When Hyperparameters Help: Beneficial Parameter Combinations in Distributional Semantic Models

Alicia Krebs
a.m.krebs@student.rug.nl

Denis Paperno
denis.paperno@unitn.it

Center for Mind and Brain Sciences (CIMeC), University of Trento, Rovereto, Italy

Abstract

Distributional semantic models can predict many linguistic phenomena, including word similarity, lexical ambiguity, and semantic priming, or even to pass TOEFL synonymy and analogy tests (Landauer and Dumais, 1997; Griffiths et al., 2007; Turney and Pantel, 2010). But what does it take to create a competitive distributional model? Levy et al. (2015) argue that the key to success lies in hyperparameter tuning rather than in the model's architecture. More hyperparameters trivially lead to potential performance gains, but what do they actually do to improve the models? Are individual hyperparameters' contributions independent of each other? Or are only specific parameter combinations beneficial? To answer these questions, we perform a quantitative and qualitative evaluation of major hyperparameters as identified in previous research.

1 Introduction

In a rigorous evaluation, (Baroni et al., 2014) showed that neural word embeddings such as skip-gram have an edge over traditional count-based models. However, as argued by Levy and Goldberg (2014), the difference is not as big as it appears, since skip-gram is implicitly factorizing a word-context matrix whose cells are the pointwise mutual information (PMI) of word context pairs shifted by a global constant. Levy et al. (2015) further suggest that the performance advantage of neural network based models is largely due to hyperparameter optimization, and that the optimization of count based models can result in similar performance gains. In this paper we take this claim as the starting point. We experiment with three hyperparameters that have the greatest effect on model performance according to Levy et al. (2015): subsampling, shifted PMI and context distribution smoothing. To get a more detailed picture, we use a greater range of hyperparameter values than in previous work, comparing all hyperparameter value combinations, and perform a qualitative analysis of their effect.

2 Hyperparameters Explored

2.1 Context Distribution Smoothing (CDS)

Mikolov et al. (2013b) smoothed the original contexts distribution raising unigram frequencies to the power of alpha. Levy and Goldberg (2015) used this technique in conjunction with PMI.

$$PMI(w,c) = \log \frac{\hat{P}(w,c)}{\hat{P}(w) \cdot \hat{P}_\alpha(c)}$$

$$\hat{P}_\alpha(c) = \frac{\#(c)^\alpha}{\sum_c \#(c)^\alpha}$$

After CDS, either PPMI or Shifted PPMI may be applied. We implemented CDS by raising every count to the power of α, exploring several values for α, from .25 to .95 to 1 (no smoothing).

2.2 Shifted PPMI

Levy and Goldberg introduced Shifted Positive Pointwise Mutual Information (SPPMI) as an association measure more efficient than PPMI. For every word w and every context c, the SPPMI of w is the higher value between 0 and its PMI value minus the log of a constant k.

$$PPMI(w,c) = \max(\log \frac{P(w,c)}{P(w)P(c)}, 0)$$

$$SPPMI_k(w,c) = \max(PMI(w,c) - \log k, 0)$$

2.3 Subsampling

Subsampling was used by Mikolov et al. as a means to remove frequent words that provide less information than rare words (Mikolov et al., 2013a). Each word in the corpus with frequency above treshold *t* can be ignored with probability *p*, computed for each word using its frequency *f*:

$$p = 1 - \sqrt{\frac{t}{f}}$$

Following Mikolov et al., we used $t = 10^{-5}$. In word2vec, subsampling is applied before the corpus is processed. Levy and Goldberg explored the possibility of applying subsampling afterwards, which does not affect the context window's size, but found no significant difference between the two methods. In our experiments, we applied subsampling before processing.

3 Evaluation Setup

3.1 Corpus

For maximum consistency with previous research, we used the cooccurrence counts of the best count-based configuration in Baroni et al. (2014), extracted from the concatenation of the web-crawled ukWack corpus (Baroni et al., 2009), Wikipedia, and the BNC, for a total of 2.8 billion tokens, using a 2-word window and the 300K most frequent tokens as contexts. This corpus will be referred to as `WUB`. For comparison with a smaller corpus, similar to the one in Levy and Goldberg's setup, we also extracted cooccurrence data from Wikipedia alone, leaving the rest of the configuration identical. This corpus will be referred to as `Wiki`.

3.2 Evaluation Materials

Three data sets were used to evaluate the models. The MEN data set contains 3000 word pairs rated by human similarity judgements. Bruni et al. (2014) report an accuracy of 78% on this data-set using an approach that combines visual and textual features. The WordSim data set is a collection of word pairs associated with human judgements of similarity or relatedness. The similarity set contains 203 items (`WS sim`) and the relatedness set contains 252 items (`WS rel`). Agirre et al. achieved an accuracy of 77% on this data set using a context window approach (Agirre et al., 2009). The TOEFL data set includes 80 multiple-choice synonym questions (Landauer and Dumais, 1997). For this data set, corpus-based approaches have reached an accuracy of 92.50% (Rapp, 2003).

4 Results

4.1 Context Distribution Smoothing

Our results show that smoothing is largely ineffective when used in conjunction with PPMI. It also becomes apparent that .95 is a better parameter than .75 for smoothing purposes.

		MEN	WS rel	WS sim	toefl
WUB	.25	.6128	.3740	.5814	.62
	.50	.6592	.4419	.6283	.68
	.70	.6938	.5113	.6708	.72
	.75	.7008	.5249	.6788	.75
	.80	.7069	.5393	.6866	.76
	.85	.7119	.5517	.6950	**.77**
	.90	.7162	.5625	.6998	**.77**
	.95	.7197	**.5730**	**.7043**	**.77**
	1.0	**.7208**	.5708	.7001	.76
Wiki	.75	.7194	.4410	.6906	.76
	.85	.7251	.4488	.7001	.76
	.95	**.7277**	**.4534**	.7083	**.77**
	1.0	.7224	.4489	**.7158**	.76

Table 1: Context Distribution Smoothing

4.2 Shifted PPMI

When using SPPMI, Levy and Goldberg (2014) tested three values for k: 1, 5 and 15. On the MEN data set, they report that the best k value was 5 (.721), while on the WordSim data set the best k value was 15 (.687). In our experiments, where (in contrast to Levy and Goldberg) all other hyperparameters are set to 'vanilla' values, the best k value was 3 for all data sets.

4.3 Smoothing and Shifting Combined

The results in Table 3 show that Context Distribution Smoothing is effective when used in conjunction with Shifted PPMI. With CDS, 5 turns out to be a better value than 3 for k. These results are also consistent with the previous experiment: a smoothing of .95 is in most cases better than .75.

4.4 Subsampling

Under the best shifting and smoothing configuration, subsampling can improve the model's performance score by up to 9.2% (see Table 4). But in

		MEN	WS rel	WS sim	toefl
WUB	1	.7208	.5708	.7001	**.76**
	2	.7298	.5880	.7083	.75
	3	**.7314**	**.5891**	**.7113**	**.76**
	4	.7308	.5771	.7071	**.76**
	5	.7291	.5651	.7034	.75
	10	.7145	.5138	.6731	.72
	15	.6961	.4707	.6464	.71
Wiki	1	.7224	.4489	.7158	.76
	3	**.7281**	**.4575**	**.7380**	**.77**
	4	.7269	.4553	.7376	.75
	5	.7250	.4504	.7334	.76

Table 2: Shifted PPMI

the absence of shifting and smoothing, subsampling does not produce a consistent performance change, which ranges from -6.7% to $+7\%$.

The nature of the task is also important here: on `WS rel`, subsampling improves the model's performance by 9.2%. We assume that diversifying contextual cues is more beneficial in a relatedness task than in others, especially on a smaller corpus.

5 Qualitative Analysis

CDS and SPPMI increase model performance because they reduce statistical noise, which is illustrated in Table 5. It shows the top ten neighbours of the word *doughnut* in the vanilla PPMI configuration vs. SPPMI with CDS, in which there are more semantically related neighbours (in bold).

To visualize which dimensions of the vectors are discarded when shifting and smoothing, we randomly selected a thousand word vectors and compared the number of dimensions with a positive value for each vector in the vanilla configuration vs. log(5)cds(.95). For instance, the word *segmentation* has 1105 positive dimensions in the vanilla configuration, but only 577 in the latter.

For visual clarity, only vectors with 500 or less contexts are shown in Figure 1.

This figure indicates that the process of shifting and smoothing appears to be largely independent from the number of contexts of a vector: a word with a high number of positive contexts in the vanilla configuration may very well end up with zero positive contexts under SPPMI with CDS.

The independence of the number of positive contexts under the vanilla configuration from the probability of having at least one positive context

	MEN	WS rel	WS sim	toefl
		WUB		
log(1) cds(1.0)	.7208	.5708	.7001	.76
log(3) cds(.75)	.7319	.5969	.7146	.73
log(3) cds(.90)	.7371	.6170	.7285	.76
log(3) cds(.95)	.7379	.6201	.7315	.76
log(4) cds(.75)	.7363	.6071	.7212	.75
log(4) cds(.90)	.7398	.6222	.7351	.76
log(4) cds(.95)	.7403	**.6265**	.7392	**.77**
log(5) cds(.75)	.7387	.6115	.7281	.76
log(5) cds(.90)	.7412	.6223	.7404	**.77**
log(5) cds(.95)	**.7414**	.6257	**.7434**	**.77**
		Wiki		
log(1) cds(1.0)	.7224	.4489	.7158	**.76**
log(5) cds(.75)	**.7424**	.4787	.7378	.75
log(5) cds(.85)	.7399	.4795	.7418	.75
log(5) cds(.95)	.7362	**.4806**	**.7443**	.75

Table 3: CDS and Shifted PPMI

	MEN	WS rel	WS sim	toefl
		WUB		
log(1) cds(1.0)	.7284	.5043	.6750	**.75**
log(5) cds(.95)	**.7577**	**.5539**	**.7505**	.73
		Wiki		
log(1) cds(1.0)	.7260	.5186	.6965	.72
log(5) cds(.95)	**.7661**	**.5729**	**.7446**	**.76**

Table 4: CDS and SPPMI with subsampling

under SPPMI with CDS is confirmed by the Chi-Square test (χ = 344.26, p = .9058).

We further analysed a sample of 1504 vectors that lose all positive dimensions under SPPMI with CDS. We annotated a portion of those vectors, and found that the vast majority were numerical expressions, such as dates, prices or measurements, e.g. *1745*, which may appear in many different contexts, but is unlikely to have a high number of occurrences with any of them. This explains why its number of positive contexts drops to zero when SPPMI and CDS are applied.

6 Count vs Predict and Corpus Size

We conducted the same experimentations on two corpora: the `WUB` corpus (Wikipedia+ukWack+BNC) used by Baroni et al., and the smaller `Wiki` corpus comparable

log(1) cds(1.0)		*log(5) cds(.95)*	
doughnut	1.0	doughnut	1.0
lukeylad	.467	**donut**	.242
ricardo308	.388	**doughnuts**	.213
katie8731	.376	**donuts**	.203
holliejm	.288	**kreme**	.179
donut	.200	lukeylad	.167
lumic	.187	**krispy**	.149
notveryfast	.183	:dance	.115
adricsghost	.178	bradys	.105
doughnuts	.178	holliejm	.102

Table 5: Top 10 neighbours of *doughnut*. Semantically related neighbors are given in bold.

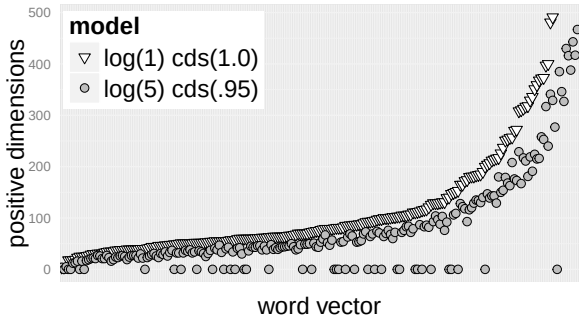

Figure 1: Along the X axis, vectors are ordered by the ascending number of positive dimensions in the vanilla model. The Y axis represents the number of positive dimensions in two models.

to the one that Levy et al. employed. With these two corpora, we found the same general pattern of results, with the exception of the WordSim relatedness task benefitting greatly from a larger corpus and MEN favoring steeper smoothing (.75) under the smaller corpus. This suggests that the smoothing hyperparameter should be adjusted to the corpus size and the task at hand.

For comparison, we give the results for a word2vec model trained on the two corpora using the best configuration reported by Baroni et al. (2014): CBOW, 10 negative samples, CDS, window 5, and 400 dimensions. We find that PPMI is more efficient when using the Wikipedia corpus alone, but when using the larger corpus the predict model still outperforms all count models.

7 Conclusion

Our investigation showed that the interaction of different hyperparameters matters more than the implementation of any single one. Smoothing only shows its potential when used in combination with shifting. Similarly, subsampling only becomes interesting when shifting and smoothing are applied. When it comes to parameter values, we recommend using .95 as a smoothing hyperparameter and log(5) as a shifting hyperparameter.

Qualitatively speaking, the hyperparameters help largely by reducing statistical noise in cooccurrence data. SPPMI works by removing low PMI values, which are likely to be noisy. CDS effectively lowers PMI values for rare contexts, which tend to be more noisy, allowing for a higher threshold for SPPMI ($\log 5$ vs. $\log 3$) to be effective. Subsampling gives a greater weight to underexploited data from rare words at the expense of frequent ones, but it amplifies the noise as well as the signal, and should be combined with the other noise-reducing hyperparameters to be useful.

In terms of corpus size, we've seen that similar performance can be achieved with a smaller corpus if the right hyperparameters are used. One exception is the WordSim relatedness task, in which models require more data to achieve the same level of performance, and benefit from subsampling much more than in the similarity task.

While the best predictive model from Baroni et al. trained on the `WUB` corpus still outperforms our best count model on the same corpus, hyperparameter tuning does significantly improve the performance of count models and should be used when a corpus is too small to build a predictive model.

>0	>300	>750	>1000	>1500
8:23	1900s	e4	1024	51
01-06-2005	7.45pm	8.4	1928.	1981.
ec3n	41.	331	1924.	17
5935	1646	1745	45,000	2500
$1.00	$25	1/3	630	1960s

Table 6: Sample of words with zero positive dimensions after SPPMI with CDS

predict	MEN	WS rel	WS sim	toefl
WUB	.80	.70	.80	.91
Wiki	.7370	.4951	.7714	.83
best count	**MEN**	**WS rel**	**WS sim**	**toefl**
WUB	.7577	.6265	.7505	.77
Wiki	.7661	.5729	.7446	.77

Table 7: Performance of count vs. predict models as a function of corpus size

Acknowledgements

We thank Marco Baroni, the COMPOSES group at the University of Trento, and three anonymous reviewers for their valuable input. This research was supported by the ERC 2011 Starting Independent Research Grant 283554 (COMPOSES) and by the European Masters Program in Language and Communication Technologies.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 238–247.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(1–47).

Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Reinhard Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.