Paper 3: Unsupervised System Evaluation
The Influence of Down-Sampling on Embeddings in Downstream Tasks

Gareth Harcombe
University of Canterbury
33186156
grh102@uclive.ac.nz

Abstract

The use of down-sampling when training word embeddings has been shown to be unstable, resulting in different word representations for the same data and algorithms. This paper investigates whether the unstable behaviour in GloVe and SGNS embedding algorithms has an impact on downstream performance in the OffensEval shared task, and finds that the performance GloVe embeddings greatly varies with down-sampling, whereas SGNS performance shows significantly less variance. These findings contrast previous research that shows GloVe being more stable than SGNS.

1 Introduction

Word embeddings are a dense vector space to represent a vocabulary, and is created by analysing the co-occurrences of words in a context window. This typically results in a vector space where similar words are close to each other, and relationships between words can be expressed mathematically.

Down-sampling is a common technique used to decrease the impact of high-frequency words when considering the co-occurrence of words, or increase the relative importance of words closer to the center of the context window (Levy et al., 2015). However, Hellrich et al. (2019) showed that down-sampling can have significant impacts on the stability of word embeddings, with models trained independently on the same text corpus with down-sampling resulting different embeddings.

This paper will investigate the impact of unstable embedding spaces on downstream tasks. Different does not always mean poor performance, and it is possible that two different embedding spaces are both capable of representing vocabularies sufficiently such that they perform well on downstream tasks. This paper trains Global Vector (GloVe) (Pennington et al., 2014) and Skip-Gram with Negative Sampling (SGNS) (Mikolov et al., 2013) models with down-sampling, and evaluates the variance of performance by embeddings on the shared task SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval) (Zampieri et al., 2019b).

2 Data

2.1 Embedding Training Data

GloVe and SGNS embeddings were trained on a relatively small corpus of the first 100M characters of the English Wikipedia data on March 3rd 2006 (Mahoney, 2011)[1]. This corpus was chosen because its small size allows multiple models to be trained quickly. Although the small corpus size may limit the accuracy of the embeddings for downstream tasks, it still illustrates any potential performance differences between embedding spaces.

2.2 Downstream Shared Task

This shared task uses English tweets from the Offensive Language Identification Dataset (OLID) dataset, presented by Zampieri et al. (2019a). The OLID dataset includes 14,100 annotated tweets, with 13,240 tweets used for training and 860 tweets used for testing. Two examples of tweets with labels from the dataset are found in Table 1.

2.3 Annotation

OLID's annotation scheme is a hierarchical scheme with three levels to include the target of the offensive language and the type of offensive language (Zampieri et al., 2019a). The first level is to classify offensive (OFF) and non-offensive (NOT) tweets. Offensive tweets can include profanity, and any kind of insult or threat, regardless of whether it is direct or veiled.

[1]The dataset can be downloaded here: http://mattmahoney.net/dc/text8.zip

Tweet	Sub-task A Label
@USER He is so generous with his offers.	NOT
IM FREEEEE!!!! WORST EXPERIENCE OF MY FUCKING LIFE	OFF

Table 1: Two tweets from the OLID dataset with labels for sub-task A

If the tweet is labelled as offensive, then the second level of annotation is whether the offensive language is targeted or not. This is a binary classification of whether the post includes targeted insults or threats to a group or individual, or whether the post contains profanity.

If the tweet contains targeted insults or threats, then the third level of annotation focuses on the target of the offensive language. There are three labels to this sub-task: individual, such as a famous person, named or unamed individual; group, such as ethnicity, gender or religion; and other, such as an organisation, event or issue.

This evaluation focuses on labels from the first level, which annotates each document as either offensive (OFF) or not offensive (NOT).

Tweets were annotated using the crowd sourcing platform Figure Eight[2]. In order to used reliable and accurate annotators, only annotators who were experienced and passed a threshold of test queries were used. All tweets were annotated by two annotators, with a third annotation and a majority vote being used in the event of a disagreement.

2.4 Class Imbalance

There is a significant class imbalance in the OLID data set, as the number of non offensive documents outnumbers the number of non offensive documents. 66.7% and 72.1% of the training and testing data respectively is labelled as not offensive. This is somewhat representative of all tweets, as the majority of tweets are not going to be offensive. However, this does also mean that a system can achieve reasonable level of accuracy by simply choosing the most common label, not offensive. In doing so, the system will not detect any offensive language, voiding the overall purpose of the shared task. To avoid this issue, the standard metric to report for this task is the macro-averaged F1 score to measure precision and recall as well as accuracy. This presents a more accurate representation of how well the system performs at detecting offensive language.

[2] https://figure-eight.com

3 Systems

3.1 GloVe

Global Vector has two steps to combine global statistics with local statistics in an effort to produce a more holistic model that incorporates both global matrix factorisation and local context window information (Pennington et al., 2014). The first step is to populate a word-word co-occurrence matrix for the whole corpus, generating global statistics on how often words occur together. The second step is to train a weighted least squares regression model to minimise J, the difference between word vector dot products and the logarithm of the words' probability of co-occurrence, defined as:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \bar{w_j} + b_i + \bar{b_j} - \log X_{ij})^2 \quad (1)$$

where w_i and b_i are the word vector and bias respectively of word i, $\bar{w_j}$, and b_j are the context word vector and bias respectively of word j, X_{ij} the co-occurrences of words i and j, and f is a weighting function. By combining global and local information, GloVe both effectively leverages statistics of the corpus and performs well on analogy tasks.

Down-sampling in GloVe is implemented by weighting co-occurrence counts. For each co-occurrence observed of words w_i and w_j in the window size s, $df \cdot ff$ is added to the corresponding entry in the word-context matrix, where df and ff are given by Equations 2 and 4 respectively. In Equation 4, $r(w)$ is each token's relative frequency, and t is a predefined threshold value.

$$df(w_i, w_j) := \frac{s + 1 - |j - i|}{s} \quad (2)$$

$$ff(w) := \begin{cases} \sqrt{t/r(w)} & \text{if} r(x) > t \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

$$ff(w_i, w_j) := ff(w_i) \cdot ff(w_j) \quad (4)$$

3.2 SGNS

Skip-Gram with Negative Sampling is an embedding method developed by (Mikolov et al., 2013) that attempts to predict the surrounding words in a window given the current word, as shown in Figure 1. This is done by considering the current word and the neighbouring context words as positive examples, and randomly sampling other words from the vocabulary as negative examples, and distinguishing between the true context words and the negative samples. This is done by using logistic regression to train a classifier to determine whether words are commonly found together in the positive examples, or whether they are not commonly found together in the negative examples. The weights of the classifier for each word then results in the embedding for each word. Negative sampling improves the efficiency of training the model, as the model parameters are only updated for a small number of negative samples instead of for the entire vocabulary.

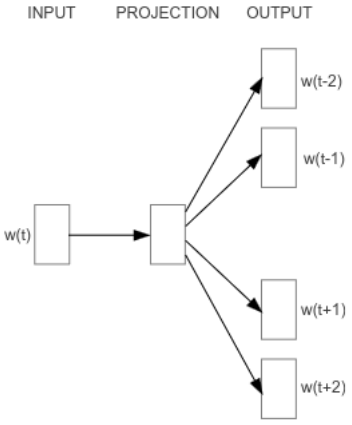

Figure 1: SGNS architecture (Mikolov et al., 2013)

Down-sampling in SGNS is implemented by processing each word-context pair with a probability of $df(w_i, w_j)$ given by Equation 2.

4 Experiment Set-up

The code for implementing these embedding systems can be found in Appendix A.

The training parameters largely follow those used by Hellrich et al. (2019). Both GloVe and SGNS embeddings are of size 50 and trained for 10 epochs with a window size of 5 to allow for the models to be trained multiple times relatively quickly. Default learning rates were used for both models. SGNS used 5 negative samples, and a frequent word down-sampling threshold of 10^{-4}. GloVe used a frequent word down-sampling threshold of 100. Five models were trained for each of GloVe and SGNS algorithms.

Intrinsic stability was quantified by choosing the 1,000 most frequent words in the text corpus from Section 2.1 as anchor words and calculating $j@10$ by Equation 5.

4.1 Intrinsic Evaluation

Hellrich et al. (2019) quantifies intrinsic stability as the overlap in sets of words considered most similar to a set of anchor words. This gives rise to the $j@n$ metric given in Equation 5: given M models and a set A of anchor words, for each model m find the n closest words to each anchor word a. To quantify this, average the Jaccard coefficient between the sets of closest words produced by the M models given by the most similar words function $msw(a, n, m)$ over all anchor words a in A.

$$j@n := \frac{1}{|A|} \sum_{a \in A} \frac{|\bigcap_{m \in M} msw(a, n, m)|}{|\bigcup_{m \in M} msw(a, n, m)|} \quad (5)$$

4.2 Extrinsic Evaluation on Downstream Tasks

The embedding spaces are evaluated on the OffensEval shared task through directly comparing embeddings. For a given training example, the embeddings for every word in the tweet are averaged together to form an embedding for the tweet, resulting in 13,240 training vectors. When classifying a test example, the embeddings for every word are again averaged together. The label for the test tweet is given by the label of the closest training embedding.

This is not the most sophisticated method for using embeddings, as it is common to use a neural network to classify embeddings. However, training an additional neural network introduces additional variance to the testing process, which may result in any variance between embeddings' classification accuracy being due to different neural network accuracy, rather than any inherent behaviour of the embedding. By directly using the embedding space in the downstream classification task, any difference in results can be directly attributed to the difference in embedding spaces.

5 Results

The F1 scores on the OffensEval task for the five down-sampled trainings for each of SGNS and GloVe are shown in Table 2, along with the mean and standard deviation of the F1 scores. The intrinsic stability of SGNS and GloVE using $j@10$ is shown in Table 3.

The stability of SGNS at 0.430 is significantly higher than the stability of GloVe at 0.022. Interestingly, this is contrary to the stability discussed by Hellrich et al. (2019), which saw the stability of GloVe higher than SGNS in five out of six corpora. The difference in results is likely due to: different training data, specifically a smaller dataset; and hyper-parameters such as number of training epochs used for the experiments.

Although the mean F1 score for SGNS and GloVE is relatively similar within each pooling method, the direct F1 scores is not particularly relevant for this paper, as the performance of the embeddings is not the focus of this paper. Instead, the standard deviation or variance of the F1 scores is more important, as this shows how the instability of embeddings impacts the performance on downstream tasks.

Hellrich et al. (2019) found GloVe to be the third most reliable or stable algorithm after Singular Value Decomposition (SVD) methods, and more stable than SGNS. However, this is not reflected in the above results, where GloVe has significantly higher variance in performance on both the intrinsic evaluation, and the downstream task than SGNS across all pooling methods. The higher variation in downstream task performance by the GloVe model is likely caused by the higher intrinsic instability in this specific implementation, which has negatively impacted the stability of the model in the downstream task. Hence, it is reasonable to assume that instability observed in intrinsic evaluations corresponds to instability in downstream task evaluations.

Average and Sum pooling methods saw similar standard deviation in performance. Given that the Average pooling method is the same as the Sum pooling method without dividing by the number of embedding vectors, this is not unexpected. However, the Max pooling method saw significantly higher variation in performance. This may indicate instability in Max pooling, whereby a few vectors with large values can dominate and ignore the majority of other vectors, especially vectors with strongly negative values that would be taken into account by the Average and Sum methods.

6 Conclusion

This paper investigated the the effect of down-sampling on word embedding stability and performance in downstream tasks, comparing the performance of GloVe and SGNS embeddings on the shared task OffensEval. The GloVe models showed significantly higher intrinsic instability than the SGNS models, contradicting the findings on embedding stability by Hellrich et al. (2019), and consequently had higher variation in performance in the downstream task.

6.1 Future Work

There are several avenues for further investigation. For example, exploring why the GloVe model shows higher instability with this training data and experiment set-up could give more insight into how embedding models learn. Similarly, it would also be useful to investigate how comparable down-sampling methods are between embedding algorithms, such as how the threshold parameter in weighting down-sampling used in GloVe compares to the probabilistic down-sampling used in SGNS, to ensure that future experiments are performed with comparable down-sampling methods.

It would also be interesting to see if GloVe's higher variance in performance still observed when a more sophisticated downstream method is used, such as using neural networks for classification.

References

Johannes Hellrich, Bernd Kampe, and Udo Hahn. 2019. The Influence of Down-Sampling Strategies on SVD Word Embedding Stability. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 18–26, Minneapolis, USA. Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Matt Mahoney. 2011. About the test data. Website. Accessed 17th May 2023.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. Technical report. ArXiv:1301.3781 [cs] type: article.

Pooling Method	Embedding	F1					Mean	Standard Deviation
Average	GloVe	0.520	0.501	0.525	0.556	0.604	0.539	0.0398
	SGNS	0.548	0.534	0.561	0.555	0.552	**0.548**	**0.00890**
Sum	GloVe	0.528	0.528	0.552	0.527	0.589	0.545	0.0269
	SGNS	0.573	0.567	0.577	0.581	0.588	**0.577**	**0.00795**
Max	GloVe	0.523	0.546	0.533	0.521	0.604	**0.545**	0.0342
	SGNS	0.562	0.556	0.530	0.523	0.523	0.539	**0.0188**

Table 2: Embedding downstream task results with different pooling methods. **Bold** indicates a higher mean or lower standard deviation.

Embedding	Stability
GloVe	0.022
SGNS	0.430

Table 3: Intrinsic Stability

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.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. 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 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

A Code

All of the code to implement the GloVe and SGNS models and evaluate them against OffensEval (including the data used for testing and training) can be found in the following repo: `https://github.com/GarethHarcombe/unsupervised-systems`