SenseBERT: Driving Some Sense into BERT

Yoav Levine Barak Lenz Or Dagan Dan Padnos Or Sharir
Shai Shalev-Shwartz Amnon Shashua Yoav Shoham

AI21 Labs, Tel Aviv, Israel

August 14th, 2019

Abstract

Self-supervision techniques have allowed neural language models to advance the frontier in Natural Language Understanding. However, existing self-supervision techniques operate at the word *form* level, which serves as a surrogate for the underlying semantic content. This paper proposes a method to employ self-supervision directly at the word *sense* level. Our model, named SenseBERT, is pre-trained to predict not only the masked words but also their WordNet supersenses. We accordingly attain a lexical-semantic level language model, without the use of human annotation. SenseBERT achieves significantly improved lexical understanding, as we demonstrate by experimenting on SemEval, and by attaining a state of the art result on the Word in Context (WiC) task. Our approach is extendable to other linguistic signals, which can be similarly integrated into the pre-training process, leading to increasingly semantically informed language models.

1 Introduction

Neural language models have recently undergone a qualitative leap forward, pushing the state of the art on various NLP tasks. Together with advances in network architecture [Vaswani et al., 2017], the use of *self-supervision* has proven to be key in these successes, as it allows the network to learn from massive amounts of unannotated text at the pre-training stage.

The self-supervision strategy employed in BERT [Devlin et al., 2018] involves masking some of the words in an input sentence, and then training the model to predict them given their context. Other proposed approaches for self-supervised objectives, including unidirectional [Radford et al., 2019], permutational [Yang et al., 2019], or word insertion-based [Chan et al., 2019], operate similarly, over words. However, since a given word form can possess multiple meanings (*e.g.*, the word 'bass' can refer to a fish, a guitar, a type of singer, and so on), the word itself is merely a surrogate of its actual meaning in a given context, referred to as its *sense*. In fact, from a lexical semantic perspective, the word-form level can be viewed as a surface level, oftentimes introducing challenging ambiguity [Navigli, 2009].

In this paper, we bring forth a novel methodology for applying self-supervision directly on the level of a word's meaning. By infusing explicit word-sense information into BERT's self-supervision signal, we expose the model to lexical semantics when pre-training on a large unannoted corpus. We name the resultant sense-informed model *SenseBERT*.

Specifically, we add a masked-word sense prediction task as an auxiliary task in BERTs pre-training. We thereby effectively train a *semantic-level language model* that predicts the missing words meaning jointly with the standard word-form level language model. In order to retain the ability to self-train on unannotated text, we make use of WordNet [Miller, 1998], an expert-constructed ontology that provides an inventory of word senses. The integration of this external linguistic knowledge base inherently improves the network's inductive bias towards lexical semantics.

We focus on a coarse-grained variant of a word's sense, referred to as its WordNet *supersense*, in order to mitigate identified brittleness of fine-grained word-sense systems, caused by arbitrary sense granularity, blurriness, and general subjectiveness [Kilgarriff, 1997, Schneider, 2014]. WordNet lexicographers organize all word senses into 45 supersense categories, 26 of which are for nouns, 15 for verbs, 3 for adjectives and 1 for adverbs (see full supersense table in the appendix). Disambiguating a word's supersense has been widely studied as a fundamental lexical categorization task [Ciaramita and Johnson, 2003, Basile, 2012, Schneider and A Smith, 2015].

We employ the masked word's allowed supersenses list from WordNet as a set of possible labels for the sense prediction task. The labeling of words with a single supersense (*e.g.*, 'sword' has only the supersense noun.artifact) is straightforward: We train the network to predict this supersense given the masked word's context. As for words with multiple supersenses (*e.g.*, 'bass' can be: noun.food, noun.animal, noun.artifact, noun.person, *etc.*), we train the model to predict any of these senses, leading to a simple yet effective soft-labeling scheme.

The introduction of contextualized word embeddings [Peters et al., 2018], for which a given word's embedding is context dependent rather than precomputed, has brought forth a promising prospect for sense-aware embeddings. Intuitively, a word's meaning and its context are highly correlated, thus adding the ability to change with context should make the embeddings carry sense information more naturally. Indeed, Coenen et al. [2019] have demonstrated that BERT captures word-sense information to some extent.

Nevertheless, we identify a clear gap in this abilty. We show that a BERT trained with the current word-level self-supervision, burdened with the implicit task of disambiguating word meanings, often fails to grasp lexical semantics, exhibiting high supersense misclassification rates. We further demonstrate that the self-supervised word-sense signal inserted at its pre-training allows SenseBERT to significantly bridge this gap.

Specifically, we show that SenseBERT$_{\text{BASE}}$ outscores both BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$ by a large margin on a supersense variant of the SemEval-based Word Sense Disambiguation (WSD) task standardized in Raganato et al. [2017]. Notably, SenseBERT receives competitive results on this task without fune-tuning, *i.e.*, when training a linear classifier over the pretrained embeddings, which serves as a testament for its self-acquisition of semantic content.

Furthermore, we show that SenseBERT$_{\text{BASE}}$ surpasses BERT$_{\text{LARGE}}$ in the Word in Context (WiC) task [Pilehvar and Camacho-Collados, 2018] from the SuperGLUE benchmark [Wang et al., 2019], which heavily depends on word-supersense awareness. A single SenseBERT$_{\text{LARGE}}$ model achieves state of the art performance on WiC with a score of 72.14, improving the score of BERT$_{\text{LARGE}}$ by 2.5 points.

The remainder of this paper is organized as follows. In section 2 we describe SenseBERT's modified pre-training procedure, and in section 3 we provide visualizations of the resultant semantic-level language model. In section 4 we present an empirical comparison in which SenseBERT demonstrates substantial improvements over BERT in grasping lexical semantics. Finally, in section 5 we conclude.

2 Incorporating Word-Supersense Information at Pre-training

The input to BERT is a sequence of words $\{x^{(j)} \in \{0,1\}^{D_W}\}_{j=1}^{N}$ where 15% of the words are replaced by a [MASK] token. Here N is the input sentence length, D_W is the word vocabulary size, and $x^{(j)}$ is a 1-hot vector corresponding to the j^{th} input word. For every masked word, the output is a word-score vector $y^{\text{words}} \in \mathbb{R}^{D_W}$ containing the per-word score. BERT's architecture can be decomposed to **(1)** an internal Transformer encoder architecture [Vaswani et al., 2017] wrapped by **(2)** an external mapping to the word vocabulary space, denoted by W.[1]

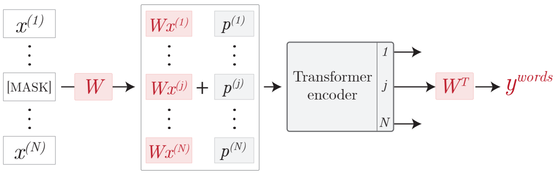
The Transformer encoder operates over a sequence of word embeddings $v_{\text{input}}^{(j)} \in \mathbb{R}^d$, where d is the Transformer encoder's hidden dimension. These are passed through multiple attention-based Transformer layers, producing a new sequence of contextualized embeddings at each layer. The Transformer encoder output is the final sequence of contextualized word embeddings $v_{\text{output}}^{(j)} \in \mathbb{R}^d$.

The external mapping $W \in \mathbb{R}^{d \times D_W}$ is effectively a translation between the external word vocabulary dimension and the internal Transformer dimension. Original words in the input sentence are translated into the Transformer block by applying this mapping (and adding positional encoding vectors $p^{(j)} \in \mathbb{R}^d$):

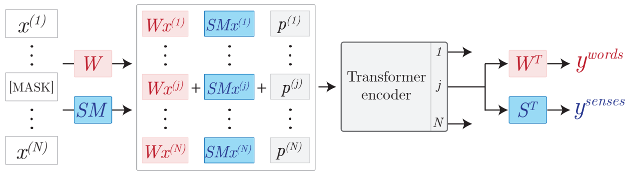
$$v_{\text{input}}^{(j)} = Wx^{(j)} + p^{(j)} \tag{1}$$

The word-score vector for a masked word at position j is extracted from the Transformer en-

[1] For clarity of presentation, we omit a description of subword tokenization which takes place for out-of-vocabulary-words, and of the Next Sentence Prediction task which we employ as in Devlin et al. [2018].



(a) BERT



(b) SenseBERT

Figure 1: SenseBERT includes a masked-word supersense prediction task, pre-trained jointly with BERT's original masked-word prediction task [Devlin et al., 2018] (see section 2.1). As in the original BERT, the mapping from the Transformer dimension to the external dimension is the same both at input and at output (W for words and S for supersenses), where M denotes a hard-coded mapping between word-forms and their allowed WordNet supersenses (see section 2.2). The vectors $p^{(j)}$ denote positional embeddings. For clarity, we omit a reference to a sentence-level Next Sentence Prediction task trained jointly with the above.

coder output by applying the transpose: $y^{\text{words}} = W^\top v_{\text{output}}^{(j)}$ [see illustration in fig. 1(a)].

In the following subsections, we frame our contribution to the above process as an addition of a parallel external mapping to the words supersenses space, denoted $S \in \mathbb{R}^{d \times D_S}$ [see illustration in fig. 1(b)], where D_S is the size of supersenses vocabulary. Specifically, in section 2.1 we describe the loss function used for learning S in parallel to W, effectively implementing a word-form and word-sense multi-task learning in the unsupervised pre-training stage. Then, in section 2.2 we describe our methodology for adding supersense information in S to the initial Transformer embedding, in parallel to word-level information added by W. Finally, in section 2.3 we describe our modification of BERT's masking strategy, prioritizing single-supersensed words which carry a stronger semantic signal.

2.1 Self-Supervised Supersense Prediction Task

Given a masked word in position j, BERT's original masked-word prediction pre-training task is to have the word-score vector $y^{\text{words}} = W^\top v_{\text{output}}^{(j)}$ get as close as possible to a 1-hot vector corresponding to the masked word. This is done by minimizing the cross-entropy loss between the softmax of the word-score vector and a 1-hot vector corresponding to the masked word:

$$\mathcal{L}_{\text{LM}} = -\log p(w|\text{context}), \tag{2}$$

where w is the masked word, the context is composed of the rest of the input sequence, and the probability is computed by:

$$p(w|context) = \frac{\exp y_w^{\text{words}}}{\sum_{w'} \exp y_{w'}^{\text{words}}}, \tag{3}$$

where y_w^{words} denotes the w^{th} entry of the word-score vector.

We follow the above procedure for training the word-level language model of SenseBERT. Jointly, for every masked word, we train the model to predict its supersense, *i.e.*, the objective is to have the sense-score vector $y^{\text{senses}} \in \mathbb{R}^{D_S} := S^\top v_{\text{output}}^{(j)}$ get as close as possible to a 1-hot vector corresponding to the word's correct supersense.

Specifically, we employ a combination of two loss terms for the supersense-level language model. The following *allowed-senses term* maximizes the probability that the predicted sense is in the set of allowed supersenses of the masked word w:

$$\begin{aligned} \mathcal{L}_{\text{SLM}}^{\text{allowed}} &= -\log p\left(s \in A(w)|\text{context}\right) \\ &= -\log \sum_{s \in A(w)} p(s|\text{context}), \end{aligned} \tag{4}$$

where $A(w)$ is the group of allowed supersenses for the masked word, and the probability for a supersense s is given by:

$$p(s|\text{context}) = \frac{\exp(y_s^{\text{senses}})}{\sum_{s'} \exp(y_{s'}^{\text{senses}})}. \tag{5}$$

Though this method of equally treating all allowed supersenses of the masked word introduces noise, the correct supersense in a given context is expected to receive a reinforced signal while incorrect supersenses are expected to statistically average out. Consider the following examples of a food context.

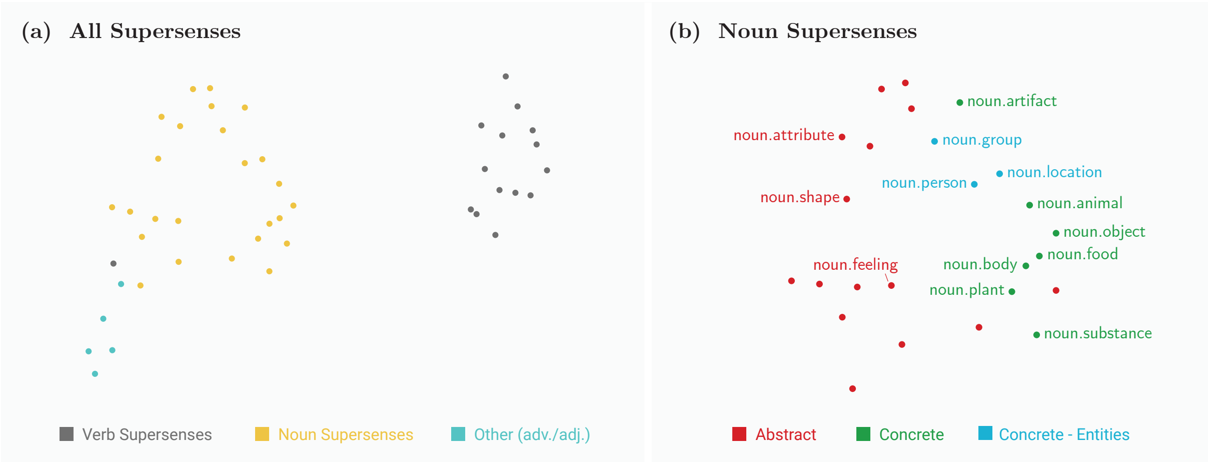


Figure 2: UMAP visualization of supersense vectors (rows of the classifier S) learned by SenseBERT at pre-training. **(a)** Clustering by the supersense's part-of speech. **(b)** Within noun supersenses, semantically similar supersenses are clustered together (see more details in appendix A).

1. "This **bass** is delicious
   (supersenses: noun.food, noun.artifact, *etc.*)
2. "This **chocolate** is delicious"
   (supersenses: noun.food, noun.attribute, *etc.*)
3. "This **pickle** is delicious"
   (supersenses: noun.food, noun.state, *etc.*)

Despite the fact that every example for this context pulls the model in different directions according to the supersenses of the specific masked word, the fact that the ground truth supersense is always there implies that incorrect supersenses cancel out, while correct senses remain significant. The assumption is that these statistics will accumulate given sufficient data.

While $\mathcal{L}_{\text{SLM}}^{\text{allowed}}$ pushes the network in the right direction, minimizing this loss could result in the network becoming overconfident in predicting a strict subset of the senses, i.e., a collapse of the prediction distribution. This is especially acute in the early stages of the training procedure, when the network could converge to the noisy signal of the weak-labeling scheme.

To mitigate this issue, the following *regularization term* is added to the loss, which encourages a uniform prediction distribution over the allowed supersenses:

$$\mathcal{L}_{\text{SLM}}^{\text{reg}} = - \sum_{s \in A(w)} \frac{1}{|A(w)|} \log p(s|\text{context}), \quad (6)$$

i.e., a cross-entropy loss with a uniform distribution over the allowed supersenses.

Finally, for training the semantic level language model, we make use of a combined loss of the form:

$$\mathcal{L}_{\text{SLM}} = \mathcal{L}_{\text{SLM}}^{\text{allowed}} + \mathcal{L}_{\text{SLM}}^{\text{reg}}. \quad (7)$$

The visualizations provided in section 3, along with the empirical results in section 4, show that a SenseBERT trained with this loss attains a good grasp of lexical semantics given contexts.

2.2 Supersense Aware Input Embeddings

Though in principle two different matrices could have been used for converting in and out of the Tranformer encoder, the BERT architecture employs the same mapping W. This approach was shown to yield models with reduced perplexity by Press and Wolf [2016]. Intuitively, constructing the Transformer encoder's input embeddings from the same mapping with which the scores are computed improves their quality as it makes the input more sensitive to the training signal.

We follow this approach, and insert our newly proposed semantic-level language model matrix S in the input in addition to W [as depicted in fig. 1(b)], such that the input vector to the Transformer encoder obeys:

$$v_{\text{input}}^{(j)} = (W + SM)x^{(j)} + p^{(j)}, \quad (8)$$

where $p^{(j)}$ are the regular positional embeddings as used in BERT, and $M \in \mathbb{R}^{D_S \times D_W}$ is a static 0/1 matrix converting between words and their allowed WordNet supersenses.

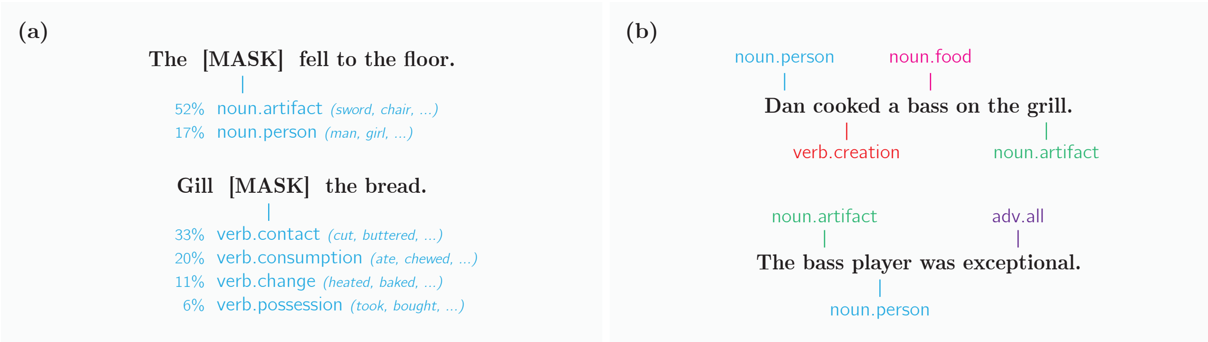


Figure 3: **(a)** A demonstration of supersense probabilities assigned to a masked position within context, as given by SenseBERT's word-supersense level semantic language model. Example words corresponding to each supersense are presented in parentheses. **(b)** Examples of SenseBERT's self-prediction on raw text, when the unmasked input sentence is given to the model. This beyond word-form abstraction ability facilitates a more natural elicitation of semantic content at pre-training.

The above strategy for constructing $v^{(j)}_{\text{input}}$ allows for the semantic level vectors in S to come into play and shape the input embeddings even for words which are rarely observed in the training corpus. For such a word, the corresponding row in W is potentially less informative, since due to the low word frequency the model did not have sufficient chance to adequately learn it. However, since the model learns a representation of its supersense, the corresponding row in S is informative of the semantic category of the word. Therefore, the input embedding in eq. 8 can potentially help the model to elicit meaningful information even when the masked word is rare, allowing for better exploitation of the training corpus.

We accordingly add 30K lower-frequency words to BERT's original 30K-word vocabulary, which contains frequent words. This results in an addition of approximately 23M parameters over the 110M parameters of BERT$_{\text{BASE}}$ and 30M additional parameters over the 340M parameters of BERT$_{\text{LARGE}}$. Similar vocabulary sizes in leading models have not resulted in increased sense awareness, as reflected in the WiC task results [Liu et al., 2019].

2.3 Single-Supersensed Word Masking

Words that have a single supersense are good anchors for obtaining an unambiguous semantic signal. These words help map contexts to supersenses, in a manner that allows the model to make correct context-based predictions even when the masked word has several supersenses. We therefore favor such words in the masking strategy, choosing 50% of the single-supersensed words in each input sequence to be masked. We stop if 40% of the overall 15% masking budget is filled with single-supersensed words (this rarly happens), and in any case we randomize the choice of the remaining words to complete this budget. In practice, 1 out of 10 words chosen for masking is shown as itself rather than as [MASK], and the prediction task is carried out as is.

3 Semantic Language Model Visualization

A SenseBERT pretrained as described in section 2 (training hyperparameters as in Devlin et al. [2018]), has an immediate non-trivial bi-product. The pre-trained mapping to the supersenses space, denoted S, acts as an additional head predicting a word's supersense given context [see fig. 1(b)]. We thereby effectively attain a semantic-level language model that predicts the missing word's meaning jointly with the standard word-form level language model.

We illustrate the resultant mapping in fig. 2, showing a UMAP dimensionality reduction [McInnes et al., 2018] of the rows of S, which corresponds to the different supersenses. A clear clustering according to the supersense part of speech is apparent in fig. 2(a). We further identify finer-grained semantic clusters, as shown for example in fig. 2(b) and given in more detail in appendix A.

SenseBERT's semantic language model allows predicting a distribution over supersenses rather than over words in a masked position. Fig. 3(a) shows the supersense probabilities assigned by SenseBERT in several contexts, demonstrating the

			BERT	SenseBERT
(a) SemEval-SS	The team used a **battery** of the newly developed "gene probes"		*noun.artifact*	*noun.group*
	Ten shirt-sleeved ringers stand in a circle, one **foot** ahead of the other in a prize-fighter's stance		*noun.quantity*	*noun.body*
(b) WiC	Sent. A: The **kick** must be synchronized with the arm movements.	Sent. B: A sidecar is a smooth drink but it has a powerful **kick**.	*Same*	*Different*
	Sent. A: Please **listen** carefully as I explain.	Sent. B: I like to **listen** to music.	*Different*	*Same*

Figure 4: Example entries from **(a)** the SemEval-SS task, where a model is to predict the supersense of the marked word, and **(b)** the Word in Context (WiC) task where a model must determine weather the underlined word is used in the same/different supersense within Setences A and B. In all displayed examples, taken from the corresponding development sets, SenseBERT predicted the correct label while BERT failed to do so. A quantitative comparison between models is presented in table 1.

model's ability to assign semantically meaningful categories to the masked position even in ambiguous cases.

Finally, we demonstrate that SenseBERT enjoys an ability to automatically view raw text at a lexical semantic level. Fig. 3(b) shows example sentences and their supersense prediction by the pretrained model. This beyond word-form perspective can help the model learn from semantically similar examples which do not share the same phrasing.

4 Lexical Semantics Experiments

In this section, we present quantitative evaluations of SenseBERT, pre-trained as described in section 2. We test the model's performance on a supersense-based variant of the SemEval WSD test sets standardized in Raganato et al. [2017], and on the Word in Context (WiC) task [Pilehvar and Camacho-Collados, 2018] (included in the recently introduced SuperGLUE benchmark [Wang et al., 2019]), both directly relying on the network's ability to perform lexical semantic categorization.

4.1 SemEval-SS: Supersense Disambiguation

We test SenseBERT on a Word Supersense Disambiguation task, a coarse grained variant of the common WSD task. We use SemCor [Miller et al., 1993] as our training dataset (226,036 annotated examples), and the SenseEval [Edmonds and Cotton, 2001, Snyder and Palmer, 2004] / SemEval [Pradhan et al., 2007, Navigli et al., 2013, Moro and Navigli, 2015] suite for evaluation (overall 7253 annotated examples), following Raganato et al. [2017]. For each word in both training and test sets, we change its fine-grained sense label to its corresponding WordNet supersense, and therefore train the network to predict a given word's supersense. We name this Supersense disambiguation task by SemEval-SS. See fig. 4(a) for an example from this modified data set.

We show results on the SemEval-SS task for two different training schemes. In the first, we trained a linear classifier over the 'frozen' output embeddings of the examined model – we do not change the transformer parameter in this scheme. This is a test for the amount of basic lexical semantics readily present in the pre-trained model, easily extricable by further downstream tasks (reminiscent of the semantic probes employed in Hewitt and Manning [2019], Coenen et al. [2019]). In the second training scheme we fine-tuned the examined model on the task, allowing its parameters to change during training (see training details in appendix B). Results attained by employing this training method reflect the model's potential to acquire word-supersense information given its pretraining.

Table 1 shows a comparison between regular BERT and SenseBERT on the supersense disambiguation task. Our semantic level pretraining signal clearly yields embeddings with enhanced word-meaning awareness, relative to embeddings trained with BERT's vanilla word-level

	SemEval-SS Frozen	SemEval-SS Fine-tuned	Word in Context
$\text{BERT}_{\text{BASE}}$	65.1	79.2	–
$\text{BERT}_{\text{LARGE}}$	67.3	81.1	69.6
$\text{SenseBERT}_{\text{BASE}}$	75.6	83.0	70.3
$\text{SenseBERT}_{\text{LARGE}}$	79.5	83.7	**72.1**

Table 1: Results on a supersense variant of the SemEval WSD test standardized in Raganato et al. [2017], which we denote SemEval-SS, and on the Word in Context (WiC) dataset [Pilehvar and Camacho-Collados, 2018] included in the recently introduced SuperGLUE benchmark [Wang et al., 2019]. These tasks require a high level of lexical semantic understanding, as can be seen in the examples in figure 4. For both tasks, SenseBERT demonstrates a clear improvement over BERT in the regular fine-tuning setup, where network weights are modified during training on the task. Notably, $\text{SenseBERT}_{\text{LARGE}}$ achieves state of the art on the WiC task. In the SemEval-SS Frozen setting, we train a linear classifier over pretrained embeddings, without changing the network weights. The results show that SenseBERT introduces a dramatic improvement in this setting, implying that its word-sense aware pre-training (section 2) yields embeddings that carries lexical semantic information which is easily extractable for the benefits of downstream tasks. Results for BERT on the SemEval-SS are attained by employing the published pre-trained BERT models (`bert-base-uncased` and `bert-large-cased`), and the $\text{BERT}_{\text{LARGE}}$ result on WiC is taken from the baseline scores published on the SuperGLUE benchmark [Wang et al., 2019] (no result has been published for $\text{BERT}_{\text{BASE}}$).

signal. $\text{SenseBERT}_{\text{BASE}}$ improves the score of $\text{BERT}_{\text{BASE}}$ on the frozen embeddings setting by over 10 points and $\text{SenseBERT}_{\text{LARGE}}$ improves that of $\text{BERT}_{\text{LARGE}}$ by over 12 points, demonstrating competitive results even without fine-tuning.

In the setting of model fine-tuning, we see a clear demonstration of the model's ability to learn word-level semantics, as $\text{SenseBERT}_{\text{BASE}}$ surpasses the score of $\text{BERT}_{\text{LARGE}}$ by 2 points.

4.2 Word in Context (WiC) Task

We test our model on the recently introduced WiC binary classification task. Each instance in WiC has a target word w for which two contexts are provided, each invoking a specific meaning of w. The task is to identify if the occurrences of w in the two contexts correspond to the same meaning or not, clearly requiring an ability to comprehend the word's semantic category. Similarly to the previous task, the WiC task is defined over supersenses – the negative examples include a word used in two different supersenses and the positive ones include a word used in the same supersense. See fig. 4(b) for an example from this data set.

Results on the WiC task are shown in table 1. $\text{SenseBERT}_{\text{BASE}}$ surpasses a larger vanilla model, $\text{BERT}_{\text{LARGE}}$. A single $\text{SenseBERT}_{\text{LARGE}}$ model achieves the state of the art score in this task, demonstrating unprecedented lexical semantic awareness.

Finally, it is worth noting that SenseBERT gains its lexical semantic knowledge without compromising performance on other downstream tasks – $\text{SenseBERT}_{\text{BASE}}$'s results on the General Language Understanding Evaluation (GLUE) benchmark [Wang et al., 2018] are competitive with the vanilla model, with an overall score of 77.9 versus 78.3 for $\text{BERT}_{\text{BASE}}$.

5 Conclusion

We introduce lexical semantic information into a neural language model's pre-training objective. This results in a boosted word-level semantic awareness of the resultant model, named SenseBERT, which considerably outperforms a regular BERT on a SemEval based Supersense Disambiguation task and achieves state of the art results on the Word in Context task. This improvement was obtained without human annotation, but rather by harnessing an external linguistic knowledge source for inductive bias. Our work indicates that additional semantic signals extending beyond the lexical level can be similarly introduced, allowing the network to elicit further insight without human supervision at the pre-training stage.

References

Pierpaolo Basile. Super-sense tagging using support vector machines and distributional features. In *International Workshop on Evaluation of Natural Language and Speech Tool for Italian*, pages 176–185. Springer, 2012.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell

Stern, and Jakob Uszkoreit. Kermit: Generative insertion-based modeling for sequences. *arXiv preprint arXiv:1906.01604*, 2019.

Massimiliano Ciaramita and Mark Johnson. 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, 2003.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.

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.

Philip Edmonds and Scott Cotton. Senseval-2: overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, 2001.

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.

Adam Kilgarriff. I dont believe in word senses. *Computers and the Humanities*, 31(2):91–113, 1997.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

Andrea Moro and Roberto Navigli. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 288–297, 2015.

Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):10, 2009.

Roberto Navigli, David Jurgens, and Daniele Vannella. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, 2013.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: 10,000 example pairs for evaluating context-sensitive representations. *arXiv preprint arXiv:1808.09121*, 2018.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 87–92, 2007.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *arXiv preprint*, 2019.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 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, 2017.

Nathan Schneider. Lexical semantic analysis in natural language text. *Unpublished Doctoral Dissertation, Carnegie Mellon University*, 2014.

Nathan Schneider and Noah A Smith. A corpus and model integrating multiword expressions and supersenses. 2015.

Benjamin Snyder and Martha Palmer. 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*, 2004.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019.

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

A Supersenses and Their Representation in SenseBERT

We present in table 3 a comprehensive list of WordNet supersenses, as they appear in the WordNet documentation. In fig. 5 we present a Dendrogram of an Agglomerative hierarchical clustering over the supersense embedding vectors learned by SenseBERT in pre-training. The clustering shows a clear separation between Noun senses and Verb senses. Furthermore, we can observe that semantically related supersenses are clustered together (i.e, noun.animal and noun.plant).

B Training Details

As hyperparameters for the fine-tuning, we used $max\_seq\_length = 128$, chose learning rates from $\{5e-6, 1e-5, 2e-5, 3e-5, 5e-5\}$, batch sizes from $\{16, 32\}$, and fine-tuned up to 10 epochs for all the datasets.

C SenseBERT's Results on GLUE

Table 2 shows SenseBERT's results on the GLUE benchmark test set. Our model's result show no significant degradation, and even improvement in some of the tasks.

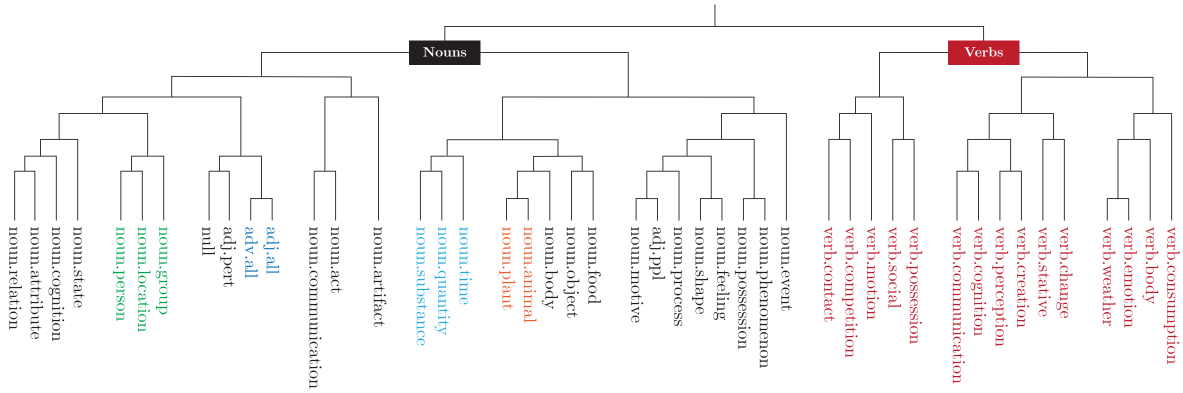


Figure 5: Dendrogram visualization of an Agglomerative hierarchical clustering over the supersense vectors (rows of the classifier S) learned by SenseBERT.

	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE
$\text{BERT}_{\text{BASE}}$	78.3	52.1	93.5	88.9/84.8	87.1/85.8	71.2/89.2	84.6	90.5	66.4
$\text{SenseBERT}_{\text{BASE}}$	77.9	54.6	92.2	89.2/85.2	83.5/82.3	70.3/88.8	83.6	90.6	67.5

Table 2: Results on the GLUE benchmark test set.

Name	Content	Name	Content
adj.all	All adjective clusters	noun.quantity	Nouns denoting quantities and units of measure
adj.pert	Relational adjectives (pertainyms)	noun.relation	Nouns denoting relations between people or things or ideas
adv.all	All adverbs	noun.shape	Nouns denoting two and three dimensional shapes
noun.Tops	Unique beginner for nouns	noun.state	Nouns denoting stable states of affairs
noun.act	Nouns denoting acts or actions	noun.substance	Nouns denoting substances
noun.animal	Nouns denoting animals	noun.time	Nouns denoting time and temporal relations
noun.artifact	Nouns denoting man-made objects	verb.body	Verbs of grooming, dressing and bodily care
noun.attribute	Nouns denoting attributes of people and objects	verb.change	Verbs of size, temperature change, intensifying, etc.
noun.body	Nouns denoting body parts	verb.cognition	Verbs of thinking, judging, analyzing, doubting
noun.cognition	Nouns denoting cognitive processes and contents	verb.communication	Verbs of telling, asking, ordering, singing
noun.communication	Nouns denoting communicative processes and contents	verb.competition	Verbs of fighting, athletic activities
noun.event	Nouns denoting natural events	verb.consumption	Verbs of eating and drinking
noun.feeling	Nouns denoting feelings and emotions	verb.contact	Verbs of touching, hitting, tying, digging
noun.food	Nouns denoting foods and drinks	verb.creation	Verbs of sewing, baking, painting, performing
noun.group	Nouns denoting groupings of people or objects	verb.emotion	Verbs of feeling
noun.location	Nouns denoting spatial position	verb.motion	Verbs of walking, flying, swimming
noun.motive	Nouns denoting goals	verb.perception	Verbs of seeing, hearing, feeling
noun.object	Nouns denoting natural objects (not man-made)	verb.possession	Verbs of buying, selling, owning
noun.person	Nouns denoting people	verb.social	Verbs of political and social activities and events
noun.phenomenon	Nouns denoting natural phenomena	verb.stative	Verbs of being, having, spatial relations
noun.plant	Nouns denoting plants	verb.weather	Verbs of raining, snowing, thawing, thundering
noun.possession	Nouns denoting possession and transfer of possession	adj.ppl	Participial adjectives
noun.process	Nouns denoting natural processes		

Table 3: A list of supersense categories from WordNet lexicographer.