Reducing Embedding Dimensionality for Classification with SHAP

Parker Glenn
Brandeis University
parkerglenn@brandeis.edu

Abstract

Pre-trained word embeddings have become ubiquitous across the downstream tasks of text similarity and classification. While much has been done on strengthening pre-trained performance for baseline similarity tasks through variance based post-processing (Mu and Viswanath, 2018) and dimensionality reduction (Raunak et al., 2019), little work has been done on improving performance on baseline classification tasks from an embedding level. This paper explores one possible alternative to the previously proposed variance-based methods, utilizing model explainability techniques to achieve effective and task-specific dimensionality reduction on dense word embeddings in a classification context.

1 Introduction

1.1 Main Contributions

The paper's main contributions are the following:

1. This paper offers an alternative to variance-based solutions for reducing word embedding dimensionality in the context of classification tasks which minimizes accuracy losses and, in many instances, even improves scores.

2. A new method for interpreting an ngram's influence on predicted classification labels given its dense embedding is introduced in the form of the $SubspaceScore$, modified from Jang et al. (2019).

3. The failure of variance-based methods for word embeddings in the context of classification is highlighted, and it is shown that random subsets of dimensions outperform their PCA counterparts on a variety of downstream classification tasks.

1.2 Relevant Past Work

Mu and Viswanath (2018), showed that through simple post-processing, off-the-shelf word representations could be purified, allowing them to perform better on downstream similarity tasks. Finding that word representations have a non-zero mean and share a large common vector, they subtract the mean. Then, citing Bullinaria and Levy in saying that the highest variance components of word embeddings are often corrupted by information other than lexical semantics, they remove the projection of the top 7 principle components. As a result of creating these purified representations, the embeddings are made to be more discriminative and perform better than off-the-shelf embeddings on similarity benchmarks. Raunak et al. (2019) went on to show that multiple levels of this post-processing could yield further performance gains. By halving the size of 300 dimensional embeddings with PCA in between two levels of Mu's post-processing algorithm, similar or better accuracy was achieved on similarity benchmarks when compared to the original Glove embeddings.

While the scores across the similarity benchmarks are promising, the processed embeddings fare worse than their off-the-shelf counterparts across most classification benchmarks. Table 1 shows the poor performance of Raunak's algorithm on classification tasks when compared to the original 300D Glove embeddings. Raunak et al. (2020) went on to further examine the utility of variance in the context of classification and machine translation, and the conclusion is made that syntactic information captured by a prinicple component is independent of the amount of variance it explains.

1.3 Shapley Values

Shapley Values were introduced in 1951 by mathematician and economist Lloyd Shapley. Rooted

Model	MR	CR	SUBJ	MPQA	SST2	SST5	TREC
Glove-300D	75.59	78.31	91.58	86.88	78.03	41	68
Algo-50D	66.52	70.49	85.6	77.5	68.92	35.48	50
Algo-100D	70.43	75.34	88.31	82.3	71.99	38.42	55
Algo-150D	73.45	77.43	89.86	85.59	76.33	40.18	59.6
Algo-200D	75.23	78.17	90.61	86.51	78.09	41.36	65.4

Table 1: Results from Raunak et al. (2019). *Algo* refers to their dimensionality reduction algorithm.

in game theory, the concept was developed in trying to assign payments to members in a society proportional to their marginal contributions.

The Shapley value is defined in the figure below.

$$\phi_i(N,v) = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N|-|S|-1)!}{|N|!}[v(S \cup \{i\}) - v(S)] \tag{1}$$

Where N is the set of all players, $S \subseteq N$ is the set of all player subsets, and $v(S \cup \{i\})$ is the value of a coalition with the agent i. Essentially, it calculates the marginal contribution of player i given the output of all possible formulations of subsets disregarding i.

It's important to note that the Shapley value for an agent i is not the change in output when agent i is removed from the coalition. Instead, the Shapley value for i is the value of contribution by agent i in a particular instance compared to the average contribution.

1.4 SHAP

Interestingly, the Shapley value has been extended into the field of machine learning by exchanging the concept of "players" for "features". However, the largest roadblock in effectively transferring the Shapley value to machine learning is its massive computational requirements. For a model with k features, there are 2^k possible coalitions. As result, different methods of estimation have been pursued. One recent and effective solution to this problem of estimation is SHAP (Lundberg and Lee, 2017).

SHAP Value for Logistic Regression Throughout the course of these experiments, the SHAP values from a simple logistic regression model are extracted. Assuming feature independence, much of the more complex estimation techniques pioneered by SHAP can be removed, as the foundational summation within logistic regression can be used to efficiently estimate Shapley values. Additionally, because we assume the dimensions of the embeddings to be independent, the ordering of feature addition does not affect the model output.

The output of logistic regression can be represented as the following function.

$$\hat{y} = \sum_{i=1}^{m} w_i x_i = W^T X \tag{2}$$

Where m is the number of training samples. Since this summation is a the core of logistic regression, we can consider it a linear model and extract Shapley values directly from the weight coefficients.

The SHAP value for a single feature in logistic regression is defined as follows.

$$\phi_i = w_i \cdot (x_i - E[x_i]) \tag{3}$$

where w_i is the coefficient for feature i, and $E[x_i]$ is the expected value of feature i. Here, "expected value" is just the average value of feature i. This formula was also noted in Strumbelj and Kononenko (2010).

2 Methodology

2.1 Downstream Tasks

The following downstream classification tasks were implemented, with their associated datasets from from SentEval (Conneau and Kiela, 2018).[1]

MR A binary sentiment classification dataset of movie reviews with 8.5k train and 2.1k test samples.

CR A binary sentiment classification dataset of customer reviews with 3k train and 750 test samples.

SUBJ A binary subjectivity status dataset of subjective reviews and objective plot summaries for movies, with 8k train and 2k test samples.

MPQA A binary sentiment classification dataset of short text snippets (3 words) from news articles, with 8.5k train and 2k test samples.

SST2 A binary sentiment classification dataset with 67k train and 1.8k test samples.

SST5 A 5-way multiclass sentiment classification dataset, with 8.5k train and 2.2k test samples.

[1]https://github.com/facebookresearch/SentEval

TREC A 6-way mutliclass question-type classification dataset, with 4.3k train and 1k test samples.

2.2 PPA and Algo

To attempt to provide an alternative to variance-based solutions for embedding reduction in the context of classification tasks, SHAP is used in place of PCA for dimensionality reduction. To frame the SHAP algorithms within existing baselines and examine the effect of variance-based processing on classification performance, algorithms 1 and 2 are built upon, later referred to as Post Processing Algorithm (PPA) and Algo respectively.

Algorithm 1 Post Processing Algorithm (PPA)

```
1: D: Threshold parameter
2: X: Word embedding matrix
3: procedure PPA($X, D$)
       /*Subtract Mean Embedding */
4:     $X = X - \overline{X}$;
       /*Compute PCA Components */
5:     $u_i$ = PCA($X$), where i = 1,2....d;
       /*Remove Top-D Components */
6:     for all $v$ in X do
7:         $v = v - \sum_{i=1}^{D}(u_i^T \cdot v)u_i$
8:     end for
9: end procedure
```

From (Mu and Viswanath, 2018)

Algorithm 2 Algo-N Algorithm

```
1: D: Threshold parameter
2: N: New dimensionality
3: X: Word embedding matrix
4: procedure ALGO($X, D$)
       /*Apply Algorithm 1 (PPA) */
5:     $X$ = PPA($X$, $D$)
       /*Reduce to D with PCA */
6:     $X$ = PCA($X$, $N$)
       /*Re-apply Algorithm 1 (PPA) */
7:     $X$ = PPA($X$, $D$)
8: end procedure
```

From (Raunak et al., 2019)

2.3 Proposed Algorithm

The proposed algorithm is a slight modification to the Algo-N algorithm, using SHAP to reduce X to N dimensions instead of PCA. Since a SHAP-based dimensionality reduction would have knowledge of the target classification domain, the intuition is that taking those dimensions with the highest marginal contributions to $\hat{y}$ in the training set of data would minimize any decline in accuracy that may be seen with fewer dimensions. As is seen later, this approach actually improved accuracy scores in many evaluations.

Algorithm 3 SHAP-Algo Algorithm

```
1: D: Threshold parameter
2: N: New dimensionality
3: X: Word embedding matrix
4: procedure SHAP-ALGO($X, D$)
       /*Apply Algorithm 1 (PPA) */
5:     $X$ = PPA($X$, $D$)
       /*Reduce to D with SHAP */
6:     $X$ = SHAP($X$, $N$)
       /*Re-apply Algorithm 1 (PPA) */
7:     $X$ = PPA($X$, $D$)
8: end procedure
```

Identifying Subspaces with SHAP Running a LinearExplainer on a trained binary classification-trained logistic regression model returns a matrix of shape $(\#samples, \#features)$. The values in the $\#features$ column correspond to the SHAP value for that particular feature, given a sample, in the logs-odd space. A higher SHAP value for a particular dimension raises the chance of the sample being classified with a 1 label, and a lower SHAP value raises the chance of being classified with a 0 label. For a multiclass classification problem, the SHAP matrix is of the shape $(\#samples, \#classes, \#features)$, where the values in $\#features$ correspond to the feature's marginal contribution for that particular class.

There are two possible interpretations of "top dimensions", given our SHAP matrix. The first would involve taking a more class-specific approach, and finding the k most influential subspaces across each class prediction, so that $k \times |Y| = N$, where $|Y|$ is the size of all unique class labels.

The second method would be to find the N most influential dimensions regardless of class distribution. Here, rather than going class by class, we would take the absolute values of all Shapley values and then calculate each dimension's mean across training samples. After sorting the dimensions by their Shapley value mean, we would take the N dimensions with the highest scores. It is vital that we first take the absolute values of the Shapley values, since as shown in the description of the SHAP matrix for binary classification, an extremely negative Shapley value corresponds to that feature being very impactful in classifying the sample as class 0. Similarly for a multi-class classification SHAP matrix, a negative Shapley value for a class is meaningful, since it can be viewed as pushing $\hat{y}$ away from that particular class.

Ultimately, the second method was used and specific class distribution was ignored. From running both methods, the resulting dimensions seemed to be largely identical. The main reason for choosing the second was mostly due to convenience: when dealing with a multi-class classification problem, a SHAP matrix of shape $(\#samples, \#classes, \#features)$ is produced, and often $\#classes$ doesn't divide perfectly into N.

Task	Top-5 SHAP Dimensions
MR	**32**, **220**, 110, 164, 200
CR	9, 46, 48, **32**, 278
SUBJ	**276**, 106, 186, 172, 254
MPQA	271, **220**, 174, **34**, **244**
SST2	**32**, 246, 112, 243, 21
SST5	**244**, 266, **276**, **34**, 288
TREC	296, 155, 181, 298, 96

Table 2: Top SHAP dimensions across classification tasks. Bold represents a dimensions appearing in at least two tasks.

Table 2 shows the top 5 dimensions identified by SHAP across the classification tasks. As would be expected, there is a decent amount of overlap between influential dimensions amongst all the sentiment classification tasks. Notably, dimension 32 appears in the top-5 for the MR, CR, and SST2 binary sentiment classification tasks, and is the single most influential dimension for the MR and SST2 tasks.

Variations on the Proposed Algorithm To gain more insight into how variance-based reduction methods interact with different classification tasks, the following variations of the SHAP-Algo algorithm were tested.

- **SHAP-PPA**: Algorithm 1, followed by SHAP-based dimensionality reduction. No re-application of Algorithm 1.
- **SHAP** No removal of the top principle components, just SHAP-based dimensionality reduction on the original embeddings.

2.4 Experimental Environment

For testing the quality of the embeddings, the sentence evaluation toolkit SentEval[2] (Conneau and Kiela, 2018) was used. Scikit-Learn's logistic regression with the prototyping config as defined in the SentEval README was used as the inference model to obtain SHAP values. For the main classifier, a PyTorch Multilayer Perceptron with the default config was used. The reason for this distinction in models is twofold: first, the simpler logistic regression allows for faster training times. Secondly, the SHAP values are able to be computed efficiently using the LinearExplainer module, with $nsamples$ set to the default 100. While the model agnostic KernelExplainer can be used to explain any deep neural network, it has significantly longer runtime, especially given our 300 features.

For all reduced embeddings, the pre-trained 300 dimensional Glove (Pennington et al., 2014) embeddings trained on the Wikipedia 2014 and Gigaword 5 corpus was the base embedding.[3] Scikit Learn's implementation of PCA was used with a random state of 324.

The code used to run the experiments and generate the visualizations shown in this paper is public on Github.[4]

3 Evaluation Results

The proposed SHAP-based algorithms were tested on SentEval classification tasks and compared with baselines at 50, 100, 150, and 200 dimensions.[5] For all implementation of the PPA algorithm, the threshold parameter D was set to 7, per the findings in Mu and Viswanath that the top 7 components are disproportionately contributing to variance. Figures 1-4 depict each embeddings' scores on select classification tasks at different dimensionalities. Table 3 shows all results. Figures 1-7 depict accuracy scores from embeddings at various dimensionalities.

3.1 Analysis and Discussion

Impact of Variance-Based Techniques As shown in Table 4, the SHAP algorithm was fairly overwhelmingly the most effective dimensionality reduction technique used across the classici-

[2]https://github.com/facebookresearch/SentEval
[3]https://nlp.stanford.edu/projects/glove/
[4]https://github.com/parkervg/shap-dim-reduction
[5]Because a 150D Glove embedding is not available, Glove 150D accuracy scores are interpolated

Model	MR	CR	SUBJ	MPQA	SST2	SST5	TREC
Glove_300D	74.89	76.52	90.85	86.76	78.2	40.77	68.4
SHAP-ALGO_50D	72.87	75.66	85.86	84.84	75.18	36.15	47
SHAP-ALGO_100D	74.51	**78.01**	89.99	86.53	76.22	40.36	58
SHAP-ALGO_150D	**75.37**	**77.94**	90.3	**86.81**	76.66	**41.09**	60
SHAP-ALGO_200D	**75.51**	**77.96**	90.28	**86.81**	**78.91**	40.45	63.4
SHAP-PPA_50D	73.55	**77.35**	88.92	85.67	76.11	35.16	48.6
SHAP-PPA_100D	74.82	**77.96**	90.33	86.93	75.62	39.86	55.2
SHAP-PPA_150D	**75.65**	**78.07**	90.47	ß **86.98**	**78.42**	40.32	59.8
SHAP-PPA_200D	**75.69**	**77.59**	90.63	**87.01**	**78.36**	39.59	63
SHAP_50D	73.03	**77.35**	90.02	85.74	75.4	36.88	53
SHAP_100D	**75.26**	**78.33**	90.64	**87.04**	77.59	38.91	63.6
SHAP_150D	**75.35**	**78.86**	**90.97**	**87.22**	78.03	**41.04**	66
SHAP_200D	**75.25**	**75.73**	**90.97**	**87.05**	**78.31**	**41.13**	64.8

Table 3: Comparison of performance (in terms of accuracy) on classification datasets. Bold represents instances where the reduced embeddings perform better than the original 300 dimensional Glove embedding. Numbers in green represents the highest score achieved for that particular task.

Embedding	Average Rank
GLOVE	3.17
ALGO	3.62
SHAP	1.75
SHAP-ALGO	3
SHAP-PPA	3

Table 4: Average rank at reduced dimensions across selected classification tasks.

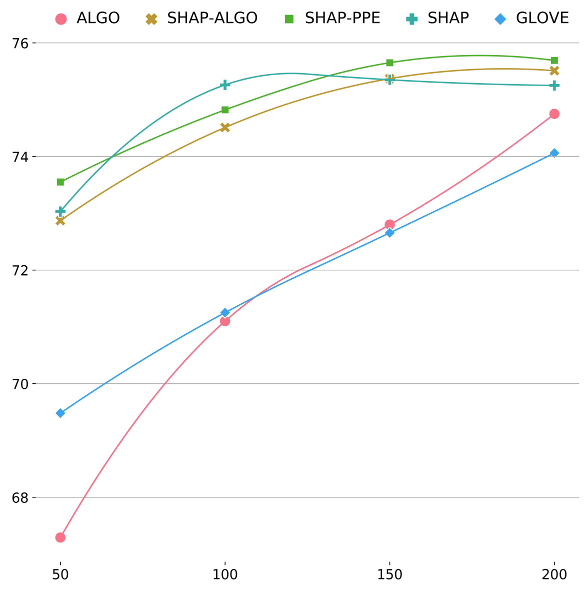

Figure 1: MR

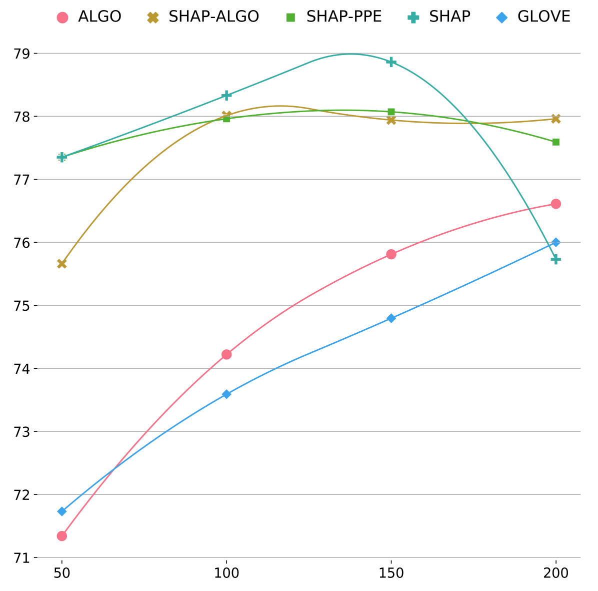

Figure 2: CR

ation tasks, with an average rank of 1.75. Interestingly, looking at the average ranks of the processed embeddings, there is a somewhat positive correlation between average rank and amount of variance-based post-processing. The Algo embeddings, which invoked PCA the most, fared the worst across the classification tasks.

Task Specific Performance As Table 3 shows, the reduced SHAP embeddings largely performed better than the original 300D Glove embeddings. Notably on the CR, with only 50 dimensions the SHAP reduced embeddings outperformed the 300D Glove embeddings by 0.83%. This trend of lower-dimensional embeddings outperforming the original 300D is relatively steady amongst the binary classification tasks; the reduced embeddings outperformed the original Glove embeddings 30 times

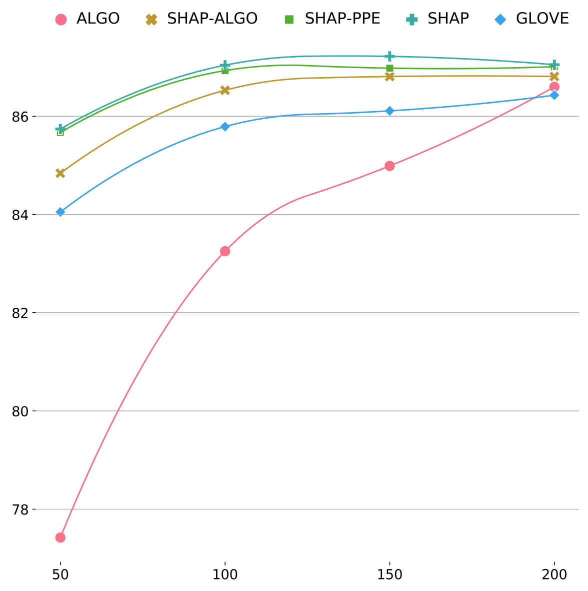

Figure 3: MPQA

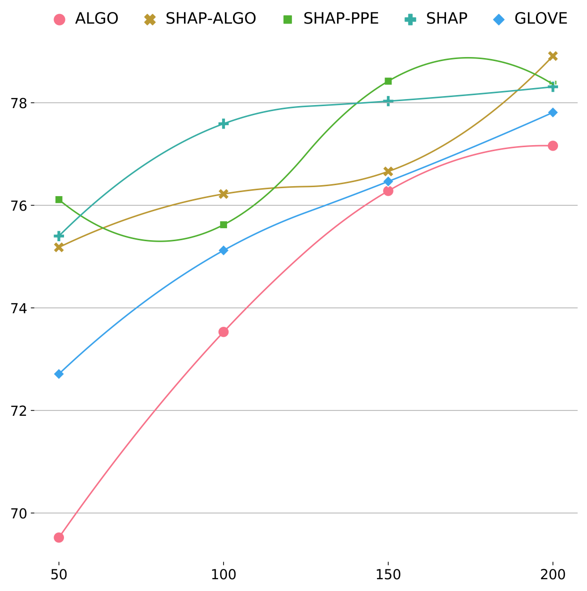

Figure 5: SST2

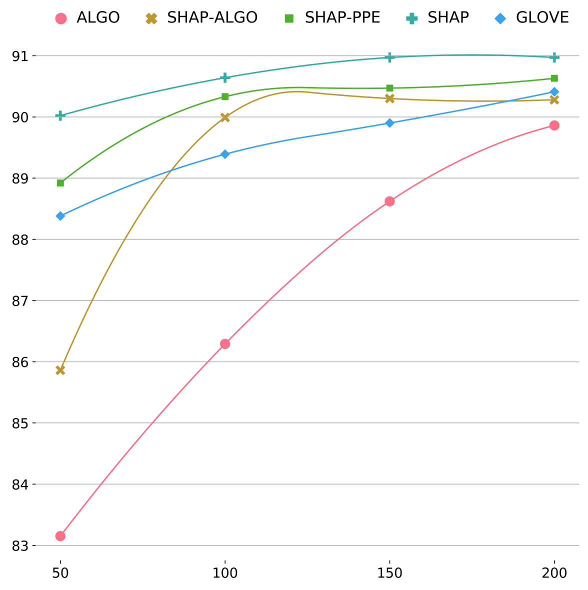

Figure 4: SUBJ

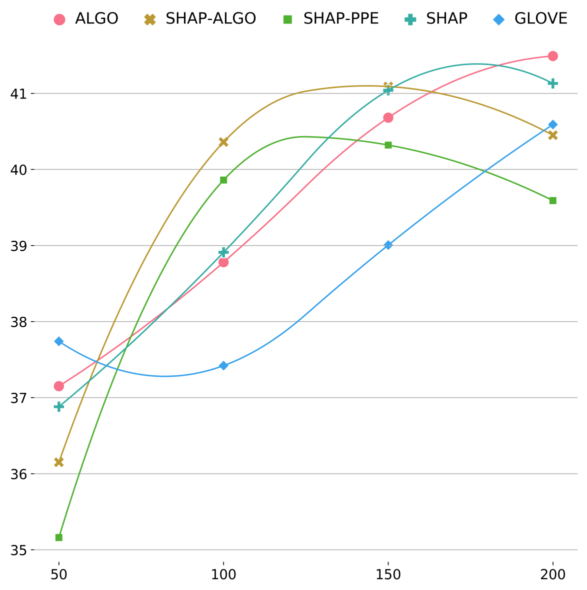

Figure 6: SST5

out of the 60 binary classification evaluations. For multi-class classification tasks, this was only true 3 out of the 24 instances. However, for the multi-class task SST5, the reduced embeddings were within 1.0% of the original embeddings 8 out of the 12 instances. All pre-trained embeddings performed poorly on TREC, and the reduced embeddings only accentuated the poor performance.

4 Explainability with the Subspace Score

Adapted from Jang et al. (2019), the SubspaceScore is defined as follows:

$$SubspaceScore(t) = \frac{\sum_{i \in S} d_i}{|S|}$$

where S is the dimensions indices of the target subspace, $|S|$ is the size of S, and d_i represents the i-th dimension of a text vector t. Text vectors were found by simply taking the mean of all word vectors within a given text.

Using this heuristic, we are able to find those sentences whose class subspace as selected by SHAP exhibit high values. While not as straight forward as explainability within a typical BoW embedding, this offers some high-level insight into which

“and supply those stupid white headphones”

Negative Ngram	Subspace Score	Positive Ngram	Subspace Score
stupid	0.525	supply	0.530
those stupid	0.512	supply those	0.522
stupid white	0.510	and supply	0.518
those stupid white	0.506	and supply those	0.516
supply those stupid	0.506	supply those stupid	0.516

Table 5: Fit to the CR dataset, with $|S|$ = 50

“The movie is about children who learn to fly. It is honestly quite ridiculous.”

Subjective Ngram	Subspace Score	Objective Ngram	Subspace Score
ridiculous	0.523	about	0.527
quite ridiculous	0.512	who	0.522
honestly quite ridiculous	0.507	is about	0.518
quite ridiculous .	0.505	about children who	0.517
children	0.503	who learn	0.516

Table 6: Fit to the SUBJ dataset, with $|S|$ = 50

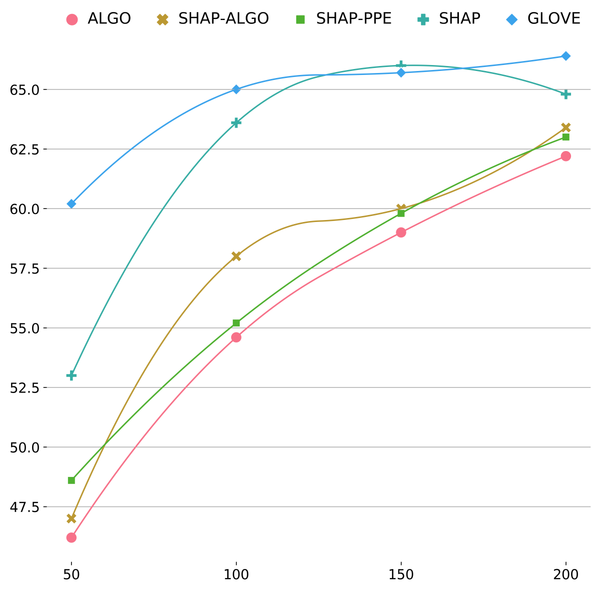

Figure 7: TREC

phrases are influential in yielding $\hat{y}$. Tables 5 and 6 show the results of applying the $SubspaceScore$ on ngrams up to 4-grams, using 300D Glove vectors scaled between 0 and 1 to create the sentence embeddings. The top 5 most influential ngrams for each class are shown.

The two examples above were ultimately classified as negative and subjective, respectively. It is interesting to note how one class prediction can prevail despite high subspace scores for the opposing class. For instance, relative clauses were shown to be highly influential in pushing a prediction to the objective class across all samples. In Table 6, the influence of the subjective dimensions outweigh this common trait amongst objective samples. Table 7 shows the top ngrams for both the subjective and objective label, with $|S|$ = 10. Since all train text was considered with no seperation between labels, it is possible for an ngram appearing in an objective text is ranked highly in the subjective subspace scores, and vice versa. Examining the subspace scores in table 7, it’s clear that relative clauses are more influential in pushing a text to the subjective label than the less coherent objective ngrams.

5 Discussion and Analysis

5.1 Transfer Applications of SHAP Subspaces

I wanted to go further to examine whether the SHAP reduced embeddings were truly cutting away syntactic noise and fitting to dimensions related to the linguistic qualities required by the specific task, or merely fitting to noisy pecularities of the dataset. Table 2 supports the first scenario, and the working hypothesis is that for the simple task of binary classification, only a small subset of the original 300 dimensions is useful in drawing discriminative lines. This was further tested on a binary sentiment classification dataset of Amazon reviews not included in the SentEval package.[6] If there is a large intersec-

[6]https://www.kaggle.com/bittlingmayer/amazonreviews

Top Ngrams Across SUBJ Train Data

Subjective N-grams	Subspace Score	Objective N-grams	Subspace Score
visceral excitement	0.328	wealthy american	0.456
zhao benshan	0.327	businessman who	0.446
cel animation	0.325	african american	0.436
drug kingpin	0.304	officer who	0.435
vincent d'onofrio	0.301	wife who	0.429
slapstick humor	0.299	writer who	0.421
cgi animation	0.286	director who	0.419
spousal abuse	0.285	american living	0.418
visual panache	0.282	marine who	0.418
insider clique	0.281	australian director	0.416

Table 7: Fit to the SUBJ dataset, with $|S| = 10$

tion of dimensions identified by SHAP for both the Amazon Review dataset and a related dataset like the CR (Customer Review) dataset, the conclusion might be made that a subset of raw dimensions carry disproportionate syntactic features relevant to the task of sentiment classification.

While the original dataset contains 3 million reviews, subsets of 10k for training and 5k for testing were taken for the purpose of these tests. 50 dimensional reduced embeddings were tested to find the transfer potential of only those most influential dimensions.

Reduction Target	Embedding	Accuracy
	Glove-300D	83.02
Amazon Reviews	Algo-50D	72.52
Amazon Reviews	SHAP-50D	80.62
CR	SHAP-50D	77.46
MR	SHAP-50D	77.76
TREC	SHAP-50D	74.98
	Random-50D	73.28

Table 8: Accuracy scores on the Amazon reviews dataset. "Reduction Target" refers to the dataset on which logistic regression was trained to get the top 50 SHAP dimensions.

After running a 50D SHAP reduction on both the Amazon Review dataset and other SentEval datasets, the Amazon Review reduced embeddings shared 19 dimensions in common with the CR reduced embeddings, 15 in common with the MR reduced embeddings, and 7 in common with the TREC reduced embeddings. These results are in line with the hypothesis that the similar task of the CR (Customer Review) and MR (Movie Review) dataset would share more influential dimensions with the Amazon Review dataset as opposed to the unrelated query classification dataset TREC. Treating the SHAP reduction of these 50 relevant dimensions as a binary classification problem across 300 features, the matthews correlation coefficients in table 9 are produced. The matthews coefficient measures the quality of binary classifications on a scale of -1 (inverse prediction) and +1 (perfect prediction).

Task	Common Dimensions	Matthews Coefficient
CR	19	0.256
MR	15	0.16
TREC	7	-0.032

Table 9: Common dimensions between SHAP-50D reduced embeddings on specified tasks and SHAP-50D reduced embeddings on the Amazon Review dataset.

5.2 Distinction Between Classification Types

As described in section 3.1, the reduced SHAP embeddings performed far worse on the multi-class classification tasks than the binary-classification tasks. This is likely due to the complexity of the task at hand: for a simple binary classification of sentiment, simple features would likely suffice for a high accuracy score. However, for a dataset like TREC, more nuanced features must be utilized to draw discriminative lines. Below is an example of the TREC dataset.

```
NUM:dist
 How far is it from Phoenix to L.A.?

LOC:city
 What county is Phoenix, AZ in?

HUM:desc
```

Model	MR	CR	SUBJ	MPQA	SST5	TREC
PCA_50D	66.2	71.05	86.89	77.4	35.75	57.6
PCA_100D	70.67	73.3	88.77	83.18	37.47	61.2
PCA_150D	73.36	75.52	89.66	85.38	40.54	62
PCA_200D	73.69	76.53	90.28	86.22	40.09	64
Random_50D	**68.11**	70.48	85.67	**81.56**	**36.54**	55.06
Random_100D	**71.67**	**73.7**	88.49	**84.84**	**39.08**	**61.95**
Random_150D	73.29	74.98	**89.72**	**85.95**	40.16	**64.28**
Random_200D	**74.13**	75.6	**90.36**	**86.31**	**40.55**	**65.11**

Table 10: PCA reduced vs. randomly reduced embeddings. Bold represents instances where the randomly reduced embeddings outperformed their PCA counterparts.

```
  Who was Galileo?

DESC:def
  What is an atom?
```

In the above example, the line between the "Phoenix" question as a NUM and "Phoenix" as a LOC is likely not identifiable from a cursory glance at the features. Because of the complex relationship required to draw discriminative lines in these multi-class problems, methods that aim to maximize global variance seem to generally outperform the SHAP-based approaches tested here.

5.3 Failure of Variance-Based Methods

Interestingly, simply taking a random sample of dimensions from word embeddings outperformed PCA dimensionality reduction at nearly every level. Table 10 shows these results. The random embeddings were created by sampling n dimensions from the original 300D Glove embeddings 25 times. The average scores across these embeddings were then recorded.

5.4 Runtime Related Considerations

While the performance gains reported on the classification tasks are promising, it's important to note the trade-off in runtime between variance based methods like the Algo approach with any of the SHAP reduction techniques. As mentioned before, each run of a SHAP reduction must train a basic logistic regression model on the target data before extracting the top N dimensions. As a result, this approach is inherently different from a one-size-fits-all PCA-based reduction.

Running algorithm 2 from Raunak et al. took 54 seconds in a Google Colaboratory environment with no hardware accelerator.[7] Table 8 shows the runtime of a SHAP reduction to 50D from the raw 300D Glove embeddings across the different datasets. For each task, SHAP reduction on the original embeddings was run 10 times, and the average runtimes were recorded.

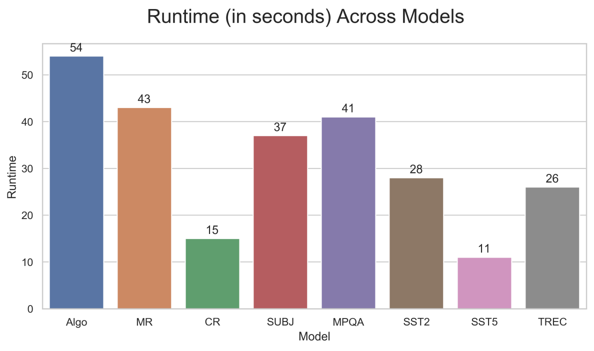

Figure 8: Runtimes across reduction algorithms

As shown in figure 8, the SHAP reductions for the given datasets are all individually faster than the Algo algorithm. However, it is important to note the practical implications of a task-specific approach like SHAP reduction. While the reduced Algo embeddings can be applied to any downstream task with similar results, the SHAP reduced embeddings have been fit to the specific task at hand. Despite yielding promising results on a sentiment classification task, the 50 dimensions fit to the CR (Customer Review) dataset will likely perform poorly on an unrelated task like that of SUBJ (a subjectivity status classification).

6 Conclusion and Future Work

This paper has offered an alternative methodology to variance-based solutions for reducing word embeddings in the context of classification tasks. As a result, accuracy losses are minimized and, in many

[7]https://github.com/vyraun/Half-Size/blob/master/algo.py

cases, improved. This work also extends the concept of the subspace score from Jang et al. (2019) to make explanations on sentence classifications that produce insight into the learned linguistic features of the model.

Moving forward, it would be interesting to see if the accuracy gains observed on the simpler logistic regression/perceptron models would be equivalent to those gains made with more widely used deep models.

Additionally, studying the isotropy (discussed by Mu and Viswanath and Arora et al.) of the SHAP reduced embeddings would be interesting in an attempt to clarify whether the accuracy gains are truly coming from capturing those syntactically relevant dimensions, or are merely a byproduct of some other geometric qualities.

Finally, it would be interesting to see if a quantitative approach to determining the optimal number of dimensions to use for a given SHAP dimensionality reduction exists. With PCA, the general rule of thumb is to take those principle components where 99% cumulative variance is explained; if any equivalent heuristic exists for a SHAP reduction would be interesting to study in the future.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A Latent Variable Model Approach to PMI-based Word Embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kyoung-Rok Jang, Sung-Hyon Myaeng, Hee-Cheol Seo, and Joohee Park. 2019. Selection and Interpretation of Embedding Subspace for Query Classification.

Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-Top: Simple and Effective Postprocessing for Word Representations.

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)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective Dimensionality Reduction for Word Embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Vikas Raunak, Vaibhav Kumar, Vivek Gupta, and Florian Metze. 2020. On Dimensional Linguistic Properties of the Word Embedding Space. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 156–165, Online. Association for Computational Linguistics.

Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications using Game Theory. *The Journal of Machine Learning Research*, 11:1–18.