The POLAR Framework: Polar Opposites Enable Interpretability of Pre-Trained Word Embeddings

Binny Mathew*
IIT Kharagpur, India
binnymathew@iitkgp.ac.in

Sandipan Sikdar*
RWTH Aachen University, Germany
sandipan.sikdar@cssh.rwth-aachen.de

Florian Lemmerich
RWTH Aachen University, Germany
florian.lemmerich@cssh.rwth-aachen.de

Markus Strohmaier†
RWTH Aachen University & GESIS, Germany
markus.strohmaier@cssh.rwth-aachen.de

ABSTRACT

We introduce 'POLAR' — a framework that adds interpretability to pre-trained word embeddings via the adoption of semantic differentials. Semantic differentials are a psychometric construct for measuring the semantics of a word by analysing its position on a scale between two polar opposites (e.g., cold – hot, soft – hard). The core idea of our approach is to transform existing, pre-trained word embeddings via semantic differentials to a new "polar" space with interpretable dimensions defined by such polar opposites. Our framework also allows for selecting the most discriminative dimensions from a set of polar dimensions provided by an oracle, i.e., an external source. We demonstrate the effectiveness of our framework by deploying it to various downstream tasks, in which our interpretable word embeddings achieve a performance that is comparable to the original word embeddings. We also show that the interpretable dimensions selected by our framework align with human judgement. Together, these results demonstrate that interpretability can be added to word embeddings without compromising performance. Our work is relevant for researchers and engineers interested in interpreting pre-trained word embeddings.

CCS CONCEPTS

• **Computing methodologies → Machine learning approaches**.

KEYWORDS

word embeddings, neural networks, interpretable, semantic differential

*Both authors contributed equally to this research.

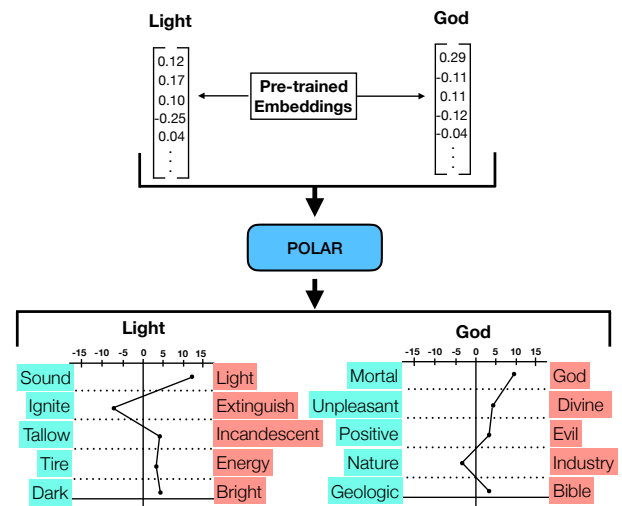†The work was done during internship at RWTH Aachen University

Figure 1: *The POLAR Framework.* **The framework takes pre-trained word embeddings as an input and generates word embeddings with interpretable (polar) dimensions as an output. In this example, the embeddings are generated by applying POLAR to embeddings pre-trained on Google News dataset with Word2Vec.**

1 INTRODUCTION

Dense distributed word representations such as Word2Vec [21] and GloVe [27] have been established as a key step for technical solutions for a wide variety of natural language processing tasks including translation [44], sentiment analysis [36], and image captioning [43]. While such word representations have substantially contributed towards improving performance of such tasks, it is usually difficult for humans to make sense of them. At the same time, interpretability of machine learning approaches is essential for many scenarios, for example to increase trust in predictions [30], to detect potential errors, or to conform with legal regulations such as General Data Protection Regulation (GDPR [6]) in Europe that recently established a "right to explanation". Since word embeddings are often crucial for downstream machine learning tasks, the non-interpretable nature of word embeddings often impairs a deeper understanding of their performance in downstream tasks.

Problem. We aim to add interpretability to an arbitrarily given pre-trained word embedding via post-processing in order make embedding dimensions interpretable for humans (an illustrative example is provided in figure 1). Our objective is explicitly *not improving* performance per se, but adding interpretability while *maintaining* performance on downstream tasks.

Approach. The POLAR framework utilizes the idea of semantic differentials (Osgood et al. [25]) that allows for capturing connotative meanings associated with words and applies it to word embeddings. To obtain embeddings with interpretable dimensions, we first take a set of polar opposites from an oracle (e.g., from a lexical database such as WordNet), and identify the corresponding polar subspace from the original embedding. The basis vectors of this polar subspace are calculated using the vector differences of the polar opposites. The pre-trained word vectors are then projected to this new polar subspace, which enables the interpretation of the transformed vectors in terms of the chosen polar opposite pairs. Because the set of polar opposites could be potentially very large, we also discuss and compare several variations to select expressive subsets of polar opposite pairs to use as basis for the new vector space.

Results and contribution. We evaluate our approach with regard to both *performance* and *interpretability*. With respect to performance, we compare the original embeddings with the proposed POLAR embeddings in a variety of downstream tasks. We find that in all cases the performance of POLAR embeddings is competitive with the original embeddings. In fact, for a few tasks POLAR even outperforms the original embeddings. Additionally, we evaluate interpretability with a human judgement experiment. We observe that in most cases, but not always, the dimensions deemed as most discriminative by POLAR, align with dimensions that appear most relevant to humans. Our results are robust across different embedding algorithms. This demonstrates that we can augment word embeddings with interpretability without much loss of performance across a range of tasks.

To the best of our knowledge, our work is the first to apply the idea of semantic differentials - stemming from the domain of psychometrics - to word embeddings. Our POLAR framework provides two main advantages: (i) It is agnostic w.r.t. the underlying model used for obtaining the word vectors, i.e., it works with arbitrary word embedding frameworks such as GloVe and Word2Vec. (ii) as a post-processing step for pre-trained embeddings, it neither requires expensive (re-)training nor access to the original textual corpus. Thus, POLAR enables the addition of interpretability to arbitrary word embeddings *post-hoc*. To facilitate reproducibility of our work and enable their use in practical applications, we make our implementation of the approach publicly available[1].

2 BACKGROUND

In this section, we provide a brief overview of prior work on interpretable word embeddings as well as the semantic differential technique pioneered by Osgood.
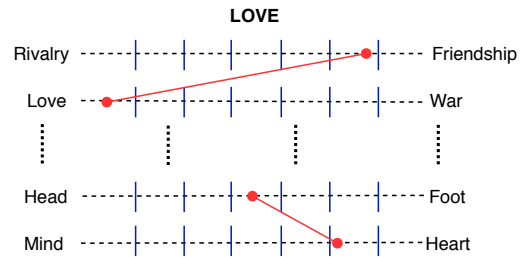

Figure 2: *An example semantic differential scale.* **The example reports the response of an individual to the word *Love*. Each dimension represents a semantically polar pair. A response close to the edge means a strong relation with the dimension and a response near the middle means no clear relation.**

2.1 Interpretable word embeddings

One of the major issues with low-dimensional dense vectors utilized by Word2Vec [21] or GloVe [27] is that the generated embeddings are difficult to interpret. Although the utility of these methods has been demonstrated in many downstream tasks, the meaning associated with each dimension is typically unknown. To solve this, there have been few attempts to introduce some sense of interpretability to these embeddings [8, 23, 26, 37].

Several recent efforts have attempted to introduce interpretability by making embeddings sparse. In that regard, Murphy et al. proposed to use a Non-Negative Sparse Embedding (NNSE) in order to to obtain sparse and interpretable word embeddings [23]. Fyshe et al. [10] introduce a joint Non-Negative Sparse Embedding (JNNSE) model to capture brain activation records along with texts. The joint model is able to capture word semantics better than text based models. Faruqui et al. [8] transform the dense word vectors derived from Word2Vec using sparse coding (SC) and demonstrate that the resulting word vectors are more similar to the interpretable features used in NLP. However, SC usually suffers from heavy memory usage since it requires a global matrix. This makes it quite difficult to train SC on large-scale text data. To tackle this, Luo et al. [17] propose an online learning of interpretable word embeddings from streaming text data. Sun et al. [38] also use an online optimization algorithm for regularized stochastic learning which makes the learning process efficient. This allows the method to scale up to very large corpus.

Subramanian et al. [37] utilize denoising k-sparse autoencoder to generate efficient and interpretable distributed word representations. The work by Panigrahi et al. [26] is to the best our knowledge, among the existing research, closest to our work. The authors propose *Word2Sense* word embeddings in which each dimension of the embedding space corresponds to a fine-grained sense, and the non-negative value of the embedding along a dimension represents the relevance of the sense to the word. Word2Sense is a generative model which recovers senses of a word from the corpus itself. However, these methods would not be applicable if the user does not have access to the corpus itself. Also, such models have high computation costs, which might make it infeasible for many users who wish to add interpretability to word embeddings.

---

[1]Code: https://github.com/Sandipan99/POLAR

Our work differs from the existing literature in several ways. The existing literature does not necessarily provide dimensions that are actually interpretable to humans in an intuitive way. By contrast, our method represents each dimension as a pair of polar opposites given by an oracle (typically end users, a dictionary, or some vocabulary), which assigns direct meaning to a dimension. Moreover, massive computation costs associated with training these models have led researchers to adopt pre-trained embeddings for their tasks. The proposed POLAR framework, being built on top of pre-trained embeddings and not requiring the corpus itself, suits this common design.

2.2 Semantic Differentials

The semantic differential technique by Osgood et al. [25] is used to measure the connotative meaning of abstract concepts. This scale is based on the presumption that a concept can have different dimensions associated with it, such as the property of speed, or the property of being good or bad. The semantic differential technique is meant for obtaining a person's psychological reactions to certain concepts, such as persons or ideas, under study. It consists of a number of bipolar words that are associated with a scale. The survey participant indicates an attitude or opinion by checking on any one of seven spaces between the two extremes of each scale. For an example, consider Figure 2. Here, each dimension of the scale represents a semantically polar pair such as 'Rivalry' and 'Friendship', 'Mind' and 'Heart'. A participant could be given a word (such as 'Love') and asked to select points along each dimension, which would represent his/her perception of the word. A point closer to the edge would represent a higher degree of agreement with the concept. The abstract nature of semantic differential allows it to be used in a wide variety of scenarios. Often, antonym pairs are used as polar opposites. For example, this is related to work by An et al. [1], in which the authors utilize polar opposites as semantic axes to generate domain-specific lexicons as well as capturing semantic differences in two corpora. It is also similar to the tag genome (Vig et al. [40]), a concept that is used to elicit user preferences in tag-based systems.

Overall, the semantic differential scale is a well established and widely used technique for observing and measuring the meaning of concepts such as information system satisfaction (Xue et al. [42]), attitude toward information technology (Bhattacherjee and Premkumar [3]) information systems planning success (Doherty et al. [7]), perceived enjoyment (Luo et al. [18]), or website performance (Huang [13]).

This paper brings together two isolated concepts: word embeddings and semantic differentials. We propose and demonstrate that the latter can be used to add interpretability to the former.

3 METHODOLOGY

In this section, we introduce the POLAR Framework and elaborate how it generates interpretable word embeddings. Note that we do not train the used word embeddings from scratch rather we generate them by post-processing embeddings already trained on a corpus.

3.1 The POLAR framework

Consider a corpus with vocabulary $\mathcal{V}$ containing V words. For each word $v \in \mathcal{V}$, the corresponding embedding trained using an algorithm a (Word2Vec, GLoVE) is denoted by $\vec{\mathbb{W}}_v^a \in \mathbb{R}^d$, where d denotes the dimension of the embedding vectors. In this setting, let $\mathbb{D} = [\vec{\mathbb{W}}_1^a, \vec{\mathbb{W}}_2^a, \vec{\mathbb{W}}_3^a, \ldots, \vec{\mathbb{W}}_V^a] \in \mathbb{R}^{V \times d}$ denote the set of pretrained word embeddings which is used as input to the *POLAR* framework. Note that $\vec{\mathbb{W}}_i^a$ is a unit vector with $||\vec{\mathbb{W}}_i^a|| = 1$.

The key idea is to identify an interpretable subspace and then project the embeddings to this new subspace in order to obtain interpretable dimensions which we call *POLAR dimensions*. To obtain this subspace we consider a set of N polar opposites. In this paper, we use a set of antonyms (e.g., hot–cold, soft–hard etc.) as an initial set of polar opposites, but this could easily be changed to arbitrary other polar dimensions. Typically, we assume that these set of polar opposites are provided by some *oracle*, i.e., an external source that provides polar, interpretable word pairs.

Given these set of N polar opposites, we now proceed to generate the polar opposite subspace. Let the set of polar opposites be denoted by $\mathbb{P} = \{(p_z^1, p_{-z}^1), (p_z^2, p_{-z}^2), \ldots, (p_z^N, p_{-z}^N)\}$. Now the direction of a particular polar opposite (p_z^1, p_{-z}^1) can be obtained by:

$$\vec{dir}_1 = \vec{\mathbb{W}}_{p_z^1}^a - \vec{\mathbb{W}}_{p_{-z}^1}^a \tag{1}$$

The direction vectors are calculated across all the polar opposites and stacked to obtain $dir \in \mathbb{R}^{N \times d}$. Note that dir represents the change of basis matrix for this new (polar) embedding subspace $\mathbb{E}$. Let a word v in the embedding subspace $\vec{\mathbb{E}}$ be denoted by $\vec{\mathbb{E}}_v$. So for v we have by the rules of linear transformation:

$$dir^T \vec{\mathbb{E}}_v = \vec{\mathbb{W}}_v^a \tag{2}$$

$$\vec{\mathbb{E}}_v = (dir^T)^{-1} \vec{\mathbb{W}}_v^a \tag{3}$$

Note that each dimension (*POLAR dimension*) in this new space $\vec{\mathbb{E}}$, can be interpreted in terms of the polar opposites. The inverse of the matrix dir is accomplished through Moore-Penrose generalized inverse [2] usually represented by dir^+. While this can in most settings be computed quickly and reliably, there is one issue: when the number of polar opposites (i.e., *POLAR dimensions*), is similar to the number of dimensions of the original embedding, the change of basis matrix dir becomes ill conditioned and hence the transformed vector $\vec{\mathbb{E}}_v$ becomes meaningless and unreliable. We discuss this in more detail later in this paper. Note that with N polar opposites, the worst case complexity of calculating the generalized inverse is $O(N^3)$. Since $N \ll V$ and the inverse needing to be calculated just once, the computation is overall very fast (e.g. < 5 seconds for $N = 1,468$). Performance can further be improved using parallel architecture (0.29 seconds on a 48 core machine). The overall architecture of the model is presented in Figure 3(a).

We illustrate using a toy example in Figure 3(b). In this setting the embeddings trained on a corpus are of dimension d. The polar opposites $\mathbb{P}$ in this case are ('hard','soft') and ('cold','hot'). In the first step, we obtain the direction of the polar opposites, which is then followed by projecting the words ('Alaska' in this example) into this new subspace. After the transformation, 'Alaska' gets aligned
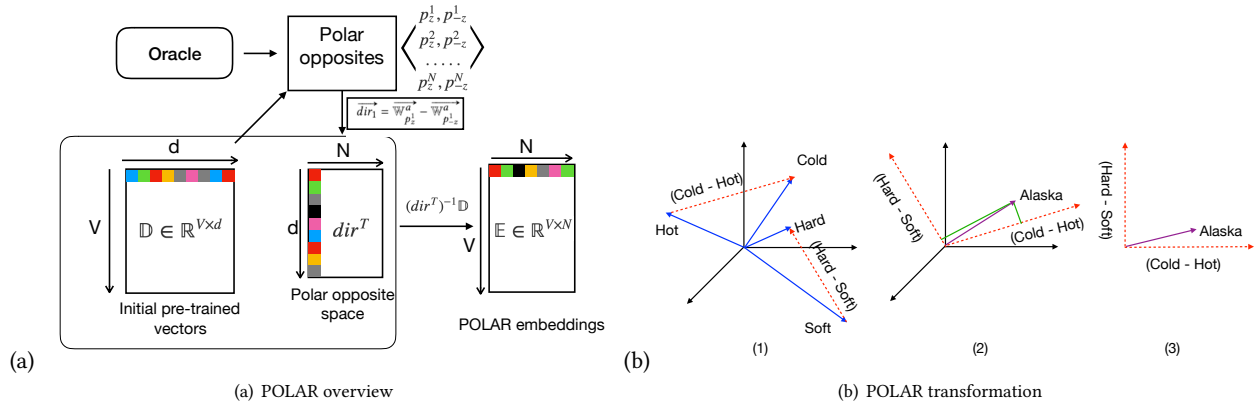
(a) POLAR overview

(b) POLAR transformation

Figure 3: (a) *Visual illustration of the POLAR framework.* **A set of pre-trained embeddings ($\mathrm{R}^{V \times d}$) represents the input to our approach, and we assume that an Oracle provides us with a list of polar opposites with which we generate the polar opposite space ($\mathrm{R}^{d \times N}$). We apply change of basis transform to obtain the final embeddings ($\mathrm{R}^{V \times N}$). Note that V is the size of the vocabulary, N is the number of polar opposites and d is the dimension of the pre-trained embeddings. (b)** *POLAR transformation.* **In this example the original size of the embeddings is three and we consider two polar opposites (cold, hot) and (hard, soft). In the first step (1) we obtain the direction of the polar opposites (vectors in the original space represented in** blue**) which also represent the change of basis vectors for the polar subspace (represented by** red **dashed lines). In the second step (2) we project the original word vectors ('Alaska' in this case) to this polar subspace. After the transformation, 'Alaska' gets aligned more to the (cold, hot) direction which is much more related to 'Alaska' than the (hard, soft) direction (3).**

more to the (cold–hot) direction which is much more related to 'Alaska' than the (hard–soft) direction.

While in our explanations we only use antonyms, polar opposites could also include other terms, such as political terms representing politically opposite ideologies (e.g. republican vs. democrat) that could be obtained from political experts, or people representing opposite views (e.g. Chomsky vs. Norvig) that could be obtained from domain experts.

3.2 Selecting POLAR dimensions

We also design a set of algorithms to select suitable dimensions as POLAR embeddings from a larger set of candidate pairs of polar opposites. For all the algorithms, we use the same notation with P denoting the initial set of polar opposite vectors ($|P| = N$) and O denoting the reduced set of polar opposite vectors (initialized to ϕ) obtained utilizing the algorithms discussed below. K denotes the specified size of O.

Random selection. In this simple method, we randomly sample K polar opposite vectors from P and add them to O. For experimental evaluation, We repeat this procedure with different randomly selected sets and report the mean value across runs.

Variance maximization. In this method, we select the dimensions (polar opposite vectors) based on the value of their variance on the vocabulary. Typically, for each dimension, we consider the value corresponding to each word in the vocabulary when projected on it and then calculate the variance of these values across each dimension. We take the top K polar opposite vectors (POLAR dimensions) from P which have the highest value of variance and add them to O. This is motivated by the idea that the polar opposites with

maximum variance encode maximum information.

Orthogonality maximization. The primary idea here is to select a subset of polar opposites in such a way that the corresponding vectors are maximally orthogonal. Typically, we follow a greedy approach to generate the subset of polar vectors as presented in Algorithm 1. First, we obtain a vector with maximum variance (as in Variance maximization) from P and add it to O. In each of the following steps we subsequently add a vector to O such that it is maximally orthogonal to the ones that are already in O. A candidate vector z at any step is selected via -

$$z = \underset{x \in P}{\mathrm{argmin}} \frac{1}{|O|} \sum_{n=1}^{n=|O|} \vec{O}_i \cdot \vec{x} \qquad (4)$$

We then continue the process until a specified number of dimensions K is reached.

4 EXPERIMENTAL SETUP

Next, we discuss our experimental setup including details on the used polar opposites, training models, and baseline embeddings. As our framework does not require any raw textual corpus for embedding generation, we use two popular pretrained embeddings:

(1) **Word2Vec** embeddings [21][2] trained on Google News dataset. The model consists of 3 million words with an embedding dimension of 300.

---

[2]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM

Algorithm 1: Orthogonality maximization

Input : P – Initial set of polar opposite vectors, K – the required size

Output: O – The reduced set of polar opposite vectors consisting of K vectors

```
1  O ← ∅;
2  U ← Select a vector from P with maximum variance;
3  O ← O ∪ U;
4  P ← P − U //Remove U from P;
5  for i ← 2 to K do
6  │   min_vec ← ∅;
7  │   min_score ← +∞;
8  │   foreach curr_vec ∈ P do
9  │   │   curr_score ← Average_Score(P,curr_vec);
10 │   │   if curr_score < min_score then
11 │   │   │   min_score ← curr_score;
12 │   │   │   min_vec ← curr_vec;
13 │   │   end
14 │   │   O ← O ∪ min_vec;
15 │   │   P ← P − min_vec //Remove U from P;
16 │   end
17 end
18 return O;
```

(2) **GloVe** embeddings [27][3] trained on Web data from Common Crawl. The model consists of 1.9 million words with embedding dimension set at 300.

As **polar opposites** we adopt the antonym pairs used in previous literature by Shwartz et al. [35][4]. These antonym pairs were collected from the Lenci/Benotto Dataset [33] as well as the EVA-Lution Dataset [34]. The antonyms in both datasets were combined to obtain a total of 4,192 antonym pairs. After this, we removed duplicates to get 1,468 unique antonym pairs. In the following experiments, we will be using these 1,468 antonym pairs to generate POLAR embeddings[5]. However, we study the effect of size of the embeddings on different downstream tasks later in this paper. It is important to reiterate at this point that we do not intend to improve the performance of the original word embeddings. Rather we intend to add interpretability *without much loss in performance* in downstream tasks.

5 EVALUATION OF PERFORMANCE

We follow the same procedure as in Faruqui et al. [8], Subramanian et al. [37][6], and Panigrahi et al. [26] to evaluate the performance of our method on downstream tasks. We use the embeddings in the following downstream classification tasks: news classification, noun phrase bracketing, question classification, capturing discriminative

---

[3]We used the Common Crawl embeddings with 42B tokens: https://nlp.stanford.edu/projects/glove/
[4]The datasets are available here: https://github.com/vered1986/UnsupervisedHypernymy/tree/master/datasets
[5] In case, a word in the antonym pair is absent from the Word2Vec/GloVe vocabulary, we ignore that pair. Ergo, we have 1,468 pairs for GloVe but 1,465 for Word2Vec.
[6]We use the evaluation code given in https://github.com/harsh19/SPINE

attributes, word analogy, sentiment classification and word similarity. In all these experiments we use the original word embeddings as baseline and compare their performance with POLAR-interpretable word embeddings.

5.1 News Classification

As proposed in Panigrahi et al. [26], we consider three binary classification tasks from the 20 news-groups dataset[7].

Task. Overall the dataset consists of three classes of news articles: (a) sports, (b) religion and (c) computer. For the 'sports' class, the task involves a binary classification problem of categorizing an article to 'baseball' or 'hockey' with training/validation/test splits (958/239/796). For 'religion', the classification problem involves 'atheism' vs. 'christian' (870/209/717) while for 'computer' it involves 'IBM' vs. 'Mac' (929/239/777).

Method. Given a news article, a corresponding feature vector is obtained by averaging over the vectors of the words in the document. We use a wide range of classifiers including support vector classifiers (SVC), logistic regression, random forest classifiers for training and report the test accuracy for the model which provides the best validation accuracy.

Result. We report a comparison of classification accuracies between classifiers with the original embeddings vs. those with POLAR interpretable embeddings in Table 1 for the three tasks. For Word2Vec embeddings, POLAR performs almost as good as the original embeddings in all the cases. In fact, the accuracy improves with POLAR for 'religion' classification by 4.5%. We achieve similar performance with GloVe embeddings as well.

5.2 Noun phrase bracketing

Task. The task involves classifying noun phrases as left bracketed or right bracketed. For example, given the noun phrase *blood pressure medicine*, the task is to decide whether it is {*(blood pressure) medicine*} (left) or {*blood (pressure medicine)*} (right). We use the dataset proposed in Lazaridou et al. [15] which constructed the Noun phrase bracketing dataset from the penn tree bank[20] that consists of 2, 227 noun phrases with three words each.

Method. Given a noun phrase, we obtain the feature vector by averaging over the vectors of the words in the phrase. We use SVC (with both linear and RBF kernel), Random forest classifier and logistic regression for the task and use the model with the best validation accuracy for testing.

Result. We report the accuracy score in Table 1. In both Word2Vec and GloVe, we obtain similar results when using POLAR instead of the corresponding original vectors (0.792, 0.761). The results are even closer in case of GloVe (0.764, 0.757).

5.3 Question Classification

Task. The question classification task [16] involves classifying a question into six different types, e.g., whether the question is about a location, about a person or about some numeric information. The training dataset contains 5, 452 labeled questions, and the test

---

[7]http://qwone.com/ jason/20Newsgroups/

Table 1: *Performance of POLAR across different downstream tasks.* We compare the embeddings generated by POLAR against the initial Word2Vec and GloVe vectors on a suite of benchmark downstream tasks. For all the tasks, we report the accuracy when using the original embeddings vis-a-vis when using POLAR embeddings (the classification model is same in both cases). In all the tasks we achieve comparable results with POLAR for both Word2Vec and GloVe (we report the percentage change in performance as well). In fact, for Religious News classification and Question classification we perform better than the original embeddings trained on Word2Vec.

Tasks		Word2Vec	Word2Vec w/ POLAR	GloVe	GloVe w/ POLAR
News Classification	Sports	0.947	0.922 ↓ 2.6%	0.951	0.951 ↕ 0.0%
	Religion	0.812	0.849 ↑ 4.6%	0.876	0.852 ↓ 2.7%
	Computers	0.737	0.717 ↓ 2.7%	0.804	0.802 ↓ 0.2%
Noun Phrase Bracketing		0.792	0.761 ↓ 3.9%	0.764	0.757 ↓ 0.9%
Question Classification		0.954	0.958 ↑ 0.4%	0.962	0.964 ↑ 0.2%
Capturing Discriminative Attributes		0.639	0.628 ↓ 1.7%	0.633	0.638 ↑ 0.7%
Word Analogy		0.740	0.704 ↓ 4.8%	0.751	0.727 ↓ 3.1%
Sentiment Classification		0.816	0.821 ↑ 0.6%	0.808	0.818 ↑ 1.2%

dataset consists of 500 questions. By isolating 10% of the training questions for validation, we use train/validation/test splits of 4,906/546/500 questions respectively.

Method. As in previous tasks, we create feature vectors for a question by averaging over the word vectors of the constituent words. We train with different classification models (SVC, random forest and logistic regression) and report the best accuracy across the trained models.

Result. From Table 1 we can see that POLAR embeddings are able to marginally outperform both Word2Vec (0.954 vs. 0.958) and GloVe embeddings (0.962 vs. 0.964).

5.4 Capturing Discriminative Attributes

Task. The Capturing Discriminative Attributes task (Krebs et al. [14]) was introduced at SemEval 2018. The aim of this task is to identify whether an attribute could help discriminate between two concepts. For example, a successful system should determine that *red* is a discriminating attribute in the concept pair *apple, banana*. The purpose of the task is to better evaluate the capabilities of state-of-the-art semantic models, beyond pure semantic similarity. It is a binary classification task on the dataset[8] with training/validation/test splits of 17,501/2,722/2,340 instances. The dataset consists of triplets of the form (*concept1, concept2, attribute*).

Method. We used the unsupervised distributed vector cosine baseline as suggested in Krebs et al. [14]. The main idea is that the discriminative attribute should be close to the word it characterizes and farther from the other concept. If the cosine similarity of *concept1* and *attribute* is greater than the cosine similarity of *concept2* and *attribute*, we say that the *attribute* is discriminative.

Result. We report the accuracy in Table 1. We achieve comparable performance when using POLAR embeddings instead of the original ones. In fact accuracy is slightly better in case of GloVe.

5.5 Word Analogy

Task. The word analogy task was introduced by Mikolov et al. [2013c; 2013a] to quantitatively evaluate the models' ability of encoding the linguistic regularities between word pairs. The dataset

contains 5 types of semantic analogies and 9 types of syntactic analogies. The semantic analogy subset contains 8,869 questions, typically about places and people, like "Athens is to Greece as X (Paris) is to France", while the syntactic analogy subset contains 10,675 questions, mostly focusing on the morphemes of adjective or verb tense, such as "run is to running as walk to walking".

Method. Word analogy tasks are typically performed using vector arithmetic (e.g. 'France + Athens - Greece') and finding the word closest to the resulting vector. We use the Gensim [29][9] to evaluate the word analogy task.

Result. We achieve comparable (although not quite as good) performances with POLAR embeddings, seecTable 1). The performance is comparatively better in case of GloVe.

5.6 Sentiment Analysis

Task. The sentiment analysis task involves classifying a given sentence into a positive or a negative class. We utilize the Stanford Sentiment Treebank dataset [36] which consists of train, validation and test splits of sizes 6,920, 872 and 1,821 sentences respectively.

Method. Given a sentence, the features are generated by averaging the embeddings of the constituent words. We use different classification models for training and report the best test accuracy across all the trained models.

Result. We report the accuracy in Table 1. We achieve comparable performance when using POLAR embeddings instead of the original ones. In fact accuracy is slightly better in case of both GloVe and Word2Vec.

5.7 Word Similarity

Task. The word similarity or relatedness task aims to capture the similarity between a pair of words. In this paper, we use Simlex-999 (Hill et al. [12]), WS353-S and WS353- R (Finkelstein et al. [9]), MC (Miller and Charles [22]), RG (Rubenstein and Goodenough [32]), MEN (Bruni et al. [5]), RW (Luong et al. [19]) and MT-771 (Halawi et al. [11], Radinsky et al. [28]). Each pair of words in these datasets is annotated by a human generated similarity score.

---

[8]The dataset is available here: https://github.com/dpaperno/DiscriminAtt

[9]https://radimrehurek.com/gensim/models/keyedvectors.html

Table 2: *Performance of POLAR on word similarity evaluation across multiple datasets*. Similarity between a word pair is measured by human annotated scores as well as cosine similarity between the word vectors. For each dataset, we report the spearman rank correlation ρ between the word pairs ranked by human annotated score as well as the cosine similarity scores (we report the percentage change in performance as well). POLAR consistently outperforms the baseline original embeddings (refer to Table 2) in case of GloVe while in case of Word2Vec, the performance of POLAR is comparable to the baseline original embeddings for most of the datasets.

Task	Dataset	Word2Vec	Word2Vec w/ POLAR	GloVe	Glove w/ POLAR
	Simlex-999	0.442	0.433 ↓ 2.0%	0.374	0.455 ↑ 21.7%
	WS353-S	0.772	0.758 ↓ 1.8%	0.695	0.777 ↑ 11.8%
	WS353-R	0.635	0.554 ↓ 12.8%	0.600	0.683 ↑ 13.8%
	WS353	0.700	0.643 ↓ 8.1%	0.646	0.733 ↑ 13.5%
Word Similarity	MC	0.800	0.789 ↓ 1.4%	0.786	0.869 ↑ 10.6%
	RG	0.760	0.764 ↑ 0.5%	0.817	0.808 ↓ 1.1%
	MEN	0.771	0.761 ↓ 1.3%	0.736	0.783 ↑ 6.4%
	RW	0.534	0.484 ↓ 9.4%	0.384	0.451 ↑ 17.5%
	MT-771	0.671	0.659 ↓ 1.8%	0.684	0.678 ↓ 0.9%

Method. For each dataset, we first rank the word pairs using the human annotated similarity score. We now use the cosine similarity between the embeddings of each pair of words and rank the pairs based on this similarity score. Finally, we report the Spearman's rank correlation coefficient ρ between the ranked list of human scores and the embedding-based rank list. Note that we consider only those pairs of words where both words are present in our vocabulary.

Result. We can observe that POLAR consistently outperforms the baseline original embeddings (refer to Table 2) in case of GloVe. In case of Word2Vec, the performance of POLAR is almost as good as the baseline original embeddings for most of the datasets.

5.8 Sensitivity to parameters

5.8.1 Effect of POLAR dimensions. In Table 1 and Table 2, we report the performance of POLAR when using 1, 468 dimensions (i.e., antonym pairs). Additionally, we studied in detail the effects of POLAR dimension size on performance across the downstream tasks. As mentioned in section 3.2, we utilize three strategies for dimension selection: (i) maximal orthogonality, (ii) maximal variance and (iii) random. In Figure 4 and Figure 5, we report the accuracy across all the downstream tasks for different POLAR dimensions across the three dimension selection strategies. We also report the performance of the original pre-trained embeddings in the same figures. Typically, we observe an increasing trend i.e., the accuracy improves with increasing POLAR dimensions. But, competitive performance is achieved with 400 dimensions for most of the tasks (even lesser for Sports and Religion News classification, Noun phrase bracketing, Question classification and Sentiment classification).

However, a numerical inconvenience occurs when the size of POLAR dimensions approaches the dimension of the pre-trained embeddings (300 in our case). In this event, the columns of the change of basis matrix *dir* loses the linear independence property making it ill-conditioned for the computation of the inverse. Hence the transformed vector of a word v, $\vec{\mathbb{E}}_v = (dir^T)^{-1} \vec{\mathbb{W}}_v^a$ (with the pseudo inverse $dir^+ = (dir^* dir)^{-1} dir^*$, dir^* denotes Hermitian transpose), is meaningless and unreliable. We hence eliminate the

region surrounding this critical value (300 in this case) from our dimension related experiments (cf. Figure 4 and Figure 5). We believe this to be a minor inconvenience as comparable results can be obtained with lower POLAR dimensions and even better with higher ones. Nevertheless, there are several regularization techniques available for finding meaningful solutions [24] for critical cases. However, exploring these techniques is beyond the scope of this paper.

We would further like to point out that while dimension reduction is useful for comparing performance, it's not always useful to reduce the dimension itself. As argued in Murphy et al. [23], interpretability often results in sparse representations and representations should model a wide range of features in the data. One of the main advantages of our method is it's flexibility. It can be made very sparse to capture a large array of meaning. It can also have low dimensions, and still be interpretable which is ideal for low resource corpora.

5.8.2 Effect of the pre-trained model. Note that we have considered embeddings trained with both Word2Vec and GloVe. The results are consistent across both the models (refer to tables 1 and 2). This goes to show that POLAR is agnostic w.r.t the underlying training model i.e., it works across specific word embedding frameworks. Furthermore, the embeddings are trained on different corpora (Google News dataset for Word2Vec and Web data from Common Crawl in case of GloVe). This demonstrates the POLAR should work irrespective of the underlying corpora.

5.8.3 Effect of dimension selection. Assuming that the number of polar opposites could be large, we have proposed three methods for selecting (reducing) dimensions. Results presented in figures 4 and 5 allow us to compare the effectiveness of these methods. Typically, we observe that all these methods have similar performances except in lower dimensions where orthogonal and variance maximization seem to perform better than random. For higher dimensions, the performances are similar.
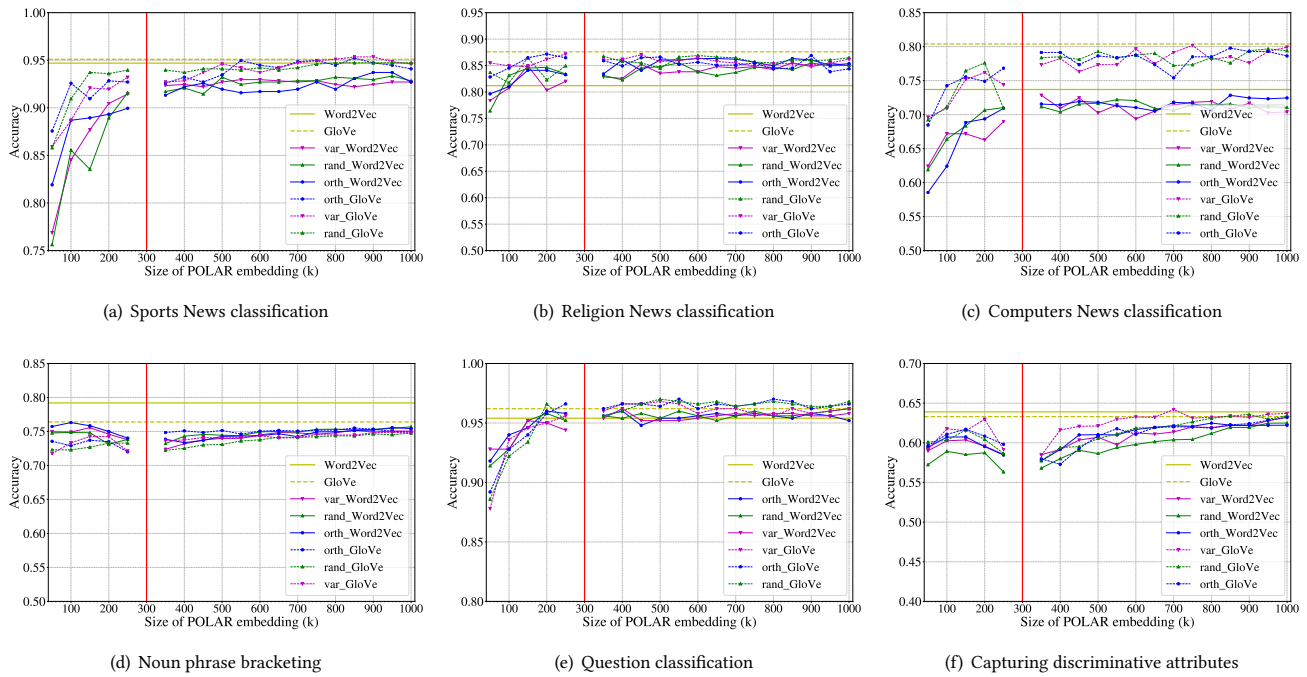
Figure 4: *Dependency on embedding size.* We report the accuracy of POLAR as well as the original embeddings for different downstream tasks for varying sizes (k) of the embeddings. The dimensions are selected using three strategies - 1. random (rand), 2. maximizing orthogonality (orth) and 3. maximizing variance (var). We also report the accuracy obtained using the original Word2Vec and GloVe embeddings. Although the performance improves as the embedding size increases, comparable performance is achieved with a dimension size of 200. However, when the dimension size approaches 300 (the dimension of the pre-trained embeddings), the change of basis vector becomes ill-conditioned and the embeddings become unreliable. We hence intentionally leave this region from the plots.

6 EVALUATION OF INTERPRETABILITY

Next, we evaluate the interpretability of the dimensions produced by the POLAR framework.

6.1 Qualitative Evaluation

As an initial step, we sample a few arbitrary words from the embedding and transform them to POLAR embeddings using their Word2Vec representation. Based on the absolute value across the POLAR dimensions, we obtain the top five dimensions for each of these words. In Table 3, we report the top 5 dimensions for these words. We can observe that the top dimensions have high semantic similarity with the word. Furthermore, our method is able to capture multiple interpretations of the words. This demonstrates that POLAR seems to be able to produce interpretable dimensions which are easy for humans to recognize.

6.2 Human Judgement

In order to assess the interpretability of the embeddings generated by our method, we design a human judgement experiment. For that purpose, we first select a set of 100 words randomly, considering only words with proper noun, verb, and adjective POS tags to make the comparison meaningful.

For each word, we sort the dimensions based on their absolute value and select the top five POLAR dimensions (see Section 6.1 for details) to represent the word. Additionally, we select five dimensions randomly from the bottom 50% from the sorted dimensions according to their polar relevance. These ten dimension are then shown as options to three human annotators each in random order with the task of selecting the five dimension which to him/her best characterize the target word. The experiment was performed on GloVe with POLAR embeddings.

For each word, we assign each dimension a score depending on the number of annotators who found it relevant and select the top five (we call these ground truth dimensions). We now compare this with the top 5 dimensions obtained using POLAR. In Table 4 we report the conditional probability of the top k dimensions in the ground truth to be also in POLAR. This conditional probability essentially measures, given the annotator has selected top k dimensions, what is the probability that they are also the ones selected by polar or simply put, in what fraction of cases the top k dimensions overlap with the Polar dimensions. In the same table we also note the random chance probabilities of the ground truth dimensions to be among the POLAR dimensions (e.g., the top dimension ($k = 1$) selected by the annotators, has a random chance probability of 0.5 to be also among the POLAR dimensions).
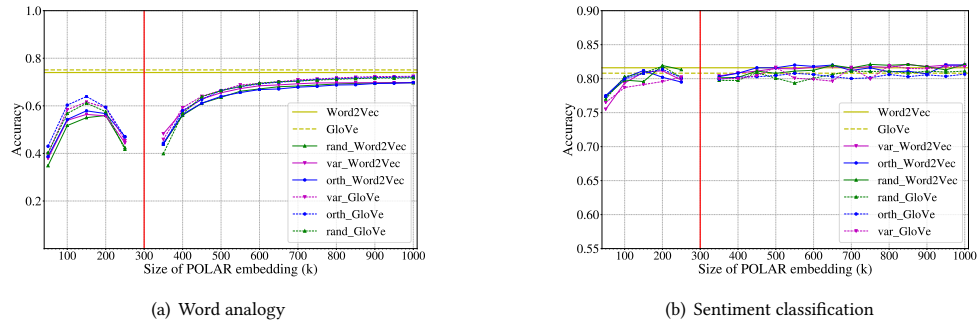
(a) Word analogy

(b) Sentiment classification

Figure 5: *Dependency on embedding size.* We report the accuracy of POLAR in (a) word analogy tasks and (b) sentiment classification task for different sizes of the embeddings. For both tasks, the performance improves with embedding size. For word analogy comparable results are obtained at dimensions close to 600 while for sentiment classification it is around 200. Owing to unreliability of the results, we leave out the results around the region of dimension size 300.

We observe that probabilities for POLAR to be much higher than random chance for all values of k (refer to table 4). In fact, the top two dimensions selected by POLAR are very much aligned to human judgement, achieving a high overlap of 0.87 and 0.67. On the other hand, the remaining 3 dimensions, although much better than random chance, do not reflect human judgement well. To delve deeper into it, we compared the responses of the 3 annotators for each word and obtained the average overlap in dimensions among them. We observed that on average the annotators agree mostly on 2-3 (mean = 2.4) dimensions (which also match with the ones selected by POLAR) but tend to differ for the rest. This goes to show that once we move out of the top 2-3 dimensions, human judgement becomes very subjective and hence difficult for any model to match. We interpret this as POLAR being able to capture the most important dimensions well, but unable to match more subjective dimensions in many scenarios.

6.3 Explainable classification

Apart from providing comparable performance across different downstream tasks, the inherent interpretability of POLAR dimensions also allows for explaining results of black box models. To illustrate, we consider the Religious news classification task and build a Random Forest model using word averaging as the feature. Utilizing the LIME framework [31], we compare the sentences that were inferred as "Christian" by the classifier to those which were classified as "Atheist". In figure 6 we consider two such examples

Table 4: *Evaluation of the interpretability.* We report the conditional probability of the top k dimensions as selected by the annotators to be among the ones selected by POLAR. We also report random chance probabilities for the selected dimension to be among the POLAR dimensions for different values of k. The probabilities for POLAR are significantly higher than the random chance probabilities indicating alignment with human judgement.

Top k	1	2	3	4	5
GloVe w\ POLAR	0.876	0.667	0.420	0.222	0.086
Random chance	0.5	0.22	0.083	0.023	0.005

and report the dimensions/features (as well as their corresponding values) that were given higher weights in each case. Notably, the dimensions like 'criminal - pastor', 'backward - progressive', 'faithful - nihilistic' are given more weights for classification which also corroborates well with the human understanding. Note that the feature values across the POLAR dimensions which essentially represent projection to the polar opposite space, are relevant as well. For example, the article classified as "Christian" in figure 6 has a value -0.15 for the dimension 'criminal - pastor' which means that it is more aligned to 'pastor' while it is the opposite in case of 'Atheist'. This demonstrates that POLAR dimensions could be used to help explain results in black box classification models.

Table 3: *Evaluation of interpretability.* The top 5 dimensions of each word using Word2Vec transformed POLAR Embedding. Note that our model is able to capture multiple interpretations of the words. Furthermore, the dimensions identified by our model are easy for humans to understand as well.

Phone		Apple		Star		Cool		run	
Mobile	Stationary	Apple	Orange	Actor	Cameraman	Cool	Geek	Run	Stop
Fix	Science	Touch	Vision	Psychology	Reality	Naughty	Nice	Flight	Walk
Ear	Eye	Look	Touch	Sky	Water	Fight	Nice	Race	Slow
Solo	Symphonic	Mobile	Stationary	Darken	Twinkle	Freeze	Heat	Organized	Unstructured
Dumb	Philosophical	Company	Loneliness	Sea	Sky	Add	Take	Labor	Machine

Atheist

Dimension	Value
Criminal - Pastor	-0.15
Pastor - Unbeliever	0.31
Backward - Progressive	0.10
Faithful - Nihilistic	0.26
Crowd - Desert	0.53

(a)

Christian

Dimension	Value
Criminal - Pastor	0.36
Faithful - Nihilistic	-0.19
Backward - Progressive	0.38
Misled - Redirect	0.28
Bind - Loose	-0.12

(b)

Figure 6: *Explaining classification results.* **We present the POLAR dimension as well as their corresponding value for the features that were assigned more weights by the classifier when classifying the articles as (a) "Atheist" or (b) "Christian". Dimensions like "criminal - pastor", "faithful - nihilist" are assigned more weights. The selected dimensions also align well with human understanding.**

7 DISCUSSION

Finally, we discuss potential application domains for the presented POLAR framework, as well as limitations and further challenges.

7.1 Applications

In this paper we introduced POLAR, a framework for adding interpretability to pre-trained word embeddings. Through a set of experiments on different downstream tasks we demonstrated that one can add interpretability to existing word embeddings while mostly maintaining performance in downstream tasks. This should encourage further research in this direction. We also believe that our proposed framework could prove useful in a number of further areas of research. We discuss some of them below.

Explaining results of black box models. In this paper, we have demonstrated how the interpretable dimensions of POLAR are useful in explaining results of a random forest classifier. We believe POLAR could also be used for generating counterfactual explanations (Wachter et al. [41]) as well. Depending on the values of the POLAR dimensions one might be able to explain why a particular data point was assigned a particular label.

Identifying bias. Bolukbasi et al. [4] noted the presence of gender bias in word embeddings, giving the example of 'Computer programmer' being more aligned to 'man' and 'Homemaker' being more aligned to 'woman'. Preliminary results indicate that POLAR dimensions can assist in measuring such biases in embeddings. For example, the word 'Nurse' has a value of -3.834 in the dimension 'man–women' which indicates that it is more strongly aligned with woman. We believe that POLAR dimensions might help identifying bias across multiple classes.

'Tunable' Recommendation. Vig et al. [39] introduced a conversational recommender system which allows user to navigate from one item to other along dimensions represented by tags. Typically, given a movie, a user can find similar movies by tuning one or more tags (e.g., a movie like 'Pulp fiction' but less dark). POLAR should allow for designing similar recommendation systems based on word embeddings in a more general way.

7.2 Limitations

Dependence of Interpretability on underlying corpora. Although we demonstrated that the performance of POLAR on downstream tasks is similar to the original embeddings, its interpretability is highly dependent on the underlying corpora. For example consider the word 'Unlock'. The dimensions 'Fee–Freebie' and 'Power–Weakness' were selected by the human judges to be the most interpretable, while these two dimension were not present in the top 5 dimension of the POLAR framework. On closer examination, we observe that the top dimensions of POLAR were not directly related to the word Unlock ('Foolish–Intelligent', 'Mobile–Stationary', 'Curve–Square', 'Fool–Smart', 'Innocent–Trouble'). We believe this was primarily due to the underlying corpora used to generate the baseline embeddings.

Identifying relevant polar opposites. Although we assume that the polar opposites are provided to POLAR by an oracle, selecting relevant polar opposites is critical to the performance of POLAR which can be challenging for smaller corpora. If antonym pairs are used as polar opposites, methods such as the ones introduced by An et al. [1] could be used to find polar words as well as handling smaller corpora.

Bias in underlying embeddings. Since, we use pre-trained embeddings, the biases present in them are also manifested in the POLAR embeddings as well. However, we believe the methods developed for removing bias, with minor modifications could be extended to POLAR as well.

8 CONCLUSION AND FUTURE DIRECTION

We have presented a novel framework (POLAR) that adds interpretability to pre-trained embeddings without much loss of performance in downstream tasks. We utilized the concept of Semantic Differential from psychometrics to transform pre-trained word embeddings into interpretable word embeddings. The POLAR framework requires a set of polar opposites (e.g. antonym pairs) to be obtained from an oracle, and then identifies a corresponding subspace (the polar subspace) from the original embedding space. The original word vectors are then projected to this polar subspace to obtain new embeddings for which the dimensions are interpretable.

To determine the effectiveness of our framework we considered several downstream tasks that utilize word embeddings, for which we systematically compared the performance of the original embeddings vs. POLAR embeddings. Across all tasks, we obtained competitive results. In some cases, POLAR embeddings even outperformed the original ones. We further performed human judgement experiments to determine the degree of interpretability of these embeddings. We observed that in most cases the dimensions deemed as most discriminative by POLAR aligned well with human understanding.

Future directions. An obvious next step would be to extend our framework to other languages as well as corpora. This would allow us to understand word contexts and biases across different cultures. We could also include other sets of polar opposites as well. Another interesting direction would be to investigate whether the POLAR framework could be applied to add interpretability to sentence and document embeddings which then might be utilized for explaining – for example – search results.

REFERENCES

[1] Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2018. SemAxis: A Lightweight Framework to Characterize Domain-Specific Word Semantics Beyond Sentiment. *ACL* (2018), 2450–2461.

[2] Adi Ben-Israel and Thomas NE Greville. 2003. *Generalized inverses: theory and applications*. Vol. 15. Springer Science & Business Media.

[3] Anol Bhattacherjee and G Premkumar. 2004. Understanding changes in belief and attitude toward information technology usage: A theoretical model and longitudinal test. *MIS quarterly* (2004), 229–254.

[4] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NIPS*. 4349–4357.

[5] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research* 49 (2014), 1–47.

[6] EU Council. 2016. EU Regulation 2016/679 General Data Protection Regulation (GDPR). *Official Journal of the European Union* 59, 6 (2016), 1–88.

[7] Neil F Doherty, CG Marples, and A Suhaimi. 1999. The relative success of alternative approaches to strategic information systems planning: an empirical analysis. *The Journal of Strategic Information Systems* 8, 3 (1999), 263–283.

[8] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A Smith. 2015. Sparse Overcomplete Word Vector Representations. In *ACL*, Vol. 1. 1491–1500.

[9] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems* 20, 1 (2002), 116–131.

[10] Alona Fyshe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2014. Interpretable Semantic Vectors from a Joint Model of Brain-and Text-Based Meaning. In *ACL*. 489–499.

[11] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *SIGKDD*. 1406–1414.

[12] Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* 41, 4 (2015), 665–695.

[13] Ming-Hui Huang. 2005. Web performance scale. *Information & Management* 42, 6 (2005), 841–852.

[14] Alicia Krebs, Alessandro Lenci, and Denis Paperno. 2018. Semeval-2018 task 10: Capturing discriminative attributes. In *SemEval*. 732–740.

[15] Angeliki Lazaridou, Eva Maria Vecchi, and Marco Baroni. 2013. Fish transporters and miracle homes: How compositional distributional semantics can help NP parsing. In *EMNLP*. 1908–1913.

[16] Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING*. 1–7.

[17] Hongyin Luo, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2015. Online learning of interpretable word embeddings. In *EMNLP*. 1687–1692.

[18] Margaret Meiling Luo, Sophea Chea, and Ja-Shen Chen. 2011. Web-based information service adoption: A comparison of the motivational model and the uses and gratifications theory. *Decision Support Systems* 51, 1 (2011), 21–30.

[19] Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*. 104–113.

[20] MP Marcus, B Santorini, and MA Marcinkiewicz. 1993. Building a large annotated corpus of english: the Penn Treebank. *Computational linguistics-Association for Computational Linguistics* 19, 2 (1993), 313–330.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS*. 3111–3119.

[22] George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes* 6, 1 (1991), 1–28.

[23] Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. *COLING* (2012), 1933–1950.

[24] Arnold Neumaier. 1998. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM review* 40, 3 (1998), 636–666.

[25] Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. 1957. *The measurement of meaning*. Number 47. University of Illinois press.

[26] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. Word2Sense : Sparse Interpretable Word Embeddings. In *ACL*.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[28] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *WWW*. 337–346.

[29] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *LREC*. ELRA, Valletta, Malta, 45–50.

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *SIGKDD*. 1135–1144.

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *SIGKDD*. 1135–1144.

[32] Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM* 8, 10 (1965), 627–633.

[33] Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. 2014. Unsupervised antonym-synonym discrimination in vector space. In *CLiC-it & EVALITA*. 328–333.

[34] Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Linked Data in Linguistics: Resources and Applications*. 64–69.

[35] Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. In *EACL*. 65–75.

[36] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*. 1631–1642.

[37] Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *AAAI*.

[38] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Sparse word embeddings using l 1 regularized online learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 2915–2921.

[39] Jesse Vig, Shilad Sen, and John Riedl. 2011. Navigating the tag genome. In *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, 93–102.

[40] Jesse Vig, Shilad Sen, and John Riedl. 2012. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2, 3 (2012), 13.

[41] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR. *Harv. JL & Tech.* 31 (2017), 841.

[42] Yajiong Xue, Huigang Liang, and Liansheng Wu. 2011. Punishment, justice, and compliance in mandatory IT settings. *Information Systems Research* 22, 2 (2011), 400–414.

[43] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *CVPR*. 4651–4659.

[44] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*. 1393–1398.