Training on Lexical Resources

Kenneth Church, Xingyu Cai, Yuchen Bian
Baidu, Sunnyvale, USA
{kennethchurch,xingyucai,yuchenbian}baidu.com

Abstract

We propose using lexical resources (thesaurus, VAD) to fine-tune pretrained deep nets such as BERT and ERNIE. Then at inference time, these nets can be used to distinguish synonyms from antonyms, as well as VAD distances. The inference method can be applied to words as well as texts such as multiword expressions (MWEs), out of vocabulary words (OOVs), morphological variants and more. Code and data are posted on `https://github.com/kwchurch/syn_ant`.

Keywords: Synonyms, Antonyms, Fallows, VAD (Valenace, Arousal, Dominance), multiword expressions

1. Introduction: Training on Lexicons

We do not normally think of lexical resources as training data, though others have trained on dictionaries (Brown et al., 1993; Chairatanakul et al., 2021). Suppose we have a thesaurus such as (Fallows, 1898).[1] Consider the thesaurus to be a set of triples: w_1, w_2, rel, where w_1 and w_2 are two words, and rel is 0 if w_1 and w_2 are synonyms and 1 if they are antonyms. We can then fine-tune a pretrained deep net such as BERT (Devlin et al., 2019) or ERNIE (Sun et al., 2020) using eq (1).

$$rel \sim w_1 + w_2 \tag{1}$$

The fine-tuning code is very simple. We modified an example from HuggingFace[2] in straightforward ways.[3] This code takes a pretrained net as input, and a set of triples, and outputs a fine-tuned net.

$text_1$	$text_2$	y_1	y_2
good	bad	-3.95	4.54
bad	evil	4.44	-5.00
good	benevolent	4.43	-5.05
bad	benevolent	-3.44	4.16
good	terrorist	-3.43	4.10
bad	terrorist	4.48	-5.10

Table 1: Inference: synonymy iff $y_1 > y_2$

Then there is another program for inference.[4] The inference program takes a fine-tuned net, and a pair of texts, and outputs two logits, y_1 and y_2, as illustrated in Table 1. The program predicts that the two input texts are synonymous iff $y_1 > y_2$.

In Table 1, the first two columns, $text_1$ and $text_2$, are single words, but the inference program accepts arbitrary texts as input (up to 512 subword units). Table 2 shows inference on multiword expressions (MWEs)[5] (Baldwin and Kim, 2010). It has been said that multiword expressions (MWEs) are a pain in the neck for natural language engineering (Sag et al., 2002), though MWEs are less of a pain in the neck for the proposed approach than for alternatives.

In other words, the inference program uses eq (2) to predict y from two input texts:

$$y \sim text_1 + text_2 \tag{2}$$

This notation is inspired by general linear models in R[6] (Guisan et al., 2002). We will start with binary classification (logistic regression). Later, classification will be replaced with regression when we consider VAD (Valance, Arousal and Dominance) distances in §5.

$text_1$	$text_2$	y_1	y_2
freedom fighter	good	2.33	-2.56
freedom fighter	bad	-1.50	2.19
white supremacist	good	-2.05	2.91
white supremacist	bad	1.67	-1.61

Table 2: Mutiword Expressions (MWEs)

The proposed approach generalizes naturally to address many other challenges in natural language processing such as OOVs (out of vocabulary words), negation, multiple languages, etc.

The remainder of this paper is organized as follows:

- Syn/Ant Binary Classification (§2)
- From Words to Texts (§3)
- Leakage with Standard Benchmarks (§4)
- VAD (Valance, Arousal, Dom) Regression (§5)

[1] `https://www.gutenberg.org/files/51155/51155-0.txt`

[2] `https://github.com/huggingface/accelerate/blob/main/examples/nlp_example.py`

[3] `https://github.com/kwchurch/syn_ant/blob/main/sentiment4.py`

[4] `https://github.com/kwchurch/syn_ant/blob/main/sentiment4_inference.py`

[5] `https://aclanthology.org/venues/mwe/`

[6] `https://www.r-project.org/`

Dataset	train	val	test
adj	5562	398	1986
noun	2836	206	1020
verb	2534	182	908
fallows	58,494	7190	7366
fallows-s	5886	753	777

Table 3: Sizes (edges) of synonym-antonym datasets

Test	Train				
	adj	noun	verb	fallows	fallows-s
adj	**0.886**	0.598	0.652	0.820	0.727
noun	0.662	**0.863**	0.685	0.706	0.638
verb	0.580	0.673	**0.899**	0.820	0.731
fallows	0.621	0.566	0.556	**0.663**	0.595
fallows-s	0.629	0.574	0.537	**0.660**	0.586

Table 4: Performance (accuracy) of Mixture of Experts (MoE) with default settings. (Chance is 0.5.)

Test	Train				
	adj	noun	verb	fallows	fallows-s
adj	**0.921**	0.859	0.852	0.897	0.868
noun	0.841	**0.917**	0.857	0.828	0.785
verb	0.813	0.829	**0.903**	0.851	0.794
fallow	0.633	0.604	0.620	**0.666**	0.634
fallow-s	**0.659**	0.602	0.591	**0.659**	0.627

Table 5: Accuracy of MoE with dLCE embeddings.

Test	Train			
	adj	noun	verb	fallows
adj	**0.908**	0.657	0.713	0.881
noun	0.773	**0.877**	0.792	0.797
verb	0.767	0.722	**0.906**	0.867
fallows	0.722	0.610	0.698	**0.947**

Table 6: Accuracy with fine-tuning.

2. Syn/Ant Binary Classification

2.1. Baseline: Mixture of Experts (MoE)

Consider the synonym-antonym task discussed in (Nguyen et al., 2017). The task is to input a pair of words and output a binary label: 0 (synonym) or 1 (antonym). We will evaluate on the datasets in Table 3. The first three datasets are borrowed from the supplemental materials of (Xie and Zeng, 2021).[7] Fallows is based on an online thesaurus (Fallows, 1898). Fallow-s (small) is a sample of Fallows. Our train/val/test splits are posted on github.[8]

Tables 4-5 show the results of the mixture of experts (MoE) method, proposed by (Xie and Zeng, 2021), using their code (see footnote 7). Table 4 uses default settings and Table 5 uses a new embedding, dLCE, from their github. They suggested that dLCE is better. Tables 4-5 provide additional evidence in favor of dLCE, though performance on fallows remains an opportunity for improvement.

They report just 3 of these 25 pairs:

1. Train on adj and test on adj,
2. Train on noun and test on noun, and
3. Train on verb and test on verb

We added the off-diagonal cases to see how well various methods generalize to mismatches between training sets and testing sets. Testing, of course, is performed on the test splits, and training is performed on the other splits.

As expected, mismatches often degrade accuracy. In general, there is a trade-off between quantity (size of training set) and quality (representativeness of training set to test set). It is sometimes said that there is no data like more data[9] (Pieraccini and Rabiner, 2012; Schmitt et al., 2021), but in this case, the relatively low scores in so many off-diagonal cells in Tables 4-5 suggest that quality often dominates quantity, at least for this task. There is an exception, of course, for fallows-s. When testing on fallows-s, it is better to train on fallows than fallows-s, since they are both sampling from the same population.

Ideally, we would like to be able to generalize from the training data to as many other cases as possible. From this perspective, the degradation in accuracy for the off-diagonal cases is disappointing. It is useful, nevertheless, to have an estimate of how much degradation is to be expected from mismatches in training and testing. Tables 4-5 provide estimates of the performance penalty for mismatches between training and test.

2.2. Proposed Method: Fine-Tuning

Table 6 uses the proposed method described in §1.[10] These results are competitive with results for MoE. Accuracy is higher on the diagonal; it is better to match testing and training conditions than not. That said, for practical applications, it would be convenient to train a system on whatever data is available at the time, even though users will run the system later on inputs that cannot always be anticipated in advance.

Differences between Tables 5-6 are shown in Table 7. Most differences are small with two exceptions highlighted in **red**. The proposed method is much better on fallows; MoE is slightly better on the datasets that it was developed on.

There are some special case rules in MoE for certain prefixes discussed in §3.2. While their ablation studies established that rules improve accuracy, such rules complicate the system and do not always generalize to new datasets such as fallows.

[7] https://aclanthology.org/2021.acl-short.71/

[8] https://github.com/kwchurch/syn_ant/tree/main/datasets/datasets_syn_ant

[9] http://www.lrec-conf.org/lrec2004/doc/jelinek.pdf

[10] The pretrained model was bert-based-uncased.

Test	Train			
	adj	noun	verb	fallows
adj	-0.013	**-0.202**	-0.139	-0.016
noun	-0.068	-0.040	-0.065	-0.031
verb	-0.046	-0.107	0.003	0.016
fallows	0.089	0.006	0.078	**0.281**

Table 7: Comparison of proposed method and MoE. Difference between two previous tables. MoE is better when difference is negative, and otherwise, proposed method is better. Large differences are shown in **red**.

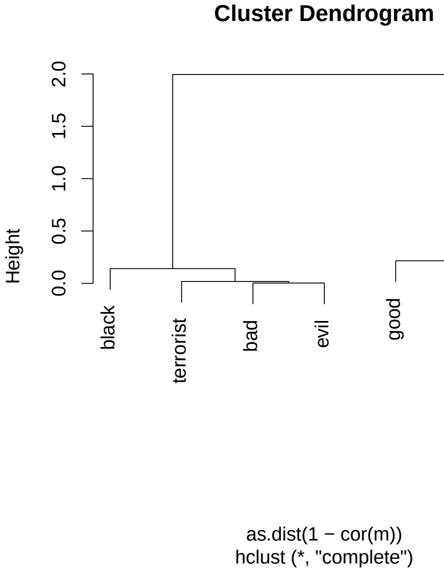

Figure 1: Clustering of correlations in Table 8 (bottom), illustrating biases in model.

3. From Words to Texts

As mentioned in §1, although the proposed method was trained on words, it can be applied to texts. This was useful for multiword expressions (MWEs) such as *freedom fighter*.

3.1. Undesirable Biases

Some generalizations are desirable and some are not. Biases are clearly problematic for embeddings and deep nets (Bolukbasi et al., 2016; Ali et al., 2019; Mehrabi et al., 2021). Figure 1 and Table 8 show that the proposed model is encoding some highly undesirable biases.

This example illustrates the use of a probing technique (Goodwin et al., 2020; Hewitt and Manning, 2019) for interpreting fine-tuned models. The probing technique starts with a set of words/phrases/texts. We run inference on all pairs of the inputs to produce logits, as shown on the top of Table 8. Figure 1 plots correlations of the logits as a dendrogram. In this case, there are two salient clusters separating words "like" *good* from words "like" *bad*.

	black	ter	bad	evil	good	ff	white
black	-3.493	-5.090	-5.05	-5.05	3.59	-3.117	-4.46
ter	-5.107	-4.635	-5.06	-5.07	4.27	-0.517	3.99
bad	-5.051	-5.105	-5.06	-5.00	4.02	1.804	3.74
evil	-5.008	-5.042	-4.99	-4.99	4.35	2.685	4.28
good	4.297	4.098	4.54	4.49	-5.04	-5.127	-5.12
ff	-1.512	0.687	2.19	3.30	-2.56	-2.713	-3.16
white	-0.612	4.313	4.20	4.51	-3.52	-5.122	-5.07

	black	ter	bad	evil	good	ff	white
black	1.000	0.884	0.885	0.859	-0.918	-0.805	-0.772
ter	0.884	1.000	0.992	0.982	-0.981	-0.813	-0.743
bad	0.885	0.992	1.000	0.997	-0.995	-0.799	-0.753
evil	0.859	0.982	0.997	1.000	-0.991	-0.786	-0.749
good	-0.918	-0.981	-0.995	-0.991	1.000	0.819	0.784
ff	-0.805	-0.813	-0.799	-0.786	0.819	1.000	0.938
white	-0.772	-0.743	-0.753	-0.749	0.784	0.938	1.000

Table 8: Biases in output model. Top (logits); Bottom (correlations of logits). Positive logits → antonyms. Headings are abbreviations for words in Figure 1.

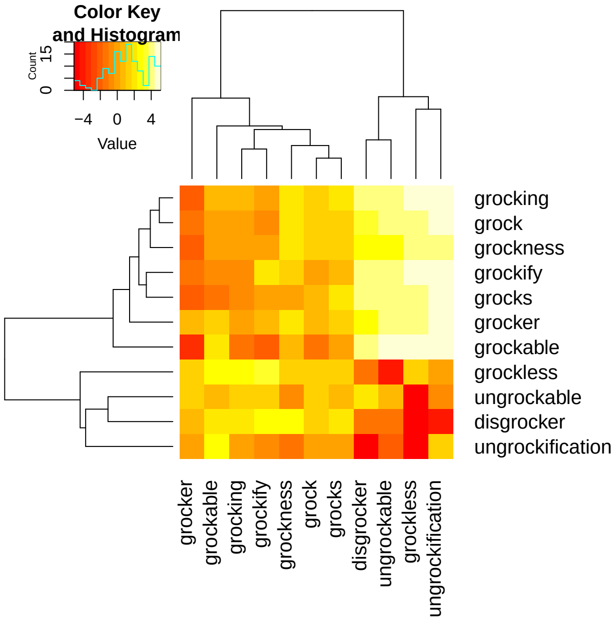

Figure 2: Heatmap of morphological variants of an out-of-vocabulary (OOV) word: *grock*.

3.2. Morphology, OOVs, MWEs, etc.

Figure 1 and Table 8 show how the proposed method can handle MWEs like *freedom fighter*. Out-of-vocabulary (OOV) words are another challenge for many methods. Figure 2[11] is similar to the previous tables and figures, but emphasizes OOVs and morphology. Note that the proposed method produces two distinct clusters: (1) positive: *grocking, grock, grockness, grockify, grocks, grocker, grockable*, and (2) negative: *grockless, ungrockable, disgrocker, ungrockification*. This example was inspired by a rule in MoE for certain

[11] The heatmap has slightly different rows and columns because predictions from the model need not be symmetric.

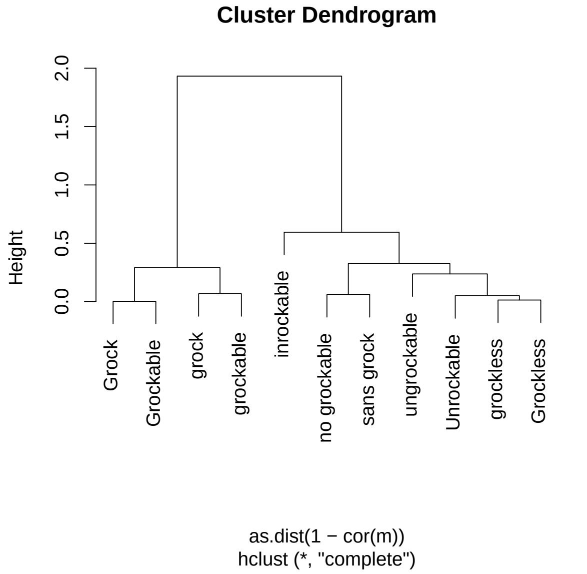

Figure 3: Clustering of morphological variants and translations of an out-of-vocabulary (OOV) word: *grock*. Base model: bert-base-multilingual-cased.

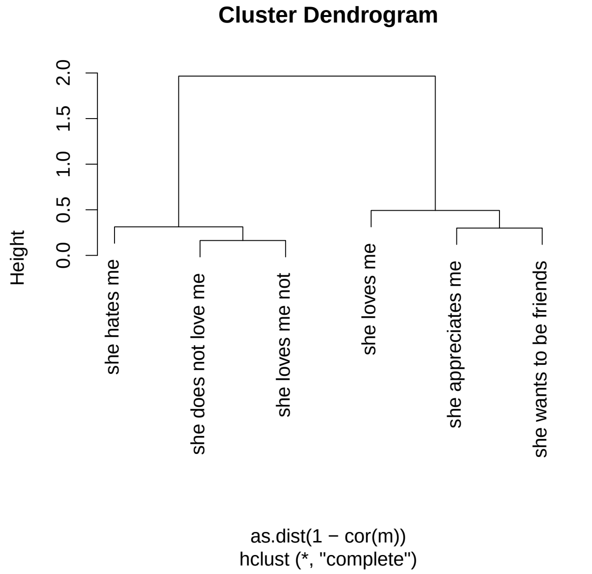

Figure 4: Clustering of correlations of logits of all pairs of six sentences.

prefixes: *de, a, un, non, in, ir, anti, il, dis, counter, im, an, sub, ab*. They attribute this rule to (Rajana et al., 2017) and (Ali et al., 2019). While such rules do well in ablation studies, they complicate the system and may not generalize well to unanticipated cases. The rule for *a-,* for example, degrades accuracy on fallows. Rather than stipulate such rules, we prefer to use such observations in probing tasks to interpret deep nets (so-called Bertology (Rogers et al., 2020)).

Examples such as Figure 2-3 suggest that the model is remarkably successful in capturing some of these more salient morphological relations including both prefixes and suffixes. Both figures cluster variants of the OOV, *grock*. Figure 3 uses Google Translate to add some additional variants from other languages. The clusters separate the positive terms from the negative ones, as we hoped they would.

These anecdotal examples are far from definitive evidence, but they suggest that the proposed approach offers a promising way forward on some challenging issues. Morphology is, of course, a huge topic.[12] Some recent papers on morphology and OOVs include: (Hofmann et al., 2021; Hofmann et al., 2020; Haley, 2020).

3.3. Negation

Many other generalizations are possible. The example in Figure 4 is similar to previous examples, but this example clusters sentences as opposed to words. These sentences involve various negations that are somewhat similar to synonyms and antonyms. It appears that fine-tuning for the synonym/antonym task may be transferring some learnings from lexical semantics that may be useful for negation in sentences.

Transfer learning has been so successful on so many tasks that it is very natural to suggest its applicability to negation (Kassner and Schütze, 2020; Khandelwal and Sawant, 2020). It ought to be possible to use lexical resources such as (Fallows, 1898) to learn facts about negation that extend well beyond lexical semantics.

4. Paths and Leaks from Train to Test

It turns out that there is substantial leakage in the datasets in §2. This leakage casts serious doubt on the results reported in Tables 4-7, as well as (Nguyen et al., 2017; Xie and Zeng, 2021). This leakage is somewhat similar to leakage that we have discovered in WN18RR, a popular benchmark based on WordNet for research on knowledge graph completion (KGC) (Church and Bian, 2021).

The clusters in the previous section suggest that there is considerable structure among synonyms and antonyms. From a theoretical point of view, one might expect synonyms and antonyms to be symmetric. Moreover, it is natural to assume transitivity. Of course, there are many exceptions in practice, but there are clearly larger structures that could leak information between splits in experiments like those reported above (Figures 4-6).

The splits can be viewed as sparse graphs, as shown in Table 9. SimLex-999 (Hill et al., 2015) and NRC-VAD are shown for comparison. NRC-VAD will be discussed in §5.

Fallows-s is a random sample of the edges in fallows. Note that fallows-s has many more connected components than fallows. More generally, the standard practice of randomly assigning edges to train, val and test splits will cut connected components. Parts of a component will end up in one split, and the rest will end up

[12] `https://sigmorphon.github.io/`

training set	V	E	CC
adj	3315	5562	285
noun	3654	2836	1204
verb	1859	2534	199
fallows	15,466	58,494	32
fallows-s	6326	5886	907
SimLex	1028	999	151
NRC-VAD	20,007	$20,007^2$	1

Table 9: Most graphs are sparse, $E \ll V^2$, except NRC-VAD. V (vertices), E (edges) and CC (connected components) are computed over training sets.

Path Length	**adj**	**noun**	**verb**	**fallows**
0				2
1	99	59	60	946
2	80	7	15	3835
3	59	3	7	1156
4+	70	2	35	639
NA	90	135	65	612
total	398	206	182	7190

Table 10: For most pairs of words in the validation set, w_1 and w_2, there is a short path from w_1 to w_2 based on edges in the training set.

in other splits. There is a risk that information could leak from one split to another if there are clues left behind suggesting how to reconstruct components.

Path lengths appear to be a useful clue for reconstructing components, as suggest in Table 10. Consider the 398 edges, $E = (w_1, w_2)$, in the validation set for adj. Table 10 reports that that 99 of these 398 edges have a path of length 1 using edges from the training set. There are another 80 of 398 with a path of length 2. All but 90 of 398 are part of a connected component in the training set. When an edge in one split is part of a connected component in another split, it is likely that the label on the edge can be inferred from the labels associated with the component in the other split. In this way, it is likely that information is leaking across splits, when edges are randomly assigned to splits in the standard way.

Consider the 99 edges of length 1. These are particularly worrisome. There are 99 pairs like *good* and *awful*, where the same edge is in both train and validation, but in different directions. This pair is clearly leaking information between the training and validation splits.

The path lengths in Table 10 were computed using a shortest path tool. The tool provides options for directed and undirected graphs. The undirected option was used to find pairs such as *good* and *awful*.

Edges of length 2 are not leaking as badly as edges of length 1, but we are concerned about them. Some examples from adj of length 2 paths are: *innocent* → *harmless* (via *harmful*), *fresh* → *old* (via *aged*), *dead* → *deceased* (via *alive*).

How can we use these paths to leak labels across splits?

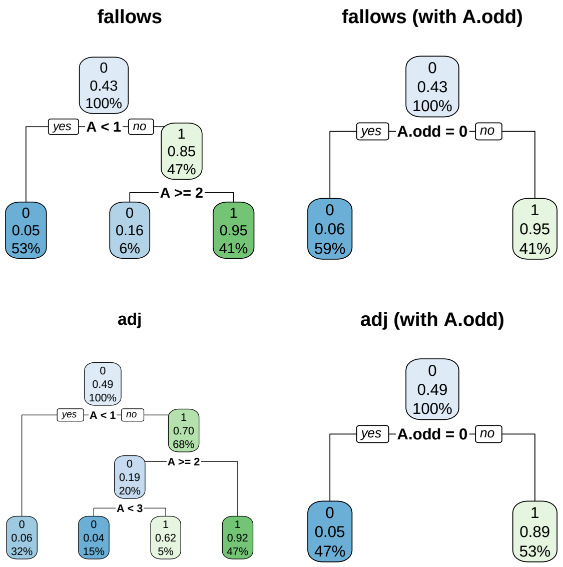

Figure 5: *A-leakage*: Decision trees learn to classify pairs as antonyms iff A is odd.

	val		test	
	acc	applicable	acc	applicable
adj	0.916	308/398	0.906	1482/1986
noun	0.930	72/206	0.983	302/1020
verb	0.872	118/182	0.882	587/908
fallows	0.945	6576/7190	0.949	6722/7366
fallows-s	0.683	223/753	0.694	241/777

Table 11: A-leakage: $Pr(ant) > Pr(syn)$ iff A is odd. Accuracy is computed over applicable edges. Denominators are borrowed from Table 3.

Let A be the number of antonym labels on a path. The decision trees in Figure 5 suggest that an edge should be labeled as an antonym iff A is odd. We will refer to this heuristic as A-*leakage*. Table 11 shows substantial A-leakage.

There are 4 trees in Figure 5. The two trees on the left fit: $gold \sim A$ for two datasets: fallows and adj. Based on these two trees, we obtained the simpler trees on the right by fitting: $gold \sim A + A.odd$. Decision trees learn to ignore A.

These trees were created with rpart.[13] There are three numbers associated with each subtree: a label (1/0), $Pr(1)$, coverage. By construction, at each level in the tree, the coverage sums to 1.

In short, the trees in Figure 5 make it clear that there is substantial leakage in these datasets. Work based on these resources may need to be retracted.

[13] `https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/rpart`

word	Val	Arousal	Dom	Dist
open	0.620	0.480	0.569	0.00
unfold	0.612	0.510	0.520	0.06
reopen	0.656	0.528	0.568	0.06
close	0.292	0.260	0.263	0.50
closed	0.240	0.164	0.318	0.55
undecided	0.286	0.433	0.127	0.56

Table 12: Words above the double line are near *open*. The last column is the Euclidean distance to *open*.

	Antonyms				Sim
	adj	noun	verb	fallows	SimLex
cor	0.55	0.48	0.44	0.52	-0.40

Table 13: VAD distances are positively correlated with antonyms, and negatively correlated with SimLex similarities, though none of these correlations are large.

5. VAD: Valence, Arousal & Dominance

Given the observations about leakage in the previous section, we became concerned about the syn/ant classification task. To address these concerns, we introduce a new task which we call *VAD regression*. Some examples of NRC-VAD[14] (Mohammad, 2018) scores are shown in Table 12.[15] NRC-VAD lists 20k lemmas with three scores between 0 and 1 for V (Valence), A (Arousal) and D (Dominance).

There is a considerable literature on VAD norms in psychology (Osgood et al., 1957). VAD has some similarities with synonyms and antonyms, but there are some important differences. While many synonyms are near one another in VAD space, and most antonyms are far from one another, there are many exceptions, as indicated by the modest correlations in Table 13.[16]

We create train, val and test data by selecting pairs of words, w_1 and w_2. Each word is assigned $VAD(w)$, a point in $\mathbb{R}^3$. y are distances between pairs of words. That is, $y(w_1, w_2) = |\text{VAD}(w_1) - \text{VAD}(w_2)|$.

VAD regression is similar to syn/ant classification. Datasets consist of three splits: train, val, test. Each split contains pairs of words and labels. Eq (1) uses training and validation splits to learn a model that takes w_1 and w_2 as input and predicts $\hat{y}$. Evaluations report loss between y and $\hat{y}$ on the test split.

There are a few differences between regression and classification.[17] Losses are reported in terms of mean squared error (and R2),[18] as opposed to classification accuracy.[19] Labels are also different; $y \in \mathbb{R}$ for regression, as opposed to $y \in \{0, 1\}$ for binary classification. In addition to concerns about leaks, another motivation for moving from syn/ant classification to VAD regression is scale. VAD has $20,000^2$ edges, many more than alternatives in Table 9. The large number of edges makes it possible to study different sampling methods. We looked at losses over a number of variables: epochs, base model, sample size, vocabulary. We had more success for larger splits than smaller splits.

Splits over vertices (V) are very different from splits over edges (E). Note that while NRC-VAD is larger than many alternatives, $V = 20k$ is too small to cover many corpora and therefore, we need to find ways to generalize to unseen words. Unfortunately, the proposed method works better when testing and training splits use the same words than when the test split contains unseen words that do not appear in the other splits.

5.1. Morpheme Diagnostic

VAD can also be viewed as an embedding, though in three dimensions, as opposed to static and contextual embeddings that typically make use of hundreds of dimensions. Figure 6 introduces a novel morpheme diagnostic to compare VAD distances to WNews300[20] and GNews300.[21] The diagnostic constructs 16 lists of word pairs based on 6 prefixes[22] and 10 suffixes.[23] The list for *over-*, for example, includes pairs of words that differ by that prefix, e.g., *overlook/look*, *overtake/take*. The plots summarize vector distances for each list. **Red** lines are provided for reference at $y = 0$ (most similar) and $y = \sqrt{2}$ (high reference).[24] The morpheme diagnostic favors VAD embeddings over WNews and GNews because affixes toward the right should have relatively large distances. That is, pairs such as *fear/fearless* have large VAD distances, like most antonyms. Conversely, affixes toward the left should have relatively small distances. That is, pairs such as *fear/fears* have relatively small VAD distances.

5.2. Related Work

It is well-known that standard embeddings do not separate synonyms from antonyms very well (Nguyen et al., 2016). (Faruqui et al., 2015) proposed retrofitting to address this issue. Our proposal of fine-tuning on VAD distances can be viewed as a form of retrofitting.

[14] https://saifmohammad.com/WebPages/nrc-vad.html

[15] See http://crr.ugent.be/archives/1003 (Warriner et al., 2013) for more VAD norms.

[16] One set of exceptions are taboo words such as the seven words you cannot say on television. As (Pinker, 2007) points out, euphemisms are different from synonyms. The f-word is more about shock than sex. These words are far apart in VAD space: |VAD(f-word) − VAD(sex)| = 0.66.

[17] https://github.com/kwchurch/syn_ant/blob/main/fine_tune_VAD_pairs.py

[18] sklearn.metric.r2_score

[19] Empirically, mean square error and R2 have a correlation near -1, and therefore, they are about equally informative. Much of the discussion below, though, will focus on R2 because it is easier to interpret. Ideally, R2 values should be near 1. Values will be near 0 (or even negative) when predictions are uninformative.

[20] https://fasttext.cc/docs/en/english-vectors.html

[21] https://code.google.com/archive/p/word2vec/

[22] *re-, pro-, under-, over-, dis-, un-*

[23] *-s, -ism, -ly, -ment, -ed, -ness, -ing, -ite, -able, -less*

[24] $|a - b| \approx \sqrt{2}$ if a and b are random vectors of unit length.

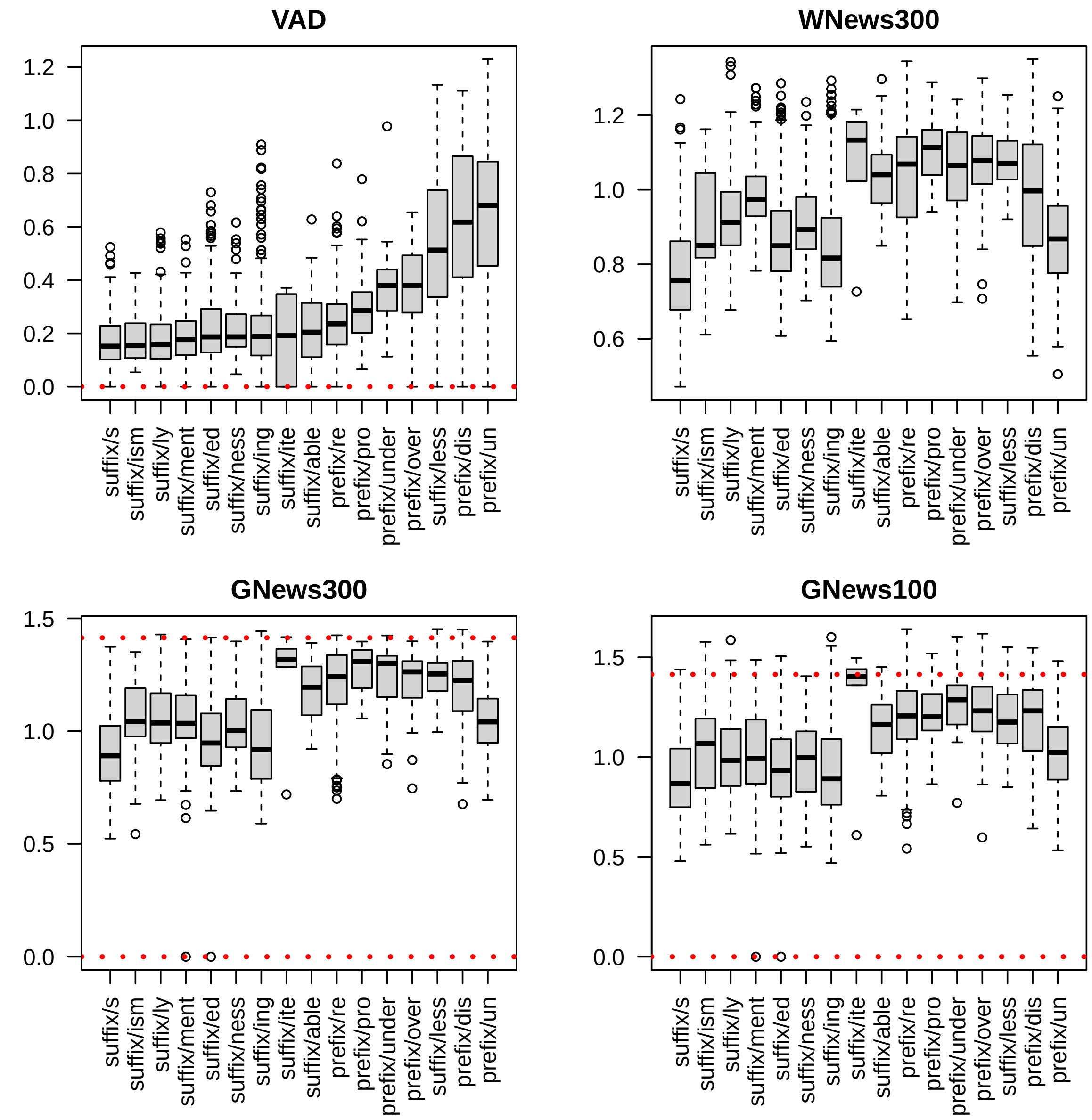

Figure 6: Morpheme diagnostic favors VAD over alternative embeddings because affixes on the right should have relatively large distances. That is, pairs such as *fear/fears* are similar to one another (small VAD distance of 0.03), unlike *fear/fearless* (large VAD distance of 0.87). For other embeddings, most pairs have large distances, even morphologically related pairs like *fear/fears*.

Figure 6 suggests this kind of retrofitting could also be useful for capturing certain morphological regularities. (Mohammad, 2020; Mohammad, 2016) surveys work on VAD and similar topics such as sentiment classification (Pang et al., 2002; Turney, 2002; Liu, 2012) and emotion classification (Park et al., 2019). This survey mentions tasks at nearly every Semeval meeting since 2013. Challenges with antonyms are discussed in §5.2 (Cruse et al., 1986; Justeson and Katz, 1991; Fellbaum, 1995; Mohammad et al., 2008).

5.3. Results and Discussion

As mentioned above, the regression task is to input two texts and output $\hat{y}$, an estimate of the VAD distance between these two texts. The test, validation and training splits are based on a vocabulary of $V = 16k$ of the $20k$ entries in NRC-VAD. The remaining entries were held back so we could evaluate how well the proposed method generalizes to words that did not appear in the three splits.

Table 14[25] shows a case where fine-tuning transfers well across splits. In this case, the splits are large and similar to one another. All three sets sample edges be-

[25] In Table 14, BERTc and BERTun refer to bert-base-cased and bert-base-uncased. Similarly, SciBERTc and SciBERTun refer to SciBERT with and without case. BERTmulti refers to bert-base-multilingual-cased. ERNIE refers to ernie-2.0-en.

base model	train	val	test	unseen
BERTun	0.993	0.993	0.993	0.092
SciBERTun	0.993	0.993	0.992	0.095
ERNIE	0.991	0.990	0.990	0.075
SciBERTc	0.988	0.988	0.987	0.110
BERTmulti	0.988	0.987	0.991	0.133
BERTc	0.995	0.995	0.988	0.062

Table 14: R2 scores are near 1.0 when the training set is large (1M edges), and the splits are sampled from the same distribution. R2 is worse on unseen words.

tween the same $V = 16k$ words. The sets are disjoint but otherwise representative of one another. The training set contains 1M edges, and the validation and test sets contain 100k edges.

On the other hand, if the sets are too small or too different from one another, then fine-tuning does not transfer well. That is, fine-tuning improves R2 on the training set, but not on the other splits. We experimented with training sets of 10k, 100k and 1M edges. With 1M edges, fine-tuning almost always transferred well. On the other hand, with 10k, fine-tuning rarely transfers well. With 100k edges, fine-tuning produced large improvements in R2 on the training set, but the improvements on the other splits are relatively modest.

Unfortunately, even our best models do not generalize well to unseen words, as shown in Table 14. The unseen column in Table 14 is a test set sampled over words in in NRC-VAD but not in the 16k vocabulary used for fine-tuning.

Figure 7 uses the morpheme diagnostic to compare two sampling methods. The top panel uses 1M sampling (as in Figure 14). The bottom panel uses ABCD sampling, where the VAD vocabulary is partitioned into 4 sets: A, B, C, D. The training set samples edges from A to B, the validation set samples edges from A to C, and the test set samples edges from A to D. Unfortunately, ABCD sampling does not transfer well, and does not pass the morpheme diagnostic.[26]

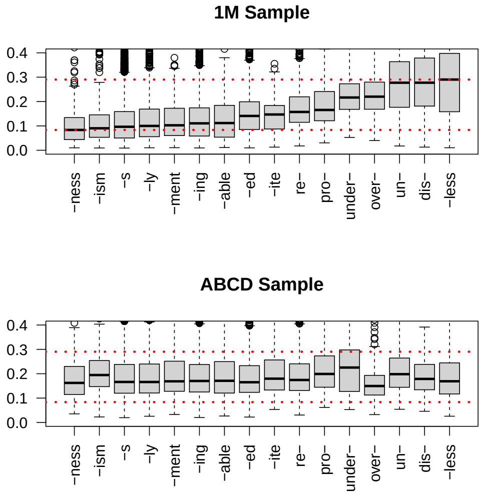

Figure 7: ABCD sampling does not pass the morpheme diagnostic. **Red** lines are provided for reference.

6. Conclusions

This paper proposed fine-tuning pretrained deep nets such as BERT and ERNIE on lexical resources such as thesauri and VAD lexicons. We do not normally think of lexical resources as training data, but maybe we should.

We proposed a simple fine-tuning framework for syn/ant classification and VAD regression. Both methods fit: $y \sim text_1 + text_2$. For syn/ant classification, $y \in \{0, 1\}$ and for VAD regression, $y \in \mathbb{R}$. We compared the proposed method to MoE (Xie and Zeng, 2021). The proposed method is competitive on standard benchmarks, and considerably better on a new benchmark based on (Fallows, 1898). The proposed method can be applied at inference time to novel words, as well as MWEs, OOVs and longer texts in multiple languages.

On a cautionary note, we found evidence of leakage across splits in standard benchmarks as well as the proposed benchmark based on (Fallows, 1898). Work based on these resources may need to be retracted.

To address these concerns with leakage, we introduced a new task, VAD regression. Since the VAD graph is fully connected, we could study different methods for creating train/val/test splits.

Much of the rest of this paper is concerned with how well the proposed method transfers to unseen words and unanticipated cases. We found that transfer learning depends on the size and representativeness of the splits. If the splits are sufficiently large and representative of one another, then it is likely that fine-tuning on the training set will not only improve the loss on the training set, but also on the other splits as well. When fine-tuning improves validation loss, results often generalize well to unseen edges, though less well to unseen nodes. That is, the proposed method appears to be effective for predicting labels for pairs of words in the training set, but less effective for pairs of unseen words.

7. References

Ali, M. A., Sun, Y., Zhou, X., Wang, W., and Zhao, X. (2019). Antonym-synonym classification based on new sub-space embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6204–6211, Jul.

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. *Handbook of natural language processing*, 2:267–292.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing

[26] In Figure 7, lists of word pairs are based on the MUSE (Conneau et al., 2017) vocabulary, which is larger than VAD.

word embeddings. *Advances in neural information processing systems*, 29:4349–4357.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Goldsmith, M. J., Hajic, J., Mercer, R. L., and Mohanty, S. (1993). But dictionaries are data too. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Chairatanakul, N., Sriwatanasakdi, N., Charoenphakdee, N., Liu, X., and Murata, T. (2021). Cross-lingual transfer for text classification with dictionary-based heterogeneous graph. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1504–1517, Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Church, K. and Bian, Y. (2021). Data collection vs. knowledge graph completion: What is needed to improve coverage? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6210–6215, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Cruse, D. A., Cruse, D. A., Cruse, D. A., and Cruse, D. A. (1986). *Lexical semantics*. Cambridge university press.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Fallows, S. (1898). *A Complete Dictionary of Synonyms and Antonyms or, Synonyms and Words of Opposite Meaning*. Good Press.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June. Association for Computational Linguistics.

Fellbaum, C. (1995). Co-occurrence and antonymy. *International journal of lexicography*, 8(4):281–303.

Goodwin, E., Sinha, K., and O'Donnell, T. J. (2020). Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online, July. Association for Computational Linguistics.

Guisan, A., Edwards Jr, T. C., and Hastie, T. (2002). Generalized linear and generalized additive models in studies of species distributions: setting the scene. *Ecological modelling*, 157(2-3):89–100.

Haley, C. (2020). This is a BERT. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341, Online, November. Association for Computational Linguistics.

Hewitt, J. and Manning, C. D. (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, June. Association for Computational Linguistics.

Hill, F., Reichart, R., and Korhonen, A. (2015). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, December.

Hofmann, V., Pierrehumbert, J., and Schütze, H. (2020). DagoBERT: Generating derivational morphology with a pretrained language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861, Online, November. Association for Computational Linguistics.

Hofmann, V., Pierrehumbert, J., and Schütze, H. (2021). Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. 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 3594–3608, Online, August. Association for Computational Linguistics.

Justeson, J. S. and Katz, S. M. (1991). Co-occurrences of antonymous adjectives and their contexts. *Computational Linguistics*, 17(1):1–20.

Kassner, N. and Schütze, H. (2020). Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online, July. Association for Computational Linguistics.

Khandelwal, A. and Sawant, S. (2020). NegBERT: A transfer learning approach for negation detection and scope resolution. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5739–5748, Marseille, France, May. European Language Resources Association.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), July.

Mohammad, S., Dorr, B., and Hirst, G. (2008). Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, Honolulu, Hawaii, October. Association for Computational Linguistics.

Mohammad, S. M. (2016). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, pages 201–237. Elsevier.

Mohammad, S. (2018). Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia, July. Association for Computational Linguistics.

Mohammad, S. M. (2020). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *CoRR*, abs/2005.11882.

Nguyen, K. A., Schulte im Walde, S., and Vu, N. T. (2016). Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany, August. Association for Computational Linguistics.

Nguyen, K. A., Schulte im Walde, S., and Vu, N. T. (2017). Distinguishing antonyms and synonyms in a pattern-based neural network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 76–85, Valencia, Spain, April. Association for Computational Linguistics.

Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1957). *The measurement of meaning*. Number 47. University of Illinois press.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics, July.

Park, S., Kim, J., Jeon, J., Park, H., and Oh, A. (2019). Toward dimensional emotion detection from categorical emotion annotations. *arXiv preprint arXiv:1911.02499*.

Pieraccini, R. and Rabiner, L., (2012). *There Is No Data like More Data*, pages 135–166. MIT Press.

Pinker, S. (2007). *The stuff of thought: Language as a window into human nature*. Penguin.

Rajana, S., Callison-Burch, C., Apidianaki, M., and Shwartz, V. (2017). Learning antonyms with paraphrases and a morphology-aware neural network. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 12–21, Vancouver, Canada, August. Association for Computational Linguistics.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for nlp. In *International conference on intelligent text processing and computational linguistics*, pages 1–15. Springer.

Schmitt, M., Ahmadi, S. A., and Hänsch, R. (2021). There is no data like more data – current status of machine learning datasets in remote sensing.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H. (2020). Ernie 2.0: A continual pre-training framework for language understanding. *AAAI*.

Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Warriner, A. B., Kuperman, V., and Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Xie, Z. and Zeng, N. (2021). A mixture-of-experts model for antonym-synonym discrimination. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 558–564, Online, August. Association for Computational Linguistics.