Evaluating Natural Alpha Embeddings on Intrinsic and Extrinsic Tasks

Riccardo Volpi
Machine Learning and Optimization,
Romanian Institute of
Science and Technology (RIST),
Cluj-Napoca, Romania
volpi@rist.ro

Luigi Malagò
Machine Learning and Optimization,
Romanian Institute of
Science and Technology (RIST),
Cluj-Napoca, Romania
malago@rist.ro

Abstract

Skip-Gram is a simple, but effective, model to learn a word embedding mapping by estimating a conditional probability distribution for each word of the dictionary. In the context of Information Geometry, these distributions form a Riemannian statistical manifold, where word embeddings are interpreted as vectors in the tangent bundle of the manifold. In this paper we show how the choice of the geometry on the manifold allows impacts on the performances both on intrinsic and extrinsic tasks, in function of a deformation parameter alpha.

1 Introduction

Word embeddings are compact representations for the words of a dictionary. Rumelhart et al. (1986) first introduced the idea of using the internal representation of a neural network to construct a word embedding. Bengio et al. (2003) employ a neural network to predict the probability of the next word given the previous ones. Mikolov et al. (2010) proposed the use of a recurrency language model based on RNN, to learn the vector representations. More recently, this approach has been exploited further, with great success by means of bidirectional LSTM (Peters et al., 2018) and transformers (Radford et al., 2018; Devlin et al., 2018; Yang et al., 2019). In this paper we focus on Skip-Gram (SG), a well-known model for the conditional probability of the context of a given central word, which it has been shown to work well at efficiently capturing syntactic and semantic information. SG is at the basis of many popular word embeddings algorithms, such as Word2Vec (Mikolov et al., 2013a,b), the contpdfinfoinuous bag of words (Mikolov et al., 2013a,b), and models based on weighted matrix factorization of the global co-occurrences as GloVe (Pennington et al., 2014), cf. Levy and Goldberg (2014). These methods are deeply related, Levy and Goldberg showed how Word2Vec SG with negative sampling is effectively performing a matrix factorization of the Shifted Positive PMI (Levy and Goldberg, 2014).

It has been noted (Mikolov et al., 2013c) how, once the embedding space has been learned, syntactic and semantic analogies between words translate in linear relations between the respective word vectors. There have been numerous works investigating the reason of the correspondence between linear properties and word relations. Pennington et al. gave a very intuitive explanation in their paper on GloVe (Pennington et al., 2014). More recently Arora et al. (Arora et al., 2016) tried to study this property by introducing a hidden Markov model, under some regularity assumptions on the distribution of the word embedding vectors, cf. (Mu et al., 2017). Word embeddings are also often used as input for another computational model, to solve more complex inference tasks. The evaluation of the quality of a word embedding, which ideally should encode syntactic and semantic information, is not easy to be determined and different approaches have been proposed in the literature. This evaluation can be in terms of performance on intrinsic tasks like word similarity (Bullinaria and Levy, 2007, 2012; Pennington et al., 2014; Levy et al., 2015), or by solving word analogies (Mikolov et al., 2013c,a), however several authors (Tsvetkov et al., 2015; Schnabel et al., 2015) has showed a low degree of correlation between the quality of an embedding for word similarities and analogies on one side, and on downstream (extrinsic) tasks, for instance on classification or prediction, to which the embedding is given in input.

Several works have highlighted the effectiveness of post-processing techniques (Bullinaria and Levy, 2007, 2012), such as PCA (Raunak, 2017; Mu et al., 2017), focusing on the fact that certain dominant components are not carriers of semantic nor syn-

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 61–71
July 9, 2020.

tactic information and thus act like noise for determinate tasks of interest. A different approach which still acts on the learned vectors after training has been recently proposed by Volpi and Malagò (2019). The authors present a geometrical framework in which word embeddings are represented as vectors in the tangent space of a probability simplex. A family of word embeddings called natural alpha embeddings is introduced, where α is a deformation parameter for the geometry of the probability simplex, known in Information Geometry in the context of α-connections (Amari and Nagaoka, 2000; Amari, 2016). Noticeably, alpha word embeddings include the classical word embeddings as a special case. In this paper we provide an experimental evaluation of natural alpha embeddings over different tasks, both intrinsic and extrinsic, including word similarities and analogies, as well as downstream tasks, such as document classification and sentiment analysis, in order to study the impact of the geometry on performances.

2 Conditional Models and the Embeddings Structure

The Skip-Gram conditional model (Mikolov et al., 2013b; Pennington et al., 2014) allows the unsupervised training of a set of word-embeddings, by predicting the conditional probability of any word χ to be in the context of a central word w

$$p(\chi|w) = p_w(\chi) = \frac{\exp(u_w^{\mathrm{T}} v_\chi)}{Z_w} \qquad (1)$$

with $Z_w = \sum_{\chi' \in \mathscr{D}} \exp(u_w^{\mathrm{T}} v_{\chi'})$ partition function. The conditional model represents an exponential family in the simplex, parameterized by two matrices U and V of size $n \times d$, where n is the cardinality of the dictionary $\mathscr{D}$, and d is the size of the embeddings. We will refer to the rows of a matrix V as v_χ or V^χ, and to its columns as V_k. It is common practice in the literature of word embedding to consider u_w or alternatively $u_w + v_w$ as embedding vectors for w (Bullinaria and Levy, 2012; Mikolov et al., 2013a,b; Pennington et al., 2014; Raunak, 2017). In the remaining part of this section we briefly review the natural alpha embeddings and limit embeddings, based on Information Geometry framework. We refer the reader to Volpi and Malagò (2019) for more details and mathematical derivations.

2.1 Alpha Embeddings

After training, the matrices U and V are fixed. For each w, the conditional model $p_w(\chi)$ is an exponential family $\mathcal{E}$ in the $n-1$ dimensional simplex, where n is the size of the dictionary. This models the probability of a word χ in the context, when w is the central word. The sufficient statistics of this model are determined by the columns of V, while each row u_w of U can be seen as an assignment for the natural parameters, i.e., each row identifies a probability distribution.

According to the language of Information Geometry, a statistical model can be modelled as a Riemannian manifold endowed with the Fisher information matrix and with a family of α-connections (Amari, 1985; Shun-Ichi and Hiroshi, 2000; Amari, 2016). The alpha embeddings are defined up to the choice of a reference distribution p_0. The natural alpha embedding of a given word w is defined as the projection of the logarithmic map $\mathrm{Log}^\alpha_{p_0} w$ onto the tangent space of the submodel $\mathrm{T}_{p_0}\mathcal{E}$. The main intuition is that a word embedding for w corresponds to the vector in the tangent space which allows to reach the distribution of the context of w from p_0. Deforming the simplex continuously with a family of isometries depending from a parameter alpha, and by considering a family of α-logarithmic maps, depending on the choice of the α-connection, a family of natural alpha embeddings $W^\alpha_{p_0}(w)$ can be defined as a function of the deformation parameter α

$$\begin{aligned} W^\alpha_{p_0}(w) &= \Pi^\alpha_0 \left(\mathrm{Log}^\alpha_{p_0} p_w\right) \\ &= I(p_0)^{-1} \sum_\chi l^\alpha_{p_0\, w}(\chi)\, \Delta V(p_0)^\chi \end{aligned} \qquad (2)$$

where $\Delta V(p_0) = V - \mathrm{E}_{p_0}[V]$ is the matrix of centered sufficient statistics in p_0 and

$$l^\alpha_{p_0\, w}(\chi) = \begin{cases} p_0(\chi)(\ln p_w(\chi) - \ln p_0(\chi)) & \alpha = 1 \\ p_0(\chi)\frac{2}{1-\alpha}\left(\left(\frac{p_w(\chi)}{p_0(\chi)}\right)^{\frac{1-\alpha}{2}} - 1\right) & \alpha \neq 1 \end{cases}. \qquad (3)$$

The Fisher metric is simply computed as the metric for an exponential family (Amari and Nagaoka, 2000)

$$I(p_0) = \mathrm{E}_{p_0}\left[\Delta V(p_0)^{\mathrm{T}} \Delta V(p_0)\right], \qquad (4)$$

and it does not depend on alpha since the family of alpha divergences induces the same Fisher information metric for any value of alpha.

The notion of alpha embeddings can be used both for downstream tasks and also to evaluate similarities and analogies in the tangent space of the manifold (Volpi and Malagò, 2019). Given two words a and b, a measure of similarity is defined by

$$\mathrm{sim}^{\alpha}_{p_0}(a,b) = \frac{\langle W^{\alpha}_{p_0}(a), W^{\alpha}_{p_0}(b)\rangle_{I(p_0)}}{||W^{\alpha}_{p_0}(a)||_{I(p_0)}||W^{\alpha}_{p_0}(b)||_{I(p_0)}} , \tag{5}$$

while analogies of the form $a : b = c : d$ can be solved by minimizing an analogy measure $\kappa^{(\alpha)}_{p_0}(p_a, p_b, p_c, p_d)$ defined as

$$\left\| W^{\alpha}_{p_0}(b) - W^{\alpha}_{p_0}(a) - W^{\alpha}_{p_0}(d) + W^{\alpha}_{p_0}(c) \right\|_{I(p_0)} . \tag{6}$$

It is possible to show that for $\alpha = 1$ and choosing p_0 equal to the uniform distribution, the embeddings of Eq. (2) reduce to the standard vectors u_w. Furthermore, by substituting the Fisher Information matrix $I(p_0)$ with the identity[1], Eqs. (5) and (6) reduce to the standard formulas used in the literature for similarities and analogies.

The embedding vectors $u + v$ have been shown to provide better results (Pennington et al., 2014) than simply u. In the context of natural alpha embeddings, the vectors $u + v$ can be interpreted as a recentering of the natural parameters u of the exponential family. This corresponds to a reweighting of the probabilities in Eq. (1)

$$p^{(+)}(\chi|w) = N_w \exp(v_w v_\chi) p(\chi|w) \tag{7}$$

based on a change of reference measure proportional to $\exp(v_w v_\chi)$, i.e., by weighting more those words χ in the context whose outer vectors are aligned to the outer vector of the central word w.

2.2 Limit Embeddings

The behavior of the alpha embeddings for α progressively approaching minus infinity turns out to be particularly interesting. In this case, $l^{\alpha}_{p_0\, w}(\chi)$ is progressively more and more peaked on

$$\chi^*_w = \arg\max_{\chi} \frac{p_w(\chi)}{p_0(\chi)} , \tag{8}$$

and presents a growing norm, see Eq. (3). By normalizing these alpha embeddings to preserve the direction of the tangent vector, a simple formula can be obtained depending only on the χ^*_w row of the matrix of sufficient statistics $\Delta V(p_0)$. The normalized limit embeddings then simplify to

$$\begin{aligned} LW^{\alpha}_{p_0}(w) &= \lim_{\alpha\to-\infty} W^{\alpha}_0(w) \\ &= I(p_0)^{-1}\Delta V(p_0)^{\chi^*_w} , \end{aligned} \tag{9}$$

leading to simple geometrical methods in the limit. Let us notice that the same row ΔV^a can be associated to multiple words, thus limit embeddings are also naturally inducing a clustering in the embedding space.

3 Experiments

We considered two corpora: English Wikipedia dump October 2017 (enwiki), with 1.5B words, and its augmented version composed by Gutenberg (Gutenberg), English Wikipedia and BookCorpus (Zhu et al., 2015; BookCorpus; Kobayashi) (geb), with 1.8B words. For each corpus we trained a set of GloVe word embeddings (Pennington et al., 2014) with vector sizes of 300 and 50, window size of 10, until convergence for a maximum of 1,000 epochs (more details in Appendix A).

The embeddings in Eq. (2) will be denoted with 'E' in figures and tables, while the limit embeddings in Eq. (9) will be denoted with 'LE'. Embeddings have been normalized either with the Fisher Information matrix (F) or with the Identity (I). Similarly after normalization, the scalar products can be computed with the respective metric (on the tasks that requires scalar product calculation). In this study, normalization and scalar product are always using the same metric. For the reference distribution needed for the computation of the alpha embeddings we have chosen the uniform distribution (0), the unigram distribution of the model (u) - obtained by marginalization of the joint distribution learned by the model, or the unigram distribution estimated from the corpus data (ud). Embeddings are denoted by 'U', if in the computation of Eqs. (2) and (9), the formula used for p_w is Eq. (1), while they will be denoted by 'U+V' if Eq. (7) is used instead.

We evaluated the alpha embeddings on intrinsic (similarities, analogies, concept categorization) and extrinsic (document classification, sentiment analysis) tasks.

3.1 Intrinsic Tasks

In Fig. 1 we report results for similarities and analogies with embedding size 300. For similarities we use: ws353 (Finkelstein et al., 2001),

[1] Proposition 3 in Volpi and Malagò (2019) provides conditions under which Fisher Information matrix is isotropic, i.e., proportional to the identity.

Table 1: Spearman correlations for similarities tasks. WG5 inside the enwiki and geb section are the wikigiga5 pretrained vectors on 6B words (Pennington et al., 2014) tested for comparison on the dictionary of the smaller corpora enwiki and geb. Lastly, U and U+V are the standard methods with the word embeddings vectors. PM are the accuracies reported by Pennington et al. (2014) on enwiki, BDK is the best setup across tasks (varying hyperparameters) reported by Baroni et al. (2014) and LGD are the best methods in cross-validation with fixed window size of 10 and 5 (for varying hyperparameters) reported by Levy et al. (2015).

	method	ws353	mc	rg	scws	ws353sim	ws353rel	men	mturk287	rw	simlex999	all
enwiki	LE-U+V-ud-F	**75.5**	**83.4**	**81.5**	**63.5**	**77.8**	**69.2**	75.6	60.1	**55.6**	**41.6**	**62.6**
	WG5-U+V	65.1	73.8	77.6	62.2	71.3	60.7	**77.2**	65.7	51.5	41.0	61.3
	U	60.2	69.3	69.8	58.3	67.1	56.4	69.2	67.2	47.1	31.4	53.6
	U+V	63.8	74.5	75.2	58.7	69.5	60.9	71.6	**67.3**	45.5	32.2	55.1
geb	LE-U+V-ud-F	**77.0**	**81.2**	**83.5**	**65.0**	**80.3**	**68.7**	**79.6**	62.4	**59.3**	**46.9**	**65.2**
	WG5-U+V	65.1	73.8	77.9	61.8	71.3	60.7	77.2	65.7	53.2	40.6	60.4
	U	61.3	73.0	76.3	58.7	68.6	54.0	68.7	**68.1**	48.9	30.6	51.9
	U+V	64.9	77.4	79.9	59.1	71.5	58.8	71.4	**68.1**	48.5	32.5	53.7
PM 6B		65.8	72.7	77.8	53.9	-	-	-	-	38.1	-	-
BDK		73	-	83	-	78	68	80	-	-	-	-
LGD win5		-	-	-	-	74.5	61.7	74.6	63.1	41.6	38.9	-
LGD win10		-	-	-	-	74.6	64.3	75.4	61.6	26.6	37.5	-

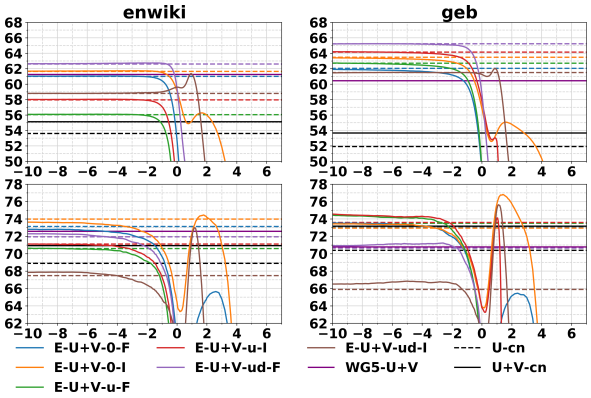

Figure 1: Word similarities (top) and word analogies (bottom) for different values of α.

mc (Miller and Charles, 1991), rg (Rubenstein and Goodenough, 1965), scws (Huang et al., 2012), men (Bruni et al., 2014), mturk287 (Radinsky et al., 2011), rw (Luong et al., 2013) and simlex999 (Hill et al., 2015). For analogies we use the Google analogy dataset (Mikolov et al., 2013a). The limit embeddings (colored dotted lines) achieve good performances on both tasks, above the competitor methods from the literature U and U+V centered and normalized by column, as described in Pennington et al. (2014). Comparison with baseline methods from literature on word similarity is presented in Tables 1, we compare with the limit embeddings since they usually seem to be the best performing on the similarity task, see Fig. 1 top row. The limit embedding methods reported in the table outperform Wiki Giga 5 pretrained vectors (Pennington et al., 2014) (6B words corpus) and other comparable baselines from the literature with similar window size. In Table 2 we report

Table 2: Analogy tasks for the different methods on enwiki and geb. The best alpha is selected with a 3-fold cross validation (α between -10 and 10), unless the limit embedding is the best performing. PM are the accuracies reported by Pennington et al. (2014) on enwiki, BDK is the best setup across tasks (varying hyperparameters) reported by Baroni et al. (2014).

	method	sem	syn	tot
enwiki	E-U+V-0-I	**84.5** $\pm$ 0.4	67.33 $\pm$ 0.6	**74.4** $\pm$ 0.1
	WG5-U+V	79.4	**67.5**	72.6
	U	77.8	62.1	68.9
	U+V	80.9	63.4	70.9
geb	E-U+V-0-I	**83.8** $\pm$ 0.4	**72.2** $\pm$ 0.4	**76.7** $\pm$ 0.3
	WG5-U+V	78.7	65.2	70.7
	U	75.7	66.8	70.4
	U+V	80.0	68.5	73.2
PM 1.6B		80.8	61.5	70.3
PM 6B		77.4	67.0	71.7
BDK		80.0	68.5	73.2

best performances on analogy task on alpha embeddings, where alpha is selected with cross-validation (Table 3). For enwiki syn, the limit embedding has been found to work better instead. The errors reported are obtained averaging the performances on test of the top three alpha selected based on best performances on validation. The errors obtained are relatively small which indicates that tuning alpha is easy also on tasks with small amount of data in cross-validation. The best tuned alpha on the geb dataset completely outperform the baselines.

The last intrinsic tasks considered are cluster purity for concept categorization datasets AP (Al-

muhareb, 2006) and BLESS (Baroni and Lenci, 2011). The purity curves (Fig. 2) are more noisy, this is because the datasets available for this task are quite limited in size. Almost all the curves exhibit a peak which is relatively more pronounced for smaller embedding sizes, while the limit behaviour for very negative alphas is better performing for larger embedding size. This points to the fact that the natural clustering performed by the limit embeddings of Eq. 9 is better behaved when the dimension of the embedding grows. Increasing the embedding size, increases the number of sufficient statistics, thus allowing more flexibility for the limit clustering during training.

Table 3: Best cross-validated alphas for methods of Table 2 (enwiki and geb).

	method	sem	syn	tot
en	E-U+V-0-I	1.8 ± 0.1	$-\infty$	1.7 ± 0.1
geb	E-U+V-0-I	1.7 ± 0.1	1.3 ± 0.1	1.3 ± 0.1

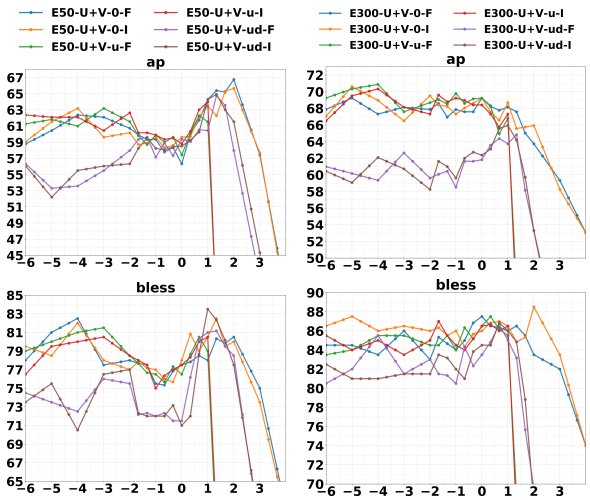

Figure 2: Cluster purity on concept categorization task.

3.2 Extrinsic Tasks

As extrinsic tasks we choose 20 Newsgroup multi classification (Lang, 1995) and IMDBReviews sentiment analysis (Maas et al., 2011). Embeddings are normalized before training either with I or F. We use a linear architecture (BatchNorm+Dense) for both tasks, while for sentiment analysis we also use a recurrent architecture (Bidirectional LSTM 32 channels, GlobalMaxPool1D, Dense 20 + Dropout 0.05, Dense). In Tables 4 and 5 we report the best methods chosen with respect to the validation set and the best limit embedding performances for embedding size 300. A more complete set of experiments can be found in Appendix. Limit Embeddings have been generalized, instead of considering only the max row χ^* (see Sec. 2.2), by considered the top k rows from ΔV. Limit embeddings are evaluated with respect to top 1, 3, and 5, denoted -t1/3/5. Furthermore we denote by -w if a weighted average (with weights $p_w(\chi)/p_0(\chi)$) is performed for the top rows of ΔV. The improvements reported in the Tables are small but consistent, of above 0.5% accuracy on both Newsgroups and IMDBReviews, furthermore the improvement persist also with increased complexity of the network architecture (bidirectional LSTM). Fig. 3 reports curves for the values on test with early stopping based on validation for embedding sizes of 50 and 300. The improvements for tuning alpha are higher on size 50 exhibiting a more evident peak. For size 300 improvements are smaller but consistent. In particular a peak performance for alpha can be always easily identified for a chosen reference distribution and a chosen normalization.

Table 4: AUC and accuracy on test of 20 Newsgroups multiclass classification, compared to baseline vectors. Best alpha and best limit method (on validation) are reported in parenthesis.

method	20 Newsgroups	
	AUC	acc
U+V	96.34	65.06
E-U+V-0-F	96.76 (0.2)	65.86 (0.4)
E-U+V-u-F	96.79 (0.2)	66.30 (0.2)
E-U+V-ud-F	96.79 (0.4)	65.24 (0.6)
LE-U+V-0-F	96.65 (t3-w)	64.47 (t1)
LE-U+V-u-F	96.65 (t3-w)	64.54 (t1)
LE-U+V-ud-F	96.38 (t5-w)	64.76 (t3-w)

Table 5: Accuracy on test of IMDBReviews sentiment analysis binary classification, with linear and with BiLSTM architecture, compared to baseline vectors. Best alpha and best limit method (on validation), are reported in parenthesis.

method	IMDB Reviews	
	acc lin	acc BiLSTM
U+V	83.76	88.00
E-U+V-0-F	83.58 (2.4)	88.12 (−4.0)
E-U+V-u-F	83.72 (−3.0)	88.56 (−4.0)
E-U+V-ud-F	84.23 (−3.0)	88.48 (−2.2)
LE-U+V-0-F	84.00 (t1)	88.36 (t1)
LE-U+V-u-F	84.29 (t1)	88.66 (t1)
LE-U+V-ud-F	84.00 (t3-w)	88.49 (t3-w)

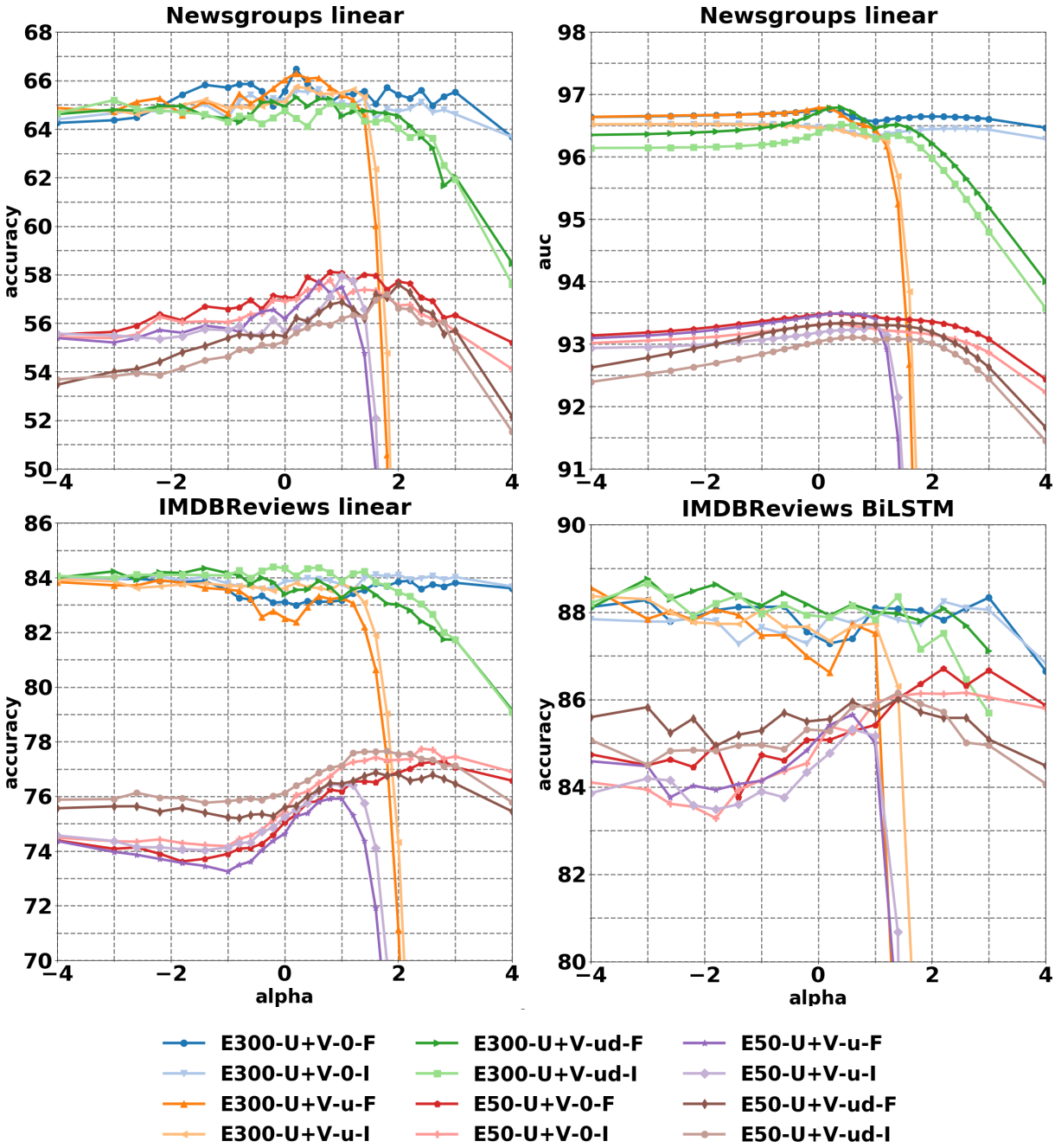

Figure 3: Performances on 20 Newsgroups and IMDB Reviews for varying alphas. Metrics I and F refers to embeddings normalization before training.

4 Conclusions

For word similarities and analogies alpha embeddings provide significant improvements over baseline methods (corresponding to $\alpha = 1$). For the other tasks the improvements are smaller but consistent, depending on the value of α, the chosen reference distribution (0, u, ud) and the chosen normalization method (I, F). The improvements persist also when increasing the complexity of the networks used (linear vs BiLSTM). This motivates further studies on more complex architectures, for example on models employing transformers with the aim to close the experimental gap with the state of the art.

The best value of alpha depends both on the task and on the dataset. Alpha embeddings thus provide an extra handle on the optimization problem, allowing to choose the deformation parameter based on data. Alpha values lower than 1 and negative seems to be preferred across most tasks. Limit embeddings provide a simple method which does not require validation over alpha, but can still offer an improvement on several tasks of interest. Furthermore limit embeddings can be interpreted as a natural clustering in space learned by the SG model itself during training. Performances of the limit embeddings grow with increasing dimension, pointing to the possibility to have a consistent improvement in higher embedding dimensions without tuning alpha.

Acknowledgments

R. Volpi and L. Malagò are supported by the DeepRiemann project, co-funded by the European Regional Development Fund and the Romanian Government through the Competitiveness Operational Programme 2014-2020, Action 1.1.4, project ID P_37_714, contract no. 136/27.09.2016.

References

Abdulrahman Almuhareb. 2006. *Attributes in lexical acquisition*. Ph.D. thesis, University of Essex.

Shun-ichi Amari. 1985. *Differential-geometrical methods in statistics*, volume 28 of *Lecture Notes in Statistics*. Springer-Verlag, New York.

Shun-ichi Amari. 2016. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer Japan, Tokyo.

Shun-ichi Amari and Hiroshi Nagaoka. 2000. *Methods of information geometry*. American Mathematical Society, Providence, RI. Translated from the 1993 Japanese original by Daishi Harada.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. Rand-walk: A latent variable model approach to word embeddings. *arXiv preprint arXiv:1502.03520*.

Attardi. Wikiextractor: A tool for extracting plain text from wikipedia dumps. https://github.com/attardi/wikiextractor. Accessed: 2017-10.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't 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: Long Papers)*, pages 238–247.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

BookCorpus. Aligning books and movie: Towards story-like visual explanations by watching movies and reading books. https://yknzhu.wixsite.com/mbweb. Accessed: 2019-09.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of artificial intelligence research*, 49:1–47.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

Gutenberg. Free ebooks - project gutenberg. https://www.gutenberg.org. Accessed: 2019-09.

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

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Sosuke Kobayashi. Homemade bookcorpus. https://github.com/soskek/bookcorpus. Accessed: 2019.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

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.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

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

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

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.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *NAACL-HLT*, pages 746–751.

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

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. *arXiv:1702.01417 [cs, stat]*. ArXiv: 1702.01417.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv:1802.05365 [cs]*. ArXiv: 1802.05365.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346.

Vikas Raunak. 2017. Simple and Effective Dimensionality Reduction for Word Embeddings. *arXiv:1708.03629 [cs]*. ArXiv: 1708.03629.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.

Amari Shun-Ichi and Nagaoka Hiroshi. 2000. *(Translations of mathematical monographs 191) Methods of information geometry*. American Mathematical Society.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal. Association for Computational Linguistics.

Riccardo Volpi and Luigi Malagò. 2019. Natural alpha embeddings. ArXiv:1912.02280.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ArXiv*, abs/1906.08237.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Additional Details

We have performed experiments using two corpora: english Wikipedia dump October 2017 (enwiki) and also we augmented this last one with Guthenberg(Gutenberg) and BookCorpus(BookCorpus; Kobayashi) calling this geb (guthenberg, enwiki, bookcorpus). We used the wikiextractor python script(Attardi) to parse the Wikipedia dump xml file. A minimal preprocessing have been used: lower case all the letters, remove stop-words and remove punctuation. We use a cut-off minimum frequency ($m0$) of 1000 during GloVe training (Pennington et al., 2014). We obtained a dictionary of about 67k words for both enwiki and geb. The window size was set to be 10 as in (Pennington et al., 2014), with decaying weighting rate from the center of $1/d$ for the calculation of cooccurrences. We trained the models for a maximum of 1000 epochs. Embedding sizes used are 50 and 300.

Table 6: AUC on Newsgroups with linear architecture (BatchNorm + Dense). We use geb embeddings, fixed during the classifiers training. The alpha for which to report performances on test is chosen based on the best measure on the validation set and we report both performances on validation and on test (α between -4 and 4 with adaptive step: 0.2 between [-1, 1] and 0.4 in between [-3, 3] and 1 between [-4, 4]). We also report limit embedding performances.

method	AUC val	AUC test
E-U+V-0-I ($\alpha = 1.0$)	0.96347	0.96342
E-U+V-0-F ($\alpha = 0.2$)	0.96765	0.9676
E-U+V-u-F ($\alpha = 0.2$)	0.96792	0.96787
E-U+V-ud-F ($\alpha = 0.4$)	0.96798	0.96792
LE-U+V-0-F-t3-w	0.9666	0.96654
LE-U+V-u-F-t3-w	0.96662	0.96655
LE-U+V-ud-F-t5-w	0.96388	0.96381

Table 7: Accuracy on Newsgroups (BatchNorm + Dense).

method	accuracy val	accuracy test
E-U+V-0-I ($\alpha = 1.0$)	0.66846	0.65056
E-U+V-0-F ($\alpha = 0.4$)	0.67708	0.65858
E-U+V-u-F ($\alpha = 0.2$)	0.68068	0.66298
E-U+V-ud-F ($\alpha = 0.6$)	0.67744	0.65242
LE-U+V-0-F-t1	0.66739	0.64472
LE-U+V-u-F-t1	0.66954	0.64545
LE-U+V-ud-F-t3-w	0.6602	0.64763

Table 8: Accuracy on IMDBReviews with linear architecture (BatchNorm + Dense).

method	accuracy val	accuracy test
E-U+V-0-I ($\alpha = 1.0$)	0.83426	0.83758
E-U+V-0-F ($\alpha = 2.4$)	0.83574	0.83582
E-U+V-u-F ($\alpha = -3.0$)	0.83434	0.83721
E-U+V-ud-F ($\alpha = -3.0$)	0.8360	0.8423
LE-U+V-0-F-t1-f	0.8351	0.84001
LE-U+V-u-F-t1-wf	0.83724	0.84293
LE-U+V-ud-F-t3-wf	0.83493	0.84001

Table 9: Accuracy on IMDBReviews with BiLSTM-pool architecture (Bidirectional LSTM 32 channels, GlobalMaxPool1D, Dense 20 + Dropout 0.05, Dense).

method	accuracy val	accuracy test
E-U+V-0-I ($\alpha = 1.0$)	0.87813	0.88002
E-U+V-0-F ($\alpha = -4.0$)	0.88066	0.88117
E-U+V-u-F ($\alpha = -4.0$)	0.88173	0.88565
E-U+V-ud-F ($\alpha = -2.2$)	0.88366	0.88481
LE-U+V-0-F-t1	0.88258	0.88365
LE-U+V-u-F-t1	0.87761	0.88656
LE-U+V-ud-F-t1	0.88117	0.8825

Table 10: Spearman correlations for similarities tasks for the different methods on enwiki and geb. LE represents the cos product between limit embeddings on the exponential family model. WG5 inside the enwiki and geb section are the wikigiga5 pretrained vectors on 6B words (Pennington et al., 2014) tested for comparison on the dictionary of the smaller corpora enwiki and geb. Lastly, U and U+V are the non-geometric methods with the word embeddings vectors.

	method	ws353	mc	rg	scws	ws353sim	ws353rel	men	mturk287	rw	simlex999	all
enwiki	LE-U+V-0-F	70.7	77.2	77.3	64.0	75.7	66.6	74.7	**68.7**	54.2	37.7	61.0
	LE-U+V-0-I	72.1	82.7	81.3	**64.2**	**76.5**	67.1	74.8	65.9	54.8	40.0	61.7
	LE-U+V-u-F	69.6	77.1	77.5	63.6	74.7	65.2	74.5	**69.1**	54.1	36.7	60.5
	LE-U+V-u-I	**72.5**	81.9	**81.7**	**64.3**	76.7	**67.8**	**75.6**	67.7	**55.9**	39.1	**62.1**
	LE-U+V-ud-F	**75.5**	**83.4**	**81.5**	63.5	**77.8**	**69.2**	**75.6**	60.1	**55.6**	**41.6**	**62.6**
	LE-U+V-ud-I	68.6	**82.9**	78.9	59.3	73.6	57.2	71.3	50.3	53.8	**41.6**	58.8
	WG5-U+V	65.1	73.8	77.6	62.2	71.3	60.7	**77.2**	65.7	51.5	**41.0**	61.3
	U	60.2	69.3	69.8	58.3	67.1	56.4	69.2	67.2	47.1	31.4	53.6
	U+V	63.8	74.5	75.2	58.7	69.5	60.9	71.6	67.3	45.5	32.2	55.1
geb	LE-U+V-0-F	72.9	80.5	83.9	65.4	78.6	66.3	77.2	**70.7**	57.6	39.6	62.0
	LE-U+V-0-I	74.3	**82.2**	**84.6**	**66.0**	79.3	67.1	78.0	67.3	58.6	43.4	63.5
	LE-U+V-u-F	74.1	81.4	**84.6**	**65.8**	79.9	67.5	78.2	**70.4**	57.7	40.4	62.7
	LE-U+V-u-I	**75.7**	**82.1**	**84.8**	**66.0**	**80.5**	**68.2**	**79.2**	67.0	**58.8**	**44.1**	**64.1**
	LE-U+V-ud-F	**77.0**	81.2	83.5	65.0	**80.3**	**68.7**	**79.6**	62.4	**59.3**	46.9	**65.2**
	LE-U+V-ud-I	71.5	78.2	79.9	60.9	76.8	58.9	74.7	52.4	57.2	**48.1**	61.5
	WG5-U+V	65.1	73.8	77.9	61.8	71.3	60.7	77.2	65.7	53.2	40.6	60.4
	U	61.3	73.0	76.3	58.7	68.6	54.0	68.7	68.1	48.9	30.6	51.9
	U+V	64.9	77.4	79.9	59.1	71.5	58.8	71.4	68.1	48.5	32.5	53.7

Table 11: Analogy tasks for the different methods on enwiki and geb. The best alpha is selected with a 3-fold cross validation (α between -10 and 10). The methods reported are implementing either euclidean normalization (I) or normalization with the Fisher (F) in different points on the manifold $(0, u)$. Scalar products (-p) are always calculated with respect to the Identity in this table (I).

corpus	method	semantic		syntactic		total	
		alpha	acc	alpha	acc	alpha	acc
enwiki 1.5B	E-U+V-0-nF-pI	1.7 ± 0.1	$\mathbf{85.7} \pm 0.3$	-9.5 ± 0.5	65.9 ± 0.4	-9.5 ± 0.5	73.6 ± 0.4
	E-U+V-0-nI-pI	1.8 ± 0.0	$\mathbf{84.6} \pm 0.4$	-2.2 ± 5.5	66.6 ± 0.3	1.7 ± 0.1	$\mathbf{74.4} \pm 0.1$
	E-U+V-u-nF-pI	-7.2 ± 3.3	81.8 ± 0.2	-9.5 ± 0.7	65.7 ± 0.5	-9.5 ± 0.7	72.7 ± 0.4
	E-U+V-u-nI-pI	-8.5 ± 0.3	82.3 ± 0.4	-9.1 ± 1.2	67.1 ± 0.4	-8.5 ± 1.1	73.6 ± 0.4
	LE-U+V-0-nF-pI	$-\infty$	83.4	$-\infty$	66.9	$-\infty$	**74.0**
	LE-U+V-0-nI-pI	$-\infty$	82.8	$-\infty$	**67.3**	$-\infty$	**74.0**
	LE-U+V-u-nF-pI	$-\infty$	81.6	$-\infty$	66.2	$-\infty$	72.8
	LE-U+V-u-nI-pI	$-\infty$	82.0	$-\infty$	**67.5**	$-\infty$	73.7
	WG5-U+V	n/a	79.4	n/a	**67.5**	n/a	72.6
	U	n/a	77.8	n/a	62.1	n/a	68.9
	U+V	n/a	80.9	n/a	63.4	n/a	70.9
geb 1.8B	E-U+V-0-nF-pI	1.9 ± 0.2	$\mathbf{84.6} \pm 0.3$	-8.5 ± 2.0	68.1 ± 0.2	-9.9 ± 0.1	73.8 ± 0.3
	E-U+V-0-nI-pI	1.7 ± 0.1	83.8 ± 0.4	1.3 ± 0.1	$\mathbf{72.2} \pm 0.4$	1.3 ± 0.1	$\mathbf{76.7} \pm 0.3$
	E-U+V-u-nF-pI	-9.1 ± 1.3	80.0 ± 0.2	-9.7 ± 0.4	69.7 ± 0.4	-9.7 ± 0.4	73.9 ± 0.3
	E-U+V-u-nI-pI	1.0 ± 0.0	81.8 ± 0.3	-2.1 ± 4.4	70.3 ± 0.8	1.0 ± 0.0	75.2 ± 0.2
	LE-U+V-0-nF-pI	$-\infty$	82.1	$-\infty$	67.1	$-\infty$	73.2
	LE-U+V-0-nI-pI	$-\infty$	81.2	$-\infty$	67.3	$-\infty$	72.9
	LE-U+V-u-nF-pI	$-\infty$	80.1	$-\infty$	68.0	$-\infty$	72.9
	LE-U+V-u-nI-pI	$-\infty$	80.9	$-\infty$	68.5	$-\infty$	73.5
	WG5-U+V	n/a	78.7	n/a	65.2	n/a	70.7
	U	n/a	75.7	n/a	66.8	n/a	70.4
	U+V	n/a	80.0	n/a	68.5	n/a	73.2