstract and informative representations, instead of naively aggregating all information within the context sentence.

4.4 Probing finetuned embeddings

We have done “classical” probing: extracting features from pretrained BERT and feeding them to diagnostic classifiers. However, pretrained BERT needs to be adapted, i.e., finetuned, for good performance on tasks (Devlin et al., 2019; Peters et al.,

	POS	SST2	MRPC	NER
Ours	.977	.928	.853	.946
Devlin et al. (2019)	n/a	.927	.867	.964

Table 3: Dev set performance of finetuning BERT (bert-base-uncased). For NER, we report micro F_1 . For other tasks, we report accuracy.

2019). Thus, it is necessary to investigate how finetuning BERT affects the contextualization of words and analyze how the pretrained knowledge and probed features change.

4.4.1 Finetuning tasks

We finetune BERT on four tasks: part-of-speech (POS) tagging on the Penn Treebank (Marcus et al., 1993), named-entity recognition (NER) on the CoNLL-2003 Shared Task (Tjong Kim Sang and De Meulder, 2003), binary sentiment classification on the Stanford Sentiment Treebank (SST2) (Socher et al., 2013) and paraphrase detection on the Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005). For SST2 and MRPC, we use the GLUE train and dev sets (Wang et al., 2018). For POS, sections 0-18 of WSJ are train and sections 19-21 are dev (Collins, 2002). For NER, we use the official data splits.

Following Devlin et al. (2019), we put a linear layer on top of the pretrained BERT, then finetune all parameters. We use Adam (Kingma and Ba, 2014) with learning rate $5e-5$ for 5 epochs. We save the model from the step that performs best on dev (of MRPC/SST2/POS/NER), extract representations from Wiki-PSE using this model and then report results on Wiki-PSE dev.

Table 3 reports the finetuning results. Our finetuned models perform comparably to Devlin et al. (2019) on SST2 and MRPC. Our NER result is slightly worse, this may be due to the fact that Devlin et al. (2019) use “maximal document context” while we use sentence-level context of 128 max sequence length. More finetuning details are available in Appendix §B.

4.4.2 Probing results

We now quantify the contextualization of word representations from finetuned BERT models. Two setups are considered: (a) directly apply the MLPs in §4.2 (trained with pretrained embeddings) to finetuned BERT embeddings; (b) train and evaluate a new set of MLPs on the finetuned BERT embeddings.

Comparing (a) with probing results on pretrained BERT (§4.2) gives us an intuition about how many changes occurred to the knowledge captured during pretraining. Comparing (b) with §4.2 reveals whether or not the pretrained knowledge about contextualization is still preserved in finetuned models.

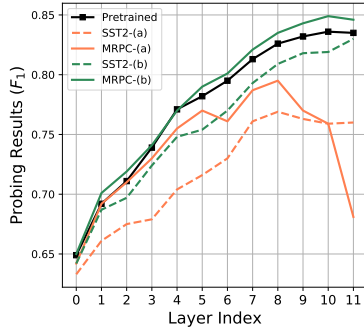
Figure 6 shows s-class probing results of finetuned BERT with setup (a) and (b). For example in (ii), layer 11 s-class inference performance of the POS-finetuned BERT decreases by 0.763 ($0.835 \rightarrow 0.072$, from “Pretrained” to “POS-(a)”) when using the MLPs from §4.2.

Comparing setup (a) and “Pretrained”, we see that finetuning brings significant changes to the word representations. Finetuning on POS and NER introduces more obvious probing accuracy drops than finetuning on SST2 and MRPC. This may be due to the fact that the training objective of SST2 and MRPC takes as input only the [CLS] token while all words in a sentence are involved in the training objective of POS and NER.

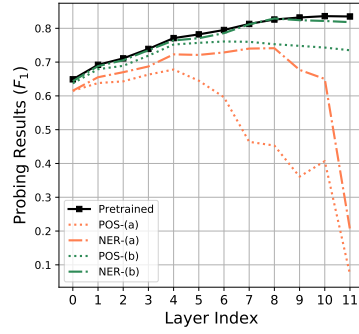
Comparing setup (b) and “Pretrained”. Finetuning BERT on MRPC introduces small but consistent improvements on s-class inference. For SST2 and NER, very small s-class inference accuracy drops are observed. Finetuning on POS brings more noticeable changes. Solving POS requires more syntactic information than the other tasks, inducing BERT to “propagate” the syntactic information that is represented in lower layers to the upper layers; due to their limited capacity, the fixed-size vectors from the upper layers may lose some semantic information, yielding a more noticeable performance drop on s-class inference.

Comparing (a) and (b), we see that the knowledge about contextualizing words captured during pretraining is still well preserved after finetuning. For example, the MLPs trained with layer 11 embeddings computed by the POS-finetuned BERT still achieve a reasonably good score of 0.735 (a 0.100 drop compared with “Pretrained” – compare black and green dotted lines in Figure 6 (ii)). Thus, the semantic information needed for inferring s-classes is still present to a large extent.

Finetuning may introduce large changes (setup (a)) to the representations – similar to the projection utilized to uncover divergent information in uncontextualized word embeddings (Artetxe et al., 2018) – but relatively little information about contextualization is lost as the good performance of the newly trained MLPs shows (setup (b)). Similarly,

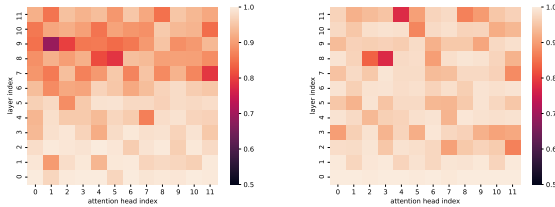


(i) MRPC and SST2



(ii) POS and NER

Figure 6: Comparing s-class inference results of pretrained BERT and BERT finetuned on MRPC, SST2, POS, and NER. “Pretrained”: probing results on weight-frozen pretrained BERT in §4.2. For (a), we directly apply the MLPs in §4.2 (trained with pretrained embeddings) to finetuned BERT embeddings; for (b), we train and evaluate a new set of MLPs on the finetuned BERT embeddings.



(i) pretrained vs. POS

(ii) pretrained vs. MRPC

Figure 7: Cosine similarity of flattened self attention weights. X-axis: index of the 12 self attention heads; y-axis: layer index. Darker colors: smaller similarities, i.e., larger changes brought by finetuning.

Merchant et al. (2020) show that finetuned BERT still well preserves the probed “linguistic features” in pretrained BERT.

Comparing BERT layers. Contextualized embeddings from BERT’s top layers are strongly affected by finetuning, especially for setup (a). In contrast, lower layers are more invariant and show s-class inference results similar to the pretrained model. Hao et al. (2019), Lee et al. (2019), Koval-eva et al. (2019) make similar observations: lower layer representations are more transferable across different tasks and top layer representations are more task-specific after finetuning.

Figure 7 shows the cosine similarity of the flattened self attention weights computed by pretrained, POS-, and MRPC-finetuned BERT using the dev set examples. We see that top layers are more sensitive to finetuning (darker color) while lower layers are barely changed (lighter color). Top layers have more changes for POS than for MRPC, in line with probing results in Figure 6.

5 Related Work

Interpreting deep networks. Pretrained language models (McCann et al., 2017; Peters et al., 2018a; Radford et al., 2019; Devlin et al., 2019) advance NLP by contextualized representations of words. A key goal of current research is to understand how these models work and what they represent on different layers.

Probing is a recent strand of work that investigates – via diagnostic classifiers – desired syntactic and semantic features encoded in pretrained language model representations. Shi et al. (2016) show that string-based RNNs encode syntactic information. Belinkov et al. (2017) investigate word representations at different layers in NMT. Linzen et al. (2016) assess the syntactic ability of LSTM (Hochreiter and Schmidhuber, 1997) encoders and Goldberg (2019) of BERT. Tenney et al. (2019a) find that information on POS tagging, parsing, NER, semantic roles, and coreference is represented on increasingly higher layers of BERT. Yaghoobzadeh et al. (2019) assess the disambiguation properties of type-level word representations. Liu et al. (2019) and Lin et al. (2019) investigate the linguistic knowledge encoded in BERT. Adi et al. (2016), Conneau et al. (2018), and Wieting and Kiela (2019) study sentence embedding properties via probing. Peters et al. (2018b) probe how the network architecture affects the learned vectors.

In all of these studies, probing serves to analyze representations and reveal their properties. We employ probing to investigate the contextualization of words in pretrained language models quantitatively. In addition, we exploit how finetuning affects word contextualization.

Ethayarajh (2019) quantitatively investigates contextualized embeddings, using unsupervised cosine-similarity-based evaluation. Inferring s-classes, we address a complementary set of questions because we can quantify contextualization with a uniform set of semantic classes. Brunner et al. (2020) employ token identifiability to compute the deviation of a contextualized embedding from the uncontextualized embedding. Voita et al. (2019) address this from the mutual information perspective, e.g., low mutual information between an uncontextualized embedding and its contextualized embedding can be viewed as a reflection of more contextualization. Similar observations are made: higher layer embeddings are more contextualized while lower layer embeddings are less contextualized. In contrast, we draw the observations from the perspective of s-class inference. The higher layer embeddings perform better when evaluating the semantic classes – they are better contextualized and have higher fitness to the context than the lower layer embeddings.

Two-stage NLP paradigm. Recent work (Dai and Le, 2015; Howard and Ruder, 2018; Devlin et al., 2019) introduces a “two-stage paradigm” in NLP: pretrain a language encoder on a large amount of unlabeled data via self-supervised learning, then finetune the encoder on task-specific benchmarks like GLUE (Wang et al., 2018, 2019). This transfer-learning pipeline yields good and robust results compared to models trained from scratch (Hao et al., 2019).

In this work, we shed light on how BERT’s pretrained knowledge about contextualization changes during finetuning by comparing s-class inference ability of pretrained and finetuned models. Merchant et al. (2020) analyze BERT models finetuned on different downstream tasks with the edge probing suite (Tenney et al., 2019b) and make similar observations as us. They focus on “linguistic features” while we focus on the contextualization of words.

6 Conclusion

We presented a quantitative study of the contextualization of words in BERT by investigating BERT’s semantic class inference capabilities. We focused on two key factors for successful contextualization by BERT: layer index and context size. By comparing pretrained and finetuned models, we showed that word-level tasks like part-of-speech tagging

bring more noticeable changes than sentence-level tasks like paraphrase classification; and top layers of BERT are more sensitive to the finetuning objective than lower layers. We also found that BERT’s pretrained knowledge about contextualizing words is still well retained after finetuning.

We showed that exploiting the full context may be unnecessary in applications where the quadratic complexity of full-sentence attention is problematic. Future work may evaluate this phenomenon on more datasets and downstream tasks.

Acknowledgments

We thank the anonymous reviewers for the insightful comments and suggestions. This work was funded by the European Research Council (ERC #740516) and a Zentrum Digitalisierung.Bayern fellowship award.

References

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.
- Mikel Artetxe, Gorika Labaka, Iñigo Lopez-Gazpio, and Eneko Agirre. 2018. [Uncovering divergent linguistic information in word embeddings with lessons for intrinsic and extrinsic evaluation](#). In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 282–291, Brussels, Belgium. Association for Computational Linguistics.
- Yonatan Belinkov and James Glass. 2019. [Analysis methods in neural language processing: A survey](#). *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017. [Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Rubén Izquierdo Beviá, Armando Suárez Cueto, and Germán Rigau Claramunt. 2007. Exploring the automatic selection of basic level concepts.
- Ezra Black. 1988. An experiment in computational discrimination of english word senses. *IBM Journal of research and development*, 32(2):185–194.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with](#)

- subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. [On identifiability in transformers](#). In *International Conference on Learning Representations*.
- Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 168–175. Association for Computational Linguistics.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Michael Collins. 2002. [Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms](#). In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8. Association for Computational Linguistics.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [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. Association for Computational Linguistics.
- Andrew M. Dai and Quoc V. Le. 2015. [Semi-supervised sequence learning](#).
- 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.
- William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Philip Edmonds and Scott Cotton. 2001. [SENSEVAL-2: Overview](#). In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- William A Gale, Kenneth W Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics.
- Yoav Goldberg. 2019. Assessing bert’s syntactic abilities. *arXiv preprint arXiv:1901.05287*.
- Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. [Visualizing and understanding the effectiveness of BERT](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4134–4143, Hong Kong, China. Association for Computational Linguistics.
- Han He and Jinho D. Choi. 2020. [Establishing Strong Baselines for the New Decade: Sequence Tagging, Syntactic and Semantic Parsing with BERT](#). In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference, FLAIRS’20*. Best Paper Candidate.
- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Rubén Izquierdo, Armando Suárez, and German Rigau. 2009. An empirical study on class-based word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 389–397. Association for Computational Linguistics.

- Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Upali Sathyajith Kohomban and Wee Sun Lee. 2005. [Learning semantic classes for word sense disambiguation](#). In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 34–41, Ann Arbor, Michigan. Association for Computational Linguistics.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. [Revealing the dark secrets of BERT](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4356–4365, Hong Kong, China. Association for Computational Linguistics.
- Jaesun Lee, Raphael Tang, and Jimmy Lin. 2019. [What would elsa do? freezing layers during transformer fine-tuning](#).
- Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside bert’s linguistic knowledge. *arXiv preprint arXiv:1906.01698*.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. [Assessing the ability of LSTMs to learn syntax-sensitive dependencies](#). *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. [Linguistic knowledge and transferability of contextual 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bernardo Magnini and Gabriela Cavaglià. 2000. [Integrating subject field codes into WordNet](#). In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.
- Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. [What happens to bert embeddings during fine-tuning?](#)
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global Vectors for Word Representation](#). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. [Dissecting contextual word embeddings: Architecture and representation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.
- Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. [To tune or not to tune? adapting pre-trained representations to diverse tasks](#). In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepLanLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.
- Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. [Mimicking word embeddings using subword RNNs](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112, Copenhagen, Denmark. Association for Computational Linguistics.

- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. [Word sense disambiguation: A unified evaluation framework and empirical comparison](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.
- Philip Resnik. 1993. [Semantic classes and syntactic ambiguity](#). In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.
- Timo Schick and Hinrich Schütze. 2020. [Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8766–8774. AAAI Press.
- Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. [Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing](#). 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.
- Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. [Green ai](#).
- Xing Shi, Inkit Padhi, and Kevin Knight. 2016. [Does string-based neural MT learn source syntax?](#) In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.
- Benjamin Snyder and Martha Palmer. 2004. [The English all-words task](#). In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Elena Voita, Rico Sennrich, and Ivan Titov. 2019. [The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4387–4397, Hong Kong, China. Association for Computational Linguistics.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Super-glue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. *arXiv preprint arXiv:1901.10444*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Transformers: State-of-the-art natural language processing](#).

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yadollah Yaghoobzadeh, Katharina Kann, T. J. Hazen, Eneko Agirre, and Hinrich Schütze. 2019. [Probing for semantic classes: Diagnosing the meaning content of word embeddings](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5740–5753, Florence, Italy. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ArXiv*, abs/1906.08237.

David Yarowsky. 1992. [Word-sense disambiguation using statistical models of Roget’s categories trained on large corpora](#). In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

A Reproducibility Checklist

A.1 Computing infrastructure

All experiments are conducted on GeForce GTX 1080 Ti and GeForce GTX 1080.

A.2 Number of parameters

We use a set of 34 binary MLPs to conduct our probing task. Each MLP has input dimension 768, hidden dimension 1024 and output dimension 2. As a result, the total number of parameters is 26,843,204. For finetuning, we use the BERT-base-uncased model containing about 110 million parameters (<https://github.com/google-research/bert>).

A.3 Validation performance

Following Table 5 and Table 6 report the validation performance of probing uncontextualized and contextualized embeddings.

A.4 Evaluation metric

Our evaluation is the micro F_1 over all decisions of the 34 probing classifiers. More details are available in §3.2 of the main paper.

A.5 Hyperparameter search

For probing tasks, we do not conduct hyperparameter search since our goal is to analyze the contextualization. The probing classifiers are trained with learning rate $1e-3$ and 400 epochs. For finetuning BERT, we do not search hyperparameters but directly adopt the setup in Devlin et al. (2019) as shown in Table 4.

A.6 Datasets

List of the 34 semantic classes (s-classes), number of word-s-class combinations and contexts per s-class in the sampled Wiki-PSE (Yaghoobzadeh et al., 2019) are listed in Table 8. Some annotated contexts in Wiki-PSE are also displayed in Table 9. The Wiki-PSE developed by Yaghoobzadeh et al. (2019) is publicly available at <https://github.com/yyaghoobzadeh/WIKI-PSE>.

When finetuning BERT, we use the GLUE (Wang et al., 2018) splits of MRPC and SST2 from <https://gluebenchmark.com/>. Our POS dataset is from the linguistic data consortium (LDC). For NER (Tjong Kim Sang and De Meulder, 2003), we use the official shared task dataset: <https://www.clips.uantwerpen.be/conll2003/ner/>.

	POS	SST2	MRPC	NER
batch size	150	200	350	32
learning rate	5e-5	5e-5	5e-5	5e-5
max epoch	5	5	5	5
max sequence length	128	128	128	128

Table 4: Hyperparameters for finetuning.

B Finetuning Details

Hyperparameters in Table 4 are used when we finetune BERT on POS, NER, SST2, and MRPC. For SST2 and MRPC, we use the embedding of [CLS] as the representation of the sentence (pair). For POS and NER, we use the embedding of the last wordpiece of the word as Liu et al. (2019).

A plain Adam (Kingma and Ba, 2014) optimizer is used and we did not use strategies like learning rate warmup and layer-wise learning rate (Howard and Ruder, 2018) during finetuning to avoid potential side effects to ensure a clear comparison of different BERT layers.

C Context Sizes in POS

We investigate how the findings from §4.3 in the main paper transfer to downstream tasks. To this end we perform standard finetuning of BERT for different tasks, but we prune the attention matrix to a context size of length k . That is we apply a mask on the attention matrix such that each word can only attend to k left and k right words. This has great benefits as it reduces the memory and computation requirements from $\mathcal{O}(n^2)$ to $\mathcal{O}(nk)$ where n is the sequence length. We only consider part-of-speech tagging as for sentence pair classification tasks such as SST2 and MRPC this is not a sensible approach.

Table 7 confirms that small context windows are sufficient to achieve full performance for POS-tagging. This indicates that the finding from the main paper (i.e., local context is sufficient for BERT to achieve a high degree of contextualization) is to some degree applicable to a downstream tasks, as well. Note that the median sentence length in the Penn Treebank dataset is 25 words (the number of wordpieces even higher). Thus masking the context to the next 4 or 8 words does indeed reduce the available context words. In future work we plan to investigate this effect not only during finetuning but also during pretraining.

	Standard Embeddings					AVG-BERT- ℓ											
	Rand	BERTw	fastText	GloVe	PSE	0	1	2	3	4	5	6	7	8	9	10	11
dev	.269	.653	.625	.681	.790	.746	.759	.764	.775	.786	.791	.794	.805	.811	.812	.813	.809
test	.267	.652	.626	.680	.787	.744	.756	.762	.773	.783	.788	.790	.802	.806	.809	.808	.806

Table 5: S-class probing results for **uncontextualized** embeddings. Numbers are micro F_1 on Wiki-PSE. Our result (0.787 on PSE-test) is consistent with [Yaghoobzadeh et al. \(2019\)](#). Additionally, for the top 6 layers {6, 7, 8, 9, 10, 11} of AVG-BERT, we repeat the experiments 5 times with random seed in {1, 2, 3, 4, 5}. Mean and standard deviation on test per layer are: $\{.791\pm.001, .801\pm.001, .807\pm.001, .808\pm.001, .808\pm.001, .805\pm.001\}$.

	Bag-of-word context			BERT Layer											
	P-Rand	P-fastText	P-BERT	0	1	2	3	4	5	6	7	8	9	10	11
dev	.637	.707	.672	.649	.692	.711	.739	.771	.782	.795	.813	.826	.832	.836	.835
test	.630	.707	.670	.645	.688	.708	.737	.766	.777	.790	.810	.824	.828	.830	.831

Table 6: S-class probing results for **contextualized** embedding models. Numbers are micro F_1 on Wiki-PSE.

Context size	POS
0	.886
2	.973
4	.975
8	.976
16	.977
32	.977
All	.977

Table 7: POS accuracy on dev for different context sizes.

semantic classes	train		dev		test	
	comb's	contexts	comb's	contexts	comb's	contexts
location	13,474	618,932	3,408	152,470	16,859	776,848
person	15,423	617,270	3,744	151,005	19,212	765,655
organization	9,556	332,063	2,496	88,682	11,915	411,716
art	7,428	201,529	1,854	52,295	9,192	247,481
event	3,515	87,735	900	21,566	4,404	108,963
broadcast-program	2,287	67,261	530	15,062	2,828	84,343
title	1,429	43,041	311	9,646	1,792	56,333
product	3,121	49,076	766	13,438	3,808	61,585
living-thing	1,302	35,595	320	9,035	1,702	46,040
people-ethnicity	754	27,573	181	6,699	951	35,332
language	671	14,842	145	3,147	824	20,308
broadcast-network	325	12,392	80	3,036	362	13,006
time	157	7,765	39	1,997	192	9,984
religion-religion	192	6,461	45	1,760	265	9,719
award	251	7,589	61	1,776	301	8,877
internet-website	88	2,466	21	645	141	3,851
god	246	7,306	52	1,998	340	11,810
education-educational-degree	97	3,282	24	901	142	4,833
food	381	7,805	112	2,003	480	9,514
computer-programming-language	105	2,739	29	402	123	2,677
metropolitan-transit-transit-line	285	5,603	76	1,259	382	6,948
transit	135	3,781	26	628	186	4,305
finance-currency	127	3,107	30	548	166	3,388
disease	163	2,619	33	381	260	4,385
chemistry	170	3,350	43	1,254	195	3,858
body-part	135	1,901	31	415	156	2,591
finance-stock-exchange	27	617	3	5	51	795
law	23	474	6	54	27	535
medicine-medical-treatment	77	886	7	124	106	1,803
medicine-drug	50	1,023	7	54	72	1,157
broadcast-tv-channel	45	564	14	210	74	1,264
medicine-symptom	55	752	15	97	72	1,172
biology	49	485	15	118	63	911
visual-art-color	41	1,011	13	228	63	906
total	62,184	2,178,895	15,437	542,938	77,706	2,722,893

Table 8: Number of word-s-class combinations and contexts for each of the 34 semantic classes in Wiki-PSE.

word	word-s-class combination	contexts
roberta	@roberta@-art	this recording is also available on cd paired with @roberta@-art to star as huckleberry haines in the jerome kern / dorothy fields musical @roberta@-art .
	@roberta@-location	there are also learning centers in eatonton , forsyth , gray , jeffersonville , and @roberta@-location the concurrency curves to a nearly due north routing and enters @roberta@-location .
	@roberta@-person	ken williams : along with wife @roberta@-person , founded on-line systems after working at ibm mystery house is an adventure game released in 7 by @roberta@-person and ken williams for the apple ii .
larch	@larch@-comp-prog-lang	wing has been a leading member of the formal methods community , especially in the area of @larch@-comp-prog-lang . a major contribution was his involvement with the @larch@-comp-prog-lang approach to formal specification with ...
	@larch@-living-thing	the more recent plantings include @larch@-living-thing and pine . these consist mainly of oak , alder , @larch@-living-thing and corsican pine .

Table 9: Example contexts of the annotated word “roberta” and “larch”.