SensePOLAR: Word sense aware interpretability for pre-trained contextual word embeddings

Jan Engler[*]
RWTH Aachen
jan.engler@rwth-aachen.de

Sandipan Sikdar[*]
L3S Research Center
sandipan.sikdar@l3s.de

Marlene Lutz
University of Mannheim

Markus Strohmaier
University of Mannheim, GESIS, CSH Vienna

{marlene.lutz, markus.strohmaier}@uni-mannheim.de

Abstract

Adding interpretability to word embeddings represents an area of active research in text representation. Recent work has explored the potential of embedding words via so-called *polar* dimensions (e.g. good vs. bad, correct vs. wrong). Examples of such recent approaches include SemAxis, POLAR, FrameAxis, and BiImp. Although these approaches provide interpretable dimensions for words, they have not been designed to deal with polysemy, i.e. they can not easily distinguish between different senses of words. To address this limitation, we present SensePOLAR, an extension of the original POLAR framework that enables word-sense aware interpretability for pre-trained *contextual* word embeddings. The resulting *interpretable* word embeddings achieve a level of performance that is comparable to original contextual word embeddings across a variety of natural language processing tasks including the GLUE and SQuAD benchmarks. Our work removes a fundamental limitation of existing approaches by offering users sense aware interpretations for contextual word embeddings.

1 Introduction

The overwhelming success of deep neural networks (DNN) in the last decade has been accompanied by increasing concerns about the lack of *interpretability* (Ribeiro et al., 2016). This problem is amplified in the area of Natural Language Processing (NLP) where *word embeddings* are used as input to machine learning models instead of more classical, understandable features. Traditional (*static*) word embedding models like Word2Vec (Mikolov et al., 2013) or Glove (Pennington et al., 2014), that create one embedding for each word, are currently being replaced by *contextual* word embedding models like BERT (Devlin et al., 2019) which have achieved competitive performance in NLP benchmarks such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016).

To improve interpretability, recent approaches such as SemAxis (An et al., 2018), POLAR (Mathew et al., 2020), FrameAxis (Kwak et al., 2021), and BiImp (Şenel et al., 2022) have explored the potential of embedding words via polar dimensions (e.g. good vs. bad, correct vs. wrong). While these approaches have been useful for interpreting word vectors, they have not been designed to deal with polysemy, i.e. multiple senses of words.

Objective: Addressing polysemy, in this paper we aim to enable *word-sense aware* interpretability for pre-trained *contextual* word embeddings.

Approach: We base our approach on the original POLAR framework (Mathew et al., 2020) and the idea of semantic differentials (Osgood et al., 1957), which are psychometric scales between two antonym words, e.g. “right” ↔ “wrong”. SensePOLAR extends POLAR (Mathew et al., 2020) to contextual word embeddings, and defines polar *sense* instead of polar *word* scales. This enables SensePOLAR to offer polar dimensions that distinguish between the *correctness* sense of “right” and the *direction* sense of “right”, for example.

Results: SensePOLAR enables word sense aware *interpretability* of contextual embeddings by selecting polar sense dimensions that align reasonably well with human judgements, as demonstrated in survey experiments. SensePOLAR exhibits competitive *performance* on various NLP tasks where it is used as input features for a separate model (feature-based approach) as well as directly integrated in the model itself (fine-tuning approach).

Contributions: SensePOLAR introduces the notion of *sense aware* interpretations. To the best of our knowledge, SensePOLAR represents the first (semi-) supervised method that enables word sense aware interpretability for contextual word embeddings. SensePOLAR is publicly available[1].

[*] Equal contribution

[1] https://github.com/JanEnglerRWTH/

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4607–4619
December 7-11, 2022 ©2022 Association for Computational Linguistics

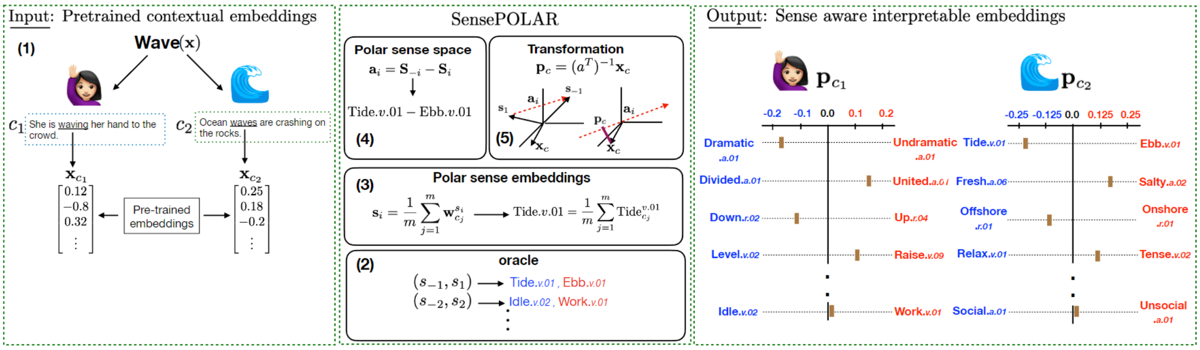


Figure 1: SensePOLAR overview. Pre-trained contextual word embeddings are transformed into an interpretable space where the word's semantics are rated on scales individually encoded by opposite senses such as "good"↔"bad". The scores across the dimensions are representative of the strength of relationship (between word and dimension) which allows us to rank the dimensions and thereby identify the most discriminative dimensions for a word. In this example, the word "wave" is used in two senses: *hand waving* and *ocean wave*. SensePOLAR not only generates dimensions that are representative of individual contextual meanings, the alignment to the respective sense spaces also aligns well with human judgement. SensePOLAR generates neutral scores for dimensions not related to the word in the given context (e.g., "idle"↔"work", "social"↔"unsocial"). We follow the WordNet convention to represent a particular sense of a word. For example, "Tide.v.01" represents the word "tide" in the sense of *surge (rise or move forward).*

2 SensePOLAR

The key idea of SensePOLAR is to transform pretrained word embeddings into an interpretable, *sense aware* space. In this space, each dimension represents a scale on which words are rated, inspired by the semantic differential technique (Osgood et al., 1957). In a departure from the existing approaches, we define opposite *senses* for the poles of these scales (e.g. "left direction" ↔ "right direction"), as opposed to opposite words (e.g. "left" ↔ "right"), as used in Mathew et al. (2020).

Given a contextual word embedding model $\mathcal{M}$, the interpretable embeddings are obtained through the following steps. 1) We use $\mathcal{M}$ to obtain the (non-interpretable) contextual embedding space. 2) We obtain polar senses with contextual information from an oracle. 3) We proceed with generating representative sense embeddings from which we 4) construct the interpretable polar sense space. 5) The original embedding is transformed into the polar sense space, which enables interpretation with regard to opposite sense pairs. We illustrate each step in figure 1 and elaborate them next.

1. Obtaining contextual embeddings: To obtain the embedding of a particular word, we forward the word with its context, i.e. an example sentence, to the embedding model $\mathcal{M}$. The embedding of the corresponding word can then be retrieved from the output of $\mathcal{M}$. Because most models deploy subword tokenization algorithms, such as WordPiece (Wu et al., 2016), embeddings of only *subword* tokens, rather than entire words, are generated by the contextual embedding models. This provides for obtaining representations for out-of-vocabulary words but, at the same time, makes embeddings of even common words not directly available. Following existing literature (McCormick and Ryan, 2019; Bommasani et al., 2020), we compute the embedding of a word by averaging over the embeddings of the constituent tokens.

2. Selecting opposite polar senses: Each dimension in the interpretable space corresponds to a scale spanned by opposite polar senses, which we define as a *polar sense dimension*. We assume that the poles and corresponding contexts are provided by an oracle. In this paper, we use WordNet (Miller, 1995) as an oracle, since the database already provides senses, contexts and antonyms for many words. Each sense of a word is represented by a unique identifier, e.g. "Right.r.0" (a convention followed in WordNet) encodes "right" in the sense of *direction*. From over 6000 sense-antonym pairs that are available in WordNet, we use only a subset (1763) that are annotated with example sentences for both words. After various post-processing steps (cf. Appendix), these example sentences are used as context in step 3.

3. Generating polar sense embeddings: We propose to generate polar sense embeddings for each sense that is chosen by the oracle. Let w denote the word of interest and s the word-sense. Furthermore,

SensePOLAR

let $C_s = \{c_1, ..., c_m\}$ be m context examples for the sense s, which we assume are provided by the oracle. In each context $c \in C_s$, the word w is used in the sense s, e.g. "A strange sound came from the right side." for the word "right" in the sense of *direction* (i.e., we intend to embed "Right.r.04"). We create polar sense embeddings in two steps. First, we input m context examples for a sense s to the embedding model $\mathcal{M}$ and retrieve an embedding $\mathbf{w}_c^s \in \mathbb{R}^d$ of the word w for each context $c \in C_s$. If the word w consists of several subword tokens, the individual subword embeddings are averaged. We also allow for senses consisting of multiple words, e.g. "keep track" ↔ "lose track", where we again average the embeddings of the individual tokens. Second, we compute the average of the contextual word embeddings per sense and define it as the sense embedding $\mathbf{s} \in \mathbb{R}^d$:

$$\mathbf{s} = \frac{1}{m} \sum_{j=1}^{m} \mathbf{w}_{c_j}^s \tag{1}$$

This is a rather straightforward way to represent individual senses of words in a (semi-) supervised manner. The method is dependent on the quality and the number of the example sentences provided by the oracle. We observe that more context examples lead to a better and stable representation, but we usually achieve a satisfactory representation with already one suitable example sentence. This is motivated by the observations in Reif et al. (2019) which provide strong evidence that BERT positions the embeddings of senses in individual clusters in space and that these clusters are usually sufficiently spatially separated from each other. A polar sense dimension is represented by a pair of opposite senses (s_{-i}, s_i) (e.g., "Right.a.02", "Wrong.a.01").

4. Constructing a polar sense space: Given n polar sense dimensions $\mathbf{S} = ((s_{-1}, s_1), ..., (s_{-n}, s_n))$ with their contexts $\mathbf{C} = ((C_{s_{-1}}, C_{s_1}), (C_{s_{-n}}, C_{s_n}))$, we compute the polar sense embedding $\mathbf{s}_i$ for each sense s_i and corresponding context C_{s_i}, following equation 1.

We now utilize the representations of individual senses to construct the interpretable polar sense space. Each polar sense dimension $(s_i, s_{-i}) \in \mathbf{S}$ defines an interpretable scale, which is encoded by the direction vector $\mathbf{a_i}$, defined as follows:

$$\mathbf{a}_i = \mathbf{s}_{-i} - \mathbf{s}_i \tag{2}$$

The direction vectors for all polar sense dimensions are then stacked to obtain the change of basis matrix $\mathbf{a} \in \mathbb{R}^{n \times d}$ for the interpretable polar sense space.

5. Transformation to interpretable embeddings: Finally, an embedding of a word x in a context c, $\mathbf{x}_c$ can be transformed into the polar sense space in the following way. Given $\mathbf{a}$ represents the change of basis matrix, we can compute the polar sense embedding $\mathbf{p}_c$ following the rules of linear algebra:

$$\mathbf{a}^T\ \mathbf{p}_c = \mathbf{x}_c \tag{3}$$

$$\mathbf{p}_c = (\mathbf{a}^T)^{-1}\ \ \mathbf{x}_c \tag{4}$$

The inverse of $\mathbf{a}^T$ is computed by the Moore-Penrose generalized inverse (Ben-Israel and Greville, 2003). The resulting contextual word embedding $\mathbf{p}_c$ in the polar sense space is of dimension $n \times 1$. The absolute value across axis $\mathbf{a_i}$ corresponds to the word's rating on the scale between the polar senses (s_{-i}, s_i) and the sign represents the direction of alignment to a particular pole. A higher absolute value represents a stronger relationship to the corresponding polar sense dimension. This allows us to obtain the most expressive polar sense dimensions for a given word and context.

Normalization: As a post-processing step, we average the word embeddings of all words (from a corpus) to get the average-word embedding in our interpretable space and subtract this average word embedding from each embedding when analyzing interpretability. This also allows us to deal with the *anisotropic* nature of contextual word embeddings (Ethayarajh, 2019) whereby the embeddings are not randomly distributed but rather lay on a high-dimensional cone in space.

3 Evaluation

Note that while SensePOLAR allows for deployment across any contextual word embedding model, in this work, we consider BERT (Devlin et al., 2019) as our model for illustration. We consider a BERT-base model which utilizes 12 transformer (Vaswani et al., 2017) encoder layers and generates embeddings of size 768. The pre-trained BERT-base model was downloaded from Huggingface[2]. In addition, we use WordNet as our oracle with 1763 polar sense pairs.

[2] https://huggingface.co/docs/transformers/model_doc/bert, we used "bert-base-uncased"

ML Model	SVM		FFN	
Task	Base	SensePOLAR	Base	SensePOLAR
Sport	0.941	0.935$\downarrow$ 0.6%	0.961	0.956$\downarrow$ 0.5%
Religion	0.891	0.848$\uparrow$ 4.8%	0.880	0.894$\uparrow$ 1.6%
Computer	0.770	0.750$\downarrow$ 2.6%	0.763	0.727$\downarrow$ 4.7%

Table 1: Performance of the original BERT (Base) embeddings and SensePOLAR embeddings on feature-based tasks with a support vector machine (SVM) and a feed-forward neural network (FFN) classifier. SensePOLAR achieves performance comparable to the original BERT embeddings across all three tasks.

	SQuAD 1.1		SQuAD 2.0	
Metric	Base	SensePOLAR	Base	SensePOLAR
EM	86.92	86.85$\downarrow$ 0.07%	80.88	81.06$\uparrow$ 0.22%
F1	93.15	93.12$\downarrow$ 0.03%	83.87	83.89$\uparrow$ 0.02%

Table 2: Results of fine-tuned BERT embeddings and with SensePOLAR transformed embeddings on the SQuAD benchmark. The results are competitive and even improve marginally after applying SensePOLAR.

3.1 Performance on downstream tasks

The goal of SensePOLAR is to add interpretability to word embeddings without major losses in performance. Hence, we evaluate SensePOLAR on a wide range of NLP downstream tasks. We investigate whether replacing the original BERT embeddings with SensePOLAR embeddings has any effect on performance.

3.1.1 Feature-based tasks

We analyze the effectiveness of SensePOLAR embeddings in a "classical" NLP pipeline, where word embeddings are generated beforehand and are used as input-features to a *separate* machine learning model. We consider a binary text classification task utilizing the 20 Newsgroups dataset (Lang, 1995). The dataset consists of $\sim 20K$ news articles covering 20 types of news. Our experiment follows the structure of Panigrahi et al. (2019), where we only consider the topics sports, computer and religion. For each topic, an article must be classified into one of two categories ("baseball" or "hockey" for sports, "IBM" or "Apple" for computer, "christianity" or "atheism" for religion). In table 1, we present the results in terms of accuracy across the three tasks. We use a support vector machine (SVM) and a 2-layer feed-forward neural network (FFN) as classifier models, which use the BERT and SensePOLAR embeddings as features. Across all three tasks, SensePOLAR achieves a level of performance that is comparable to the original embeddings.

3.1.2 Fine-tuning tasks

Integrating SensePOLAR into fine-tuned models: The models achieving state-of-the-art performances on different NLP tasks usually deploy a task specific network layer (usually a feed-forward network) on top of the embedding layers. The embedding layers and the task specific layers are then fine-tuned on the task specific dataset. Consequently, SensePOLAR embeddings need to be computed considering the fine-tuned version of the embeddings rather than the original pre-trained version. In this particular setting, we propose to utilize the embedding layer of the fine-tuned model (instead of the original pre-trained version) to construct the polar sense space. Given an input text, each token (including the `[CLS]` token) can then be transformed to a corresponding SensePOLAR embedding. Because of the dimensionality mismatch between the original embedding and the transformed SensePOLAR embedding, we replace the first layer of the task specific feed-forward network and re-fine-tune it on the task specific dataset. Note that the weights of the underlying embedding model are frozen during this re-fine-tuning procedure. This is computationally inexpensive as only the task specific layers need to be trained, which are often just 1 or 2-layered feed-forward network.

Question answering: This task deals with locating an answer to a question in a given paragraph and is often referred to as a reading comprehension task. We consider the SQuAD benchmark, including both SQuAD1.1 (Rajpurkar et al., 2016) and SQuAD2.0 (Rajpurkar et al., 2018) versions. The BERT-based QA model consists of the embedding module followed by a span-classification head, which is a 1-layer feed-forward network. The model takes both the question text and the passage text as input. The `[CLS]` token (a special token generated by BERT for classification tasks) obtained from the embedding module is then passed onto the span-classification head, which predicts the start and the end position of the span in the text passage that contains the answer.

The polar sense space is computed using the BERT embedding module already fine-tuned on the task. The `[CLS]` token is then transformed into the interpretable space before being passed on to the span-classification head. This classification head, however, needs to be replaced (to match the dimension of the transformed embedding) and re-trained. In table 2, we report the exact match

(EM) and F1 scores with the original BERT (base) and the SensePOLAR model. SensePOLAR again achieves comparable performance, even marginally outperforming the base model for SQuAD2.0.

Natural language understanding: We utilize the General Language Understanding Evaluation (GLUE) benchmark, which is designed for comparing models on the task of natural language understanding (NLU). It consists of nine tasks that cover a diverse range of text genres, dataset sizes, and degrees of difficulty (Wang et al., 2018). We point the reader to the original paper by Wang et al. (2018) for a general overview of the tasks. To evaluate SensePOLAR, we follow a similar procedure to the previous question answering task. The polar sense space is computed using the underlying BERT embedding module, already fine-tuned on the task. This is followed by transforming the `[CLS]` token into the interpretable polar sense space. The feed-forward layers on top are then replaced and retrained. In table 3, we report the results on GLUE tasks with both the original BERT (Base) and the SensePOLAR embeddings. SensePOLAR achieves competitive performances across all the tasks.

Task	Train size	Metric	Base	SensePOLAR
CoLa	8.5k	Matthew's corr.	56.62	55.05 ↓ 2.77%
SST-2	67k	Accuracy	91.51	91.40 ↓ 0.12%
MRPC	3.7k	Accuracy	84.31	82.84 ↓ 1.74%
		F1	89.00	87.41 ↓ 1.79%
STS-B	7k	Person corr.	89.03	84.17 ↓ 5.46%
QQP	364k	Accuracy	90.59	90.15 ↓ 0.49%
		F1	87.29	86.82 ↓ 0.54%
MNLI	393k	Accuracy	84.49	84.04 ↓ 0.53%
QNLI	105k	Accuracy	91.54	91.58 ↑ 0.04%
RTE	2.5k	Accuracy	63.18	59.93 ↓ 5.14%
WNLI	634	Accuracy	56.34	56.34 ↑↓0%

Table 3: Comparison of the fine-tuned BERT model and the re-fine-tuned BERT model with SensePOLAR embeddings. Mostly, comparable performance is achieved. Slightly worse performance is achieved for tasks with smaller training datasets.

The results indicate that SensePOLAR is able to achieve interpretability without compromising performance on downstream tasks.

3.2 Interpretability

We turn our attention to evaluating the interpretability of SensePOLAR.

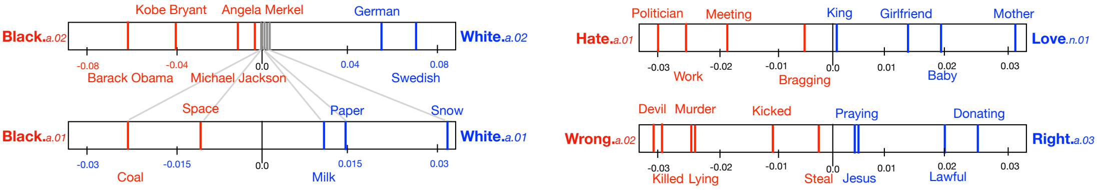
Qualitative analysis: We transform the embeddings of the words into a polar sense space and analyze the position/rating (determined by the signed value on that dimension) of different words on selected dimensions. More specifically, we consider a context in which the word is used and pass it through the BERT module. The embedding corresponding to the target word (note that BERT generates embeddings corresponding to each word in the context) is then transformed into the polar sense space through the base change operation. Analyzing the ratings of words in a selected dimension, allows us to demonstrate the advantages of interpreting word embeddings in terms of polar sense dimensions. We first consider the dimension "Black.a.02"↔"White.a.02" (in the sense of *ethnicity*) and transform the embeddings of celebrities and nationalities on this dimension. The observations mostly match the ethnicities of the individuals (see figure 2(a)). We also consider words such as milk, coal etc. which are not related to "Black.a.02"↔"White.a.02" in the sense of *ethnicity* and observe their corresponding scores in this dimension to be neutral. However, their representation on the dimension "Black.a.01"↔"White.a.01" (in the sense of *color*), captures their semantic well. This demonstrates the benefits of using polar senses as dimensions instead of words, which would have failed to differentiate between the two senses.

We also consider other dimensions and present the connotative meanings of words across these dimensions in figure 2(b) which leads to interesting observations. For example, "politician", "meeting" are more aligned towards "Hate.a.01" (in the sense of *disgust*). Similarly, "murder" and "devil" are aligned towards "Wrong.a.01" (in the sense of *morality*).

In addition to picking out interesting dimensions by hand, we also propose to evaluate interpretability by investigating the most descriptive dimensions of a given word. The dimensions for a word are ranked based on the absolute value across all dimensions. Ideally, the top dimensions should be the most descriptive and fitting for the word. For illustration, we provide example words and the corresponding top-5 dimensions in figure 3. The top dimensions mostly have a high semantic similarity with the word, and they also reasonably align with human judgement.

Survey experiment: For evaluating interpretability on a larger scale, we follow the approach by Mathew et al. (2020) and conduct a human judgment survey. We utilize the crowdsourcing platform Clickworker[3] where we randomly select 15 common English nouns, verbs and adjectives (with

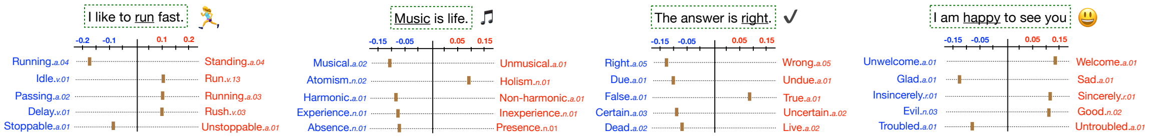
[3] `https://www.clickworker.com/`



(a) Benefits of sense dimensions

(b) Connotative meanings across dimensions

Figure 2: Illustration of polar sense dimensions. (a) SensePOLAR allows for interpretability along multiple senses. "black"↔"white" in the sense of *ethnicity* (top) can be differentiated from "black"↔"white" in the sense of *color* (bottom). Words like "snow"or "coal" - which are not semantically related to ethnicity - score neutral on the upper scale while being clearly distinguishable on the lower scale. (b) The connotative meanings of words can also be investigated through SensePOLAR. For example, "politician" is associated with "hate" while "mother" is associated with "love".



(a) Run

(b) Music

(c) Right

(d) Happy

Figure 3: Illustration of SensePOLAR embeddings. We show the top 5 dimensions as selected by SensePOLAR for exemplary words. The pre-trained embeddings are obtained using BERT. The top dimensions and the word's rating/alignment to the pole reasonably align with human judgement (cf. table 4).

short context) and compute their interpretable embedding with SensePOLAR. Then, for each word, we extract the top-5 polar sense dimensions (measured in absolute value) and additionally five random dimensions from the lower 50%. These 10 dimensions are then presented to the participants in a random order. Participants are asked to select five dimensions that are most representative of a given word and to rate each dimension based on their alignment to one of the poles on a likert scale between 1 and 7 (with 4 as neutral). Each word is assigned 3 annotators. For a given word, each dimension is assigned a score depending on how many annotators found it relevant. We then select the top 5 dimensions based on this score and we consider them as the ground-truth dimension to which we compare the ones selected by SensePOLAR.

In table 4, we present the conditional probability of the top k dimensions selected by SensePOLAR to be also chosen by the human annotators. In the same table, we also report the random chance of getting selected. For the top-1 dimension, agreement is roughly 87% and for the top-2 dimension it is still around 65%, indicating strong alignment with human judgment. We also found that the participant's ratings on these dimensions were the (absolute) highest, showing that the word is strongly connected to one of the polar senses.

Top-k	1	2	3	4	5
SensePOLAR	0.876	0.558	0.312	0.187	0.093
Random	0.5	0.22	0.083	0.023	0.004

Table 4: Alignment with human judgement. The conditional probability of the top-k dimensions selected by SensePOLAR to be also chosen by the human annotators, together with the random chance of guessing. Significantly higher probabilities than random chance are achieved, indicating that the chosen dimensions are meaningful and match human judgment reasonably well.

Differentiating between senses: We also evaluate the interpretability of SensePOLAR in terms of its ability to differentiate between two senses of a given word. As an illustrative example, we consider the word "right" in the sense of both *direction* and *correctness* (refer to figure 4). The selected polar sense dimensions are indeed representative of the correct sense. Note that the original POLAR framework would not be able to differentiate between the senses, given it generates exactly one embedding for a given word.

We follow up with another human judgement experiment where we present the top-10 polar sense dimensions of words with multiple meanings, together with the word's score on these dimensions, to the annotators. The task is to identify in which sense the target word is being used in. We limit this experiment to only two common senses for each

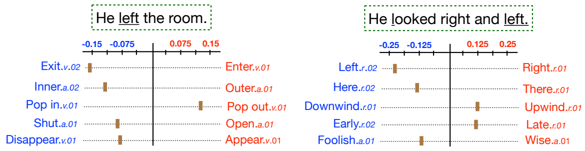


Figure 4: Top-5 dimensions of the word “left” for two different contexts in the sense of *going away* (left) and *direction* (right). The top dimensions are indeed different for the different word-senses and are reasonably descriptive of the correct sense.

word and present the WordNet definitions as the answer possibilities. Thus, by random guessing, an accuracy of 50% would be achieved. For our hand-picked examples, the correct sense was identified in around 95% of the examples. The average inter-participant agreement on the result is around 78%.

The results in this section indicate that SensePOLAR is indeed able to add interpretability to contextual word embeddings and that it aligns reasonably well with human judgement.

4 Applications

We discuss two potential use cases: (i) analyzing bias in contextual embeddings and (ii) explaining model decisions on downstream tasks.

4.1 Bias Analysis

Contextual embedding models have been shown to capture bias from training data (Nadeem et al., 2021; Ahn and Oh, 2021; Bartl et al., 2020; Mozafari et al., 2020). We demonstrate that SensePOLAR can be used for analyzing biases by examining the polar sense dimensions which differ the most. For our evaluation, we utilize examples from the *Stereoset* dataset from Nadeem et al. (2021).

Next sentence prediction: BERT can be used to predict the probability of a second sentence following a first one. Consider the sentence pair “Hispanic people are very diverse. All people like that are criminals.”. Replacing the word “Hispanic” by “American” leads to significant decrease in BERT’s confidence score. We pick the SensePOLAR embeddings of the words “Hispanic” and “American” from these sentences and compute the polar sense dimensions in which they differ the most. Noticeably, the most discriminative dimensions mostly encode senses related to being or not being a legal citizen such as “Documented.a.01”↔“Undocumented.a.01” or “Irregular.a.01”↔“Regular.a.01”. We observe that the word “Hispanic” is rated on the side of *irregular*, *undocumented*, *unrighteous* and *unconcerned*, as opposed to “American”.

Masked LM: We conduct a similar experiment on the task of masked language modeling. Consider the sentence “Because he is black, he works as a [MASK].”. Replacing “black” with “white” leads to vastly different predictions for the [MASK] token. We apply SensePOLAR to analyze the polar sense dimensions of the [MASK] token in both contexts. We find that the most discriminative dimension is “Employed.a.01”↔“Unemployed.a.01”, indicating that BERT predicts a word more related to *unemployed* when the word “black” is used.

4.2 Explaining classifier results

SensePOLAR can further be deployed to explain decisions of classifier models that make use of contextual word embeddings. To illustrate this we consider binary sentiment prediction (positive or negative) on the SST-2 dataset (Socher et al., 2013). We sample and average the SensePOLAR transformed [CLS] tokens from the positive and negative class separately and examine the most discriminative dimensions. We find the most discriminative dimensions to be “sharp”↔“dull”, “unpleasant”↔“pleasant”, “endemic”↔“cosmopolitan”, “soft”↔“loud” and “tasteless”↔“tasteful”. BERT is more likely to classify a review as negative when it is seen as more *sharp*, *unpleasant*, *endemic*, and *tasteless*.

5 Discussion

Next, we discuss issues pertinent to SensePOLAR.

Generalizability: SensePOLAR is applicable to any pre-trained contextual embedding model. It can also be deployed on top of any of the constituent transformer layers. This allows for not only comparing different contextual word embedding models in terms of interpretability or bias analysis but also performing similar analysis across transformer layers of the same embedding model.

Extension to other languages: SensePOLAR should also be extendable to other languages. The only requirement would be to be able to obtain suitable sense antonym pairs as well as example contexts via an oracle.

Interpretable decision-making. In section 4.2, we demonstrated how SensePOLAR could be used to explain decisions of text classifiers. However, the design of SensePOLAR allows for deployment across any other downstream task as well. This is in contrast with existing interpretability methods which are often developed with a particular downstream task in mind.

Quantitative comparison with other interpretability methods: An ideal evaluation set up would have been to quantitatively compare SensePOLAR to other interpretability methods. However, as pointed out in the existing literature (Sundararajan et al., 2017; Sikdar et al., 2021), when two models provide different interpretations, it is difficult to judge if one is better than the other. Involving humans makes it even harder, as one now needs to tease out a person's own subjective biases. Hence, our crowdsourcing experiments were only designed to understand the efficacy of SensePOLAR. Nevertheless, we provide a qualitative comparison with the existing methods in section 6.

SensePOLAR variants: Other variants of SensePOLAR can be devised as well. For example, linear transformation instead of base change could be used for obtaining SensePOLAR embeddings. However, we observed that linear transformation does not preserve the original structure of the embedding space, where the different senses of words are already sufficiently separated. One can also experiment with different normalization techniques, such as scaling or standardization. In this paper, we concentrated on an exhaustive evaluation setup to include more downstream tasks and crowdsourcing experiments rather than exploring other variants. We consider all the above variants promising avenues for future work.

6 Related work

In this section, we briefly summarize previous research on enabling interpretability for both static and contextual word embeddings.

Unsupervised methods: The key idea for this class of methods is to create sparse embeddings, which is achieved through a post-processing step on top of the embeddings (Murphy et al., 2012; Faruqui et al., 2015; Luo et al., 2015). Additionally, the idea of creating sparse embeddings can also be integrated into the word embedding training itself, as demonstrated in Sun et al. (2016); Chen et al. (2017). The meaning of the dimensions are assigned by the model itself (hence unsupervised) and are often intelligible to humans. Notably, Word2Sense (Panigrahi et al., 2019) proposes to create sparse nonnegative vectors through Latent Dirichlet Allocation (LDA). Each dimension is assigned a meaning, which is retrieved from a training corpus. The methods discussed above are specific to static word embeddings. Berend (2020) extends some of these ideas to contextual word embeddings.

(Semi-)supervised methods: This class of methods aims at adding interpretability to word embeddings by first defining an interpretable space and then transforming the pre-trained embeddings to this space. In this space, each dimension spans between two pole words. While SemAxis (An et al., 2018) proposes to use antonym pairs retrieved from ConceptNet (Speer et al., 2017), the POLAR framework (Mathew et al., 2020) utilizes the semantic differential technique pioneered by Osgood (Osgood et al., 1957). Similarly, BiImp (Şenel et al., 2022) proposes to use opposite semantic concepts as poles. Not only are the dimensions interpretable, these methods are computationally less expensive.

Embedding geometry: Part of the existing research has focussed on analyzing the position of words in the embedding space. Ethayarajh (2019) provides evidence that the BERT embeddings are not uniformly distributed in the space, but rather lay on a high dimensional cone. Reif et al. (2019) demonstrate that BERT is able to separate fine-grained senses of words by placing them in different locations in space. Similar observations are made by Schmidt and Hofmann (2020) as well.

Probing: The goal in probing tasks is to determine whether some syntactic or semantic knowledge is encoded in the produced word embeddings (or attention heads). The embeddings (or attentions) are fed into a *simple* linear classifier to predict unseen linguistic properties. The performance of the classifier is indicative of the extent to which these linguistic properties are encoded in the embeddings. For BERT, these probing experiments have demonstrated that the layers on the top are more contextual (Ethayarajh, 2019) and the layers at the center contain a large amount of syntactic information (Hewitt and Manning, 2019; Goldberg, 2019; Jawahar et al., 2019; Chi et al., 2020). The semantic information is generally spread across the entire network (Tenney et al., 2019; Zhao et al., 2020; Lin et al., 2019).

Visual explanations: Finally, recent work has also considered visualizing attention in transformer layers to explain contextual language models (Hoover et al., 2020; Vig, 2019). Similarly, visualizing word embeddings can also aid in explaining what a model learns as demonstrated in Liu et al. (2017); Heimerl and Gleicher (2018); Boggust et al. (2022) (static) and Sevastjanova et al. (2021); Berger (2020) (contextual).

Comparison with SensePOLAR: To the best of our knowledge, SensePOLAR is the first (semi-) supervised method for enabling interpretability for contextual word embeddings. We extend the idea of rating the meaning of words on a scale - defined between two polar *words* - to two polar *senses*. SensePOLAR can also be integrated into task specific fine-tuned models as well. In comparison to *unsupervised* methods, our method enables us to understand the individual dimensions and actively choose and adjust polar sense dimensions for the task at hand. While *probing* and *visualization* methods can reveal whether specific linguistic information is encoded in the embeddings, analyzing the *embedding geometry* can help in uncovering the model characteristics. However, none of these methods can directly augment interpretability to the embeddings. Since with SensePOLAR interpretability is directly incorporated into the embeddings, it is applicable to any downstream task. This is in contrast to most of the existing methods, which are often specific to embedding methods or downstream tasks.

7 Conclusion

We introduced SensePOLAR which enables word sense aware interpretability for contextual word embeddings. The key idea is to project word embeddings onto an interpretable space which is constructed from polar sense pairs obtained from an oracle. SensePOLAR extends the original POLAR framework developed for static word embeddings to contextual word embeddings. We demonstrated that the obtained interpretable embeddings align well with human judgement. Moreover, SensePOLAR could be integrated into fine-tuned models and can be deployed to specific applications like bias analysis and explaining prediction results of classifier models.

8 Limitations

Underlying embedding models: SensePOLAR uses embeddings of polar senses to build an interpretable subspace. Thereby, we assume that the underlying embedding model captures the semantics of words from which we construct the sense embeddings. As a result, SensePOLAR is dependent on the quality of the underlying contextual word embedding model. Compared to the original POLAR framework proposed in (Mathew et al., 2020), the present approach also depends on the ability of the model to capture individual word-senses with sufficient accuracy.

Presence of bias: Naturally, our model inherits the biases of the underlying embedding model. The word “physics”, for example, has a high rating towards “male” on the polar sense scale of “male” ↔ “female”. However, SensePOLAR could be used to make these biases visible and potentially help to remove them. One can also tap into state-of-the-art bias mitigation methods (e.g. Ahn and Oh (2021); Bartl et al. (2020); Mozafari et al. (2020)) to address this issue.

Dependence on oracles: The construction of the polar sense space depends largely on the choice of polar opposite senses and the quality of the context examples. Using the example of WordNet, we have shown how a general model can be created. However, we observed that rare senses and low-quality example sentences can lead to poor results. Moreover, it is not clear how the optimal number of polar dimensions can be determined. Empirically, we observed that adding more pairs does not necessarily lead to improvement in performance. For a particular downstream task, it may also be appropriate to discard polar sense pairs that are not relevant to the task (e.g. if they never occur in the corpus).

Counter-intuitive rating of words. We find that in some cases the rating of words on the polar sense scales does not coincide with human judgement. The word “doctor”, for example, is highly skewed towards “guilty” on a scale from “innocent” ↔ “guilty”, which does not match the typical perception of doctors. We believe this is because word embeddings by design are shaped by their context. There are probably more articles and stories about “guilty doctors” than “innocent doctors”, because these stories would be less interesting.

Acknowledgements

Sandipan Sikdar was supported in part by RWTH Aachen Startup Grant No. StUpPD384-20.

References

Jaimeen Ahn and Alice Oh. 2021. Mitigating language-dependent ethnic bias in BERT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 533–549, Online and Punta Cana, Dominican Republic.

Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2018. SemAxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2450–2461, Melbourne, Australia.

Marion Bartl, Malvina Nissim, and Albert Gatt. 2020. Unmasking contextual stereotypes: Measuring and mitigating BERT's gender bias. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 1–16.

Adi Ben-Israel and Thomas NE Greville. 2003. *Generalized inverses: Theory and Applications*, volume 15. Springer Science & Business Media.

Gábor Berend. 2020. Sparsity makes sense: Word sense disambiguation using sparse contextualized word representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8498–8508.

Matthew Berger. 2020. Visually analyzing contextualized embeddings. In *2020 IEEE Visualization Conference (VIS)*, pages 276–280. IEEE.

Angie Boggust, Brandon Carter, and Arvind Satyanarayan. 2022. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. In *27th International Conference on Intelligent User Interfaces*, pages 746–766.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* , pages 4758–4781.

Yunchuan Chen, Ge Li, and Zhi Jin. 2017. Learning sparse overcomplete word vectors without intermediate dense representations. In *International Conference on Knowledge Science, Engineering and Management*, pages 3–15. Springer.

Ethan A Chi, John Hewitt, and Christopher D Manning. 2020. Finding universal grammatical relations in multilingual bert. *arXiv preprint arXiv:2005.04511*.

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.

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*, pages 55–65, Hong Kong, China.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500, Beijing, China.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Florian Heimerl and Michael Gleicher. 2018. Interactive analysis of word vector embeddings. In *Computer Graphics Forum*, volume 37, pages 253–265. Wiley Online Library.

John Hewitt and Christopher D. Manning. 2019. 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, Minneapolis, Minnesota.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exbert: A visual analysis tool to explore learned representations in transformer models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Haewoon Kwak, Jisun An, Elise Jing, and Yong-Yeol Ahn. 2021. Frameaxis: characterizing microframe bias and intensity with word embedding. *PeerJ Computer Science*, 7:e644.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy.

Shusen Liu, Peer-Timo Bremer, Jayaraman J Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. 2017. Visual exploration of semantic relationships in neural word embeddings. *IEEE transactions on visualization and computer graphics*, 24(1):553–562.

Hongyin Luo, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2015. Online learning of interpretable word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1687–1692.

Binny Mathew, Sandipan Sikdar, Florian Lemmerich, and Markus Strohmaier. 2020. The polar framework: Polar opposites enable interpretability of pre-trained word embeddings. In *Proceedings of The Web Conference 2020*, pages 1548–1558.

Chris McCormick and Nick Ryan. 2019. BERT word embeddings tutorial. https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of the 1st International Conference on Learning Representations*, 2013.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on bert model. *PloS one*, 15(8):e0237861.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012*, pages 1933–1950.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371.

Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. 1957. *The Measurement of Meaning*. 47. University of Illinois press.

Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. word2sense: sparse interpretable word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5692–5705.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100 ,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. *Advances in Neural Information Processing Systems*, 32.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Florian Schmidt and Thomas Hofmann. 2020. Bert as a teacher: Contextual embeddings for sequence-level reward. *arXiv preprint arXiv:2003.02738*.

Lütfi Kerem Şenel, Furkan Şahinuç, Veysel Yücesoy, Hinrich Schütze, Tolga Çukur, and Aykut Koç. 2022. Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts. *Information Processing & Management*, 59(3):102925.

Rita Sevastjanova, Aikaterini-Lida Kalouli, Christin Beck, Hanna Schäfer, and Mennatallah El-Assady. 2021. Explaining contextualization in language models using visual analytics. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 464–476.

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. 2021. Integrated directional gradients: Feature interaction attribution for neural nlp models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 865–878.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Sparse word embeddings using l1 regularized online learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2915–2921.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

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.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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*.

Mengjie Zhao, Philipp Dufter, Yadollah Yaghoobzadeh, and Hinrich Schütze. 2020. Quantifying the contextualization of word representations with semantic class probing. *arXiv preprint arXiv:2004.12198*.

A Appendix

A.1 Post-processing

We point out some issues when using WordNet directly as our oracle and present ways to address them.

A.1.1 Sense Post-processing

We notice that word-senses identified by WordNet can be overly granular (e.g. according to WordNet there are five different polar sense pairs for "wet" ↔ "dry"). To get rid of redundant senses, we propose to use dimension-reduction methods such as *variance maximization* and *orthogonality maximization* from the original POLAR framework (Mathew et al., 2020). Alternatively, one could merge similar senses if the cosine similarity between their sense embeddings is high. Selecting or discarding a rare sense is often task specific.

A.1.2 Low-quality Example Sentences

The polar sense embeddings are dependent on the quality of the context (i.e., example sentences demonstrating a particular sense). While deploying WordNet as source for contexts, we encountered some issues which we elaborate on next.

Flections. When constructing polar sense dimensions, we use the respective word in its basic form and extract the embedding from the example context. However, in WordNet's example sentences, words often do not appear in their basic form, but in inflections (e.g. "She walk**s** with a slight limp" for "Walk.v.01").

Synonyms. Occasionally, the word itself is not present in the context sentence but is replaced by a synonym (e.g. "the **right** answer" for "Correct.a.01").

Misspellings. We observe that the example sentences often contain spelling mistakes (e.g. "to**u**ngued lightning" for "Tongued.a.01")

Mixed-up examples. In some cases, the example sentences of a sense are identical to those of the opposite pole (e.g. "we **docked** at noon" for "Undock.v.01").

These problems need to be addressed with manual checks of the obtained polar sense pairs and context sentences.

A.2 Sense Scales

Instead of rating words on scales defined between two pole words (e.g. "left" ↔ "right"), our scales are defined between two pole word-senses (e.g.

"left" ↔ "right" in the sense of *direction*). While the static POLAR framework rates the word "correct" highly on the dimension "left" ↔ "right", we expect our framework to rate it low on the dimension "left" ↔ "right" in the sense of *direction* but rate it high on the dimension "wrong" ↔ "right" in the sense of *correctness*.

To this aim, we analyze whether our constructed representative sense embeddings encode enough sense-related information. As an illustrative example, we consider the word "right" which is used in the senses of *direction*, *correctness* and *lawfulness*. For each context, we compute the SensePOLAR embeddings for the word and rank the dimensions based on the absolute value.

Sense-Scales \ Context	he went to the **right**	his argument is **right**	film **rights**
Direction "left" ↔ "right"	1^{st}	38^{th}	32^{nd}
Correct "wrong" ↔ "right"	44^{th}	1^{st}	291^{st}
Lawful "wrong" ↔ "right"	55^{th}	27^{th}	9^{th}

Table 5: Ability of SensePOLAR in differentiating between senses. We consider the word "right" in three different contexts and obtain a ranking of the dimensions for each case. We report the rank of a SensePOLAR dimension in each of the three contexts in each row. For example, the dimension "left" ↔ "right" representing the sense of *direction* is ranked first for the context "he went to the right", while it is ranked 38th and 32nd respectively in the other two contexts. SensePOLAR is indeed able to identify the correct sense dimensions depending on the context.

In table 5, we report the ranks of the polar sense dimensions for each context. For the word "right" in the context of *direction* "he went to the right", the dimension "left" ↔ "right" is selected as the most representative dimension (rank 1), while the *correctness* and the *lawful* dimensions are ranked much lower (44 and 55 respectively). Similar results are obtained for the other contexts (see table 5).

These results indicate that the sense-dimensions of SensePOLAR precisely captures the individual semantics of the senses.

A.3 Computational Requirements

SensePOLAR embeddings of words for a given context can be obtained at low cost if the polar sense space is pre-computed. We provide such an implementation along with the submission and encourage readers to review it. Our implementation can even be run on a personal computer.

Given n polar sense dimensions, inversion of the matrix can be computed in the worst case in $O(n^3)$. Since in our case $n = 1762$, the computation is very fast. Moreover, this computation needs to be performed only once.

Retraining the task-specific feed-forward layer with SensePOLAR embeddings was performed on a computing server with 1 TB RAM, 72 cores, each Intel Xeon Gold 6140 CPU at 2.30 GHZ and 2 Tesla P100-PCIE, 16GB GPUs. We would like to reiterate that the retraining is also quite cheap given only the task-specific feed-forward layer needs to be trained.