Self-StrAE at SemEval-2024 Task 1: Making Self-Structuring AutoEncoders Learn More With Less

Mattia Opper [a] and **N. Siddharth** [a,b]
[a] University of Edinburgh; [b] The Alan Turing Institute
{m.opper,n.siddharth}@ed.ac.uk

Abstract

This paper presents two simple improvements to the Self-Structuring AutoEncoder (Self-StrAE). Firstly, we show that including reconstruction to the vocabulary as an auxiliary objective improves representation quality. Secondly, we demonstrate that increasing the number of independent channels leads to significant improvements in embedding quality, while simultaneously reducing the number of parameters. Surprisingly, we demonstrate that this trend can be followed to the extreme, even to point of reducing the total number of non-embedding parameters to seven. Our system can be pre-trained from scratch with as little as 10M tokens of input data, and proves effective across English, Spanish and Afrikaans.

1 Introduction

Natural language is generally understood to be compositional. To understand a sentence, all you need to know are the meanings of the words and how they fit together. The mode of combination is generally conceived as an explicitly structured hierarchical process which can be described through, for example, a parse tree. Recent work by Opper et al. (2023) presents the Self-StrAE (Self-Structuring AutoEncoder), a model which learns embeddings such that they define their own hierarchical structure and extend to multiple levels (i.e. from the subword to the sentence level and beyond). The strengths of this model lie in its parameter and data efficiency achieved through the inductive bias towards hierarchy.

Learning embeddings such that they meaningfully represent semantics is crucial for many modern NLP applications. For example, retrieval augmented generation (Lewis et al., 2020) is predicated on the fact that the correct contexts for a given query can be determined. The semantic relation between a query and a context is encompassed by the notion of semantic relatedness. They are not equivalent to one another (i.e. paraphrases), but are close in meaning in a broader, more contextual sense. The focus of task one of this year's SemEval (Ousidhoum et al., 2024a,b) is capturing this notion of semantic relatedness, with a particular focus on African and Asian languages generally characterised by a lack of NLP resources.

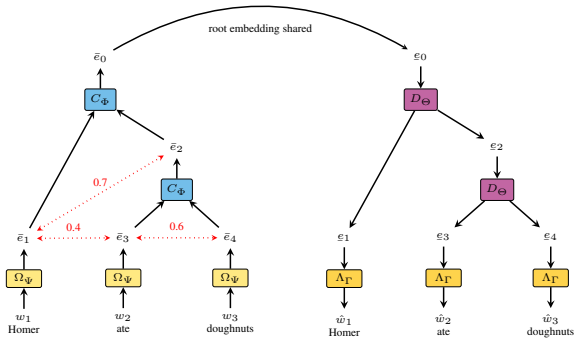

Figure 1: Self-StrAE forward pass. Red lines indicate cosine similarity between adjacent nodes. Shared colours indicate shared parameters.

In this work, we investigate whether Self-StrAE can learn embeddings which capture semantic relatedness, when trained from scratch on moderately sized pre-training corpora. We turn to the competition in order to examine whether the model can even compare with dedicated STR systems. In order to determine whether Self-StrAE can provide an alternative approach in low resource settings where systems that rely on large pre-trained transformers (Vaswani et al., 2017) may not have sufficient scale to prove effective. We show that with two simple changes, Self-StrAE's performance can be substantially improved. Moreover, we demonstrate that the the resulting system is not limited to English, but can work equally well (if not better) for both Spanish and Afrikaans [1].

[1] Code available at: https://github.com/mopper97/Self-StrAE

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 108–115

2 Model and Objectives

2.1 Model

The core architecture at the heart of this paper is the Self-StrAE. A model that processes a given sentence to generate both multi-level embeddings and a structure over the input. The forward pass begins by first *embedding* tokens to form an initial frontier, using the embedding matrix Ω_Ψ. This is followed by iterative application of the following update rule:

1. Take the cosine similarity between adjacent embeddings in the frontier.
2. Pop the most similar pair.
3. Merge the pair into a single parent representation, and insert into the frontier.
4. If len(frontier) = 1, stop

Merge is handled by the recursively applied *composition function* C_Φ, which takes the embeddings of two children and produces that of the parent. The process is illustrated in 1. In the figure, the highest cosine similarity is between the embeddings of 'ate' and 'doughnuts', so these two embeddings are merged first. At the next step, 'Homer' and 'ate doughnuts' are merged as they have the highest similarity of the remaining embeddings. At this point the frontier has shrunk to a single embedding and the root has been reached.

If we consider the merge history at the root, we can see that it has come to define a tree structure over the input. This structure is passed to the decoder, which then generates a second set of embeddings, starting from the root and proceeding to the leaves. The decoder achieves this through recursive application of the *decomposition function* D_Θ, which takes the embedding of a parent and produces the embeddings of the two children. Once the decoder reaches the leaves, it can optionally output discrete tokens through use of a *dembedding function* Λ_Γ.

We denote embeddings produced during composition as $\bar{e}$ and produced during decomposition as $\underline{e}$. For a vocabulary of size *V*, each embedding $e \in \mathbb{R}^E$ consists of *k* independent channels of size *u*. With this notation established, we can now define the four core components of a Self-StrAE.

Embedding:
$\Omega_\Psi(w_i) = w_i\Psi$, where $\Psi \in \mathbb{R}^{V\times E}$

Composition:
$C_\Phi(\bar{e}_{c1}, \bar{e}_{c2}) = hcat(\bar{e}_{c1}, \bar{e}_{c2})\Phi + \phi$
where $\Phi \in \mathbb{R}^{2u\times u}$ and $\phi \in \mathbb{R}^u$

Decomposition:
$D_\Theta(\underline{e}_p) = hsplit(\underline{e}_p\Theta + \theta)$
where $\Theta \in \mathbb{R}^{u\times 2u}$ and $\theta \in \mathbb{R}^{2u}$

Dembedding:
$\Lambda_\Gamma(\underline{e}_i) = \underline{e}_i\Gamma$ where $\Gamma \in \mathbb{R}^{E\times V}$

Note that in the above the dembedding layer is treated as a separate parameter matrix to the embedding layer, however, it can just as easily be weight tied to increase efficiency.

2.2 Objectives

There are a few options for pre-training Self-StrAE. The simplest solution is to have the model reconstruct the leaf tokens, which can be achieved by simply employing cross entropy over the output of the dembedding layer. For a given sentence $s_j = \langle w_i \rangle_{i=1}^{T_j}$, this objective is formulated as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{T_j}\sum_{i=1}^{T_j} w_i \cdot \log \hat{w}_i. \tag{1}$$

An alternative approach adopted by Opper et al. (2023) is to use contrastive loss as the reconstruction objective. For a given batch of sentences s_j, the total number of nodes (internal + leaves) in the associated structure is denoted as M. This allows for the construction of a pairwise similarity matrix $A \in \mathbb{R}^{M\times M}$ between normalised upward embeddings $\langle \bar{e}_i \rangle_{i=1}^M$ and normalised downward embeddings $\langle \underline{e}_i \rangle_{i=1}^M$, using the cosine similarity metric (where embeddings are flattened to be of shape *E*). Denoting $A_{i\bullet}, A_{\bullet j}, A_{ij}$ the i^{th} row, j^{th} column, and $(i,j)^{\text{th}}$ entry of a matrix respectively, the objective is defined as:

$$\mathcal{L}_{\text{cont}} = \frac{-1}{2M}\left[\sum_{i=1}^{M}\log \sigma_\tau(A_{i\bullet}) + \sum_{j=1}^{M}\log \sigma_\tau(A_{\bullet j})\right] \tag{2}$$

where $\sigma_\tau(\cdot)$ is the tempered `softmax` (temperature τ), normalising over the unspecified ($\bullet$) dimension.

A final option is to combine these two objectives, applying the cross entropy reconstruction over leaves and the contrastive objective over all other nodes, where constructing a vocabulary is intractable due to the number of possible combinations. The contrastive objective remains identical except that A is now defined as pairwise similarity matrix $A \in \mathbb{R}^{I \times I}$, where I is the number of internal nodes of the structure. In its simplest form, this objective, which we will henceforth refer to as CECO, can then be defined as:

$$\mathcal{L}_{\text{CECO}} = \frac{1}{2}(\mathcal{L}_{CE} + \mathcal{L}_{cont}) \quad (3)$$

3 Experiments

3.1 Setup

For all experiments, we utilise a pre-training set of ≈10 million tokens. We make this choice because Self-StrAE is intended to be data efficient, especially if it is to be useful for low resource languages where scale may not be available. For English the data was sourced from a subset of Wikipedia, while for Afrikaans and Spanish we obtained corpora from Leipzig Corpora Collection[2]. We utilise a pre-trained BPE tokenizer for each language from the BPEMB Python package (Heinzerling and Strube, 2018). Though the package also provides pre-trained embeddings, we solely use the tokenizer and learn embeddings from scratch.

During the course of model development, we utilised additional evaluation sets as a further guide. For English, we used Simlex (Hill et al., 2015) and Wordsim353 (Agirre et al., 2009) as measures of how well the model captures lexical semantics, and STS-12 (Agirre et al., 2012), STS-16 (Agirre et al., 2016) and STS-B (Cer et al., 2017). For Afrikaans, due to lack of resources, we utilised a Dutch translation of STS-B (Huertas-García et al., 2021) as the two languages are closely related. For Spanish, we utilised a Spanish translation of STS-B from the same source, as well as the labelled train and dev sets from SemRel 2024 (Ousidhoum et al., 2024a). While these sets contain labels, we apply the model fully unsupervised and solely use them for zeroshot evaluation.

We train Self-StrAE for 15 epochs using the Adam optimizer at a learning rate of 1e-3 (Kingma and Ba, 2015). We set the embedding dimension to 256, with a batch size of 512 and τ of 1.2. We conducted our primary experiments on English and then applied the same system design to Spanish and Afrikaans.

[2]For both Spanish and Afrikaans we selected the mixed corpus and took a uniform subsample to reduce size to the requisite scale.

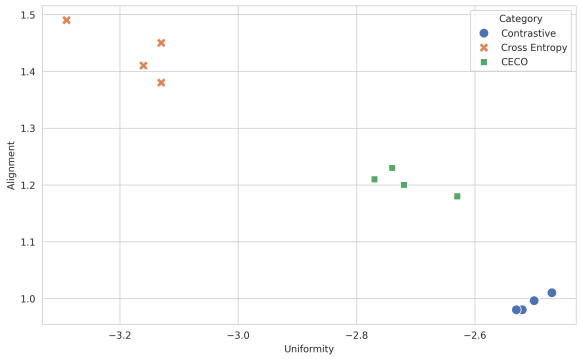

Figure 2: Uniformity and Alignment plot for contrastive, cross entropy and CECO pre-training objectives. Results taken across four random seeds. Lower is better for both measures.

3.2 Which Objective is Best?

The first thing we want to establish is which objective is most suitable for training Self-StrAE, as the original version only utilises contrastive loss. For parity with the original implementation, we treat the embeddings as square matrices (i.e. $k = u$) in this experiment.

Figure 2 show the uniformity and alignment analysis (Wang and Isola, 2020) of the representations learned by the different objectives. Uniformity describes the extent to which embeddings are spread around the space, while alignment characterises how similar positive target pairs are to each other. To be successful, representations should optimise both properties. We can observe that while the cross entropy objective leads to uniformity, it is comparatively poor at optimising alignment. This essentially implies that the decoder embeddings deviate from those of the encoder. Alignment is clearly a desireable property, as the results in table 1 show. The contrastive loss leads to both better sentence level representations and to more stable performance.

However, the best setting of all is CECO (the combination of cross entropy and contrastive). There are two factors worth considering that may explain this finding. Firstly, including reconstruction of discrete labels inherently provides additional meaningful information compared to just organising the representations alone. Secondly, at the token level the contrastive loss is most sus-

Objective	Simlex	Wordsim S	Wordsim R	STS-12	STS-16	STS-B	SemRel (Dev)
Contrastive	13.80 ± 0.41	54.33 ± 0.78	52.40 ± 0.87	31.93 ± 1.03	52.48 ± 0.44	40.05 ± 2.01	50.13 ± 0.88
CE	13.77 ± 9.43	46.43 ± 24.00	51.23 ± 23.04	17.68 ± 4.88	25.40 ± 15.60	22.43 ± 15.12	32.95 ± 14.93
CECO	**19.15 ± 2.39**	**58.33 ± 3.31**	**62.65 ± 2.76**	**41.20 ± 4.04**	**58.40 ± 1.35**	**48.35 ± 1.36**	**54.40 ± 0.81**

Table 1: Comparison of Objective Performance. Results are taken across four random intialisations. Models are trained on English.

k	u	Simlex	Wordsim S	Wordsim R	STS-12	STS-16	STS-B	SemRel (Dev)	# Params
8	32	17.50 ± 2.12	58.45 ± 1.04	62.10 ± 2.29	31.00 ± 2.67	52.53 ± 3.33	41.90 ± 2.09	49.30 ± 0.59	4192
32	8	17.28 ± 5.94	44.83 ± 27.11	49.10 ± 25.47	33.28 ± 17.49	46.75 ± 30.85	41.35 ± 25.57	43.95 ± 30.50	280
64	4	16.15 ± 9.82	48.63 ± 20.95	51.30 ± 23.05	38.88 ± 22.39	49.48 ± 31.05	43.05 ± 28.91	46.13 ± 30.35	88
128	2	17.33 ± 7.12	52.85 ± 19.33	55.15 ± 19.85	39.63 ± 20.83	50.38 ± 31.92	46.63 ± 27.95	47.78 ± 30.92	22
256	1	12.00 ± 12.84	42.80 ± 23.35	45.05 ± 24.58	29.18 ± 24.68	39.65 ± 32.22	37.35 ± 29.55	40.63 ± 29.07	7
8	32	19.4	59.4	64.3	27.6	56	44.5	50.1	4192
32	8	21.6	57.2	61.6	44.3	63.3	54.1	58.8	280
64	4	**21.7**	62.8	66.1	49.9	65.6	57.4	61.3	88
128	2	18.4	**65.1**	**67.2**	49	**67.2**	60.9	63.2	22
256	1	20.7	63.2	66.3	**50.1**	66.2	**61.6**	**63.6**	7

Table 2: Impact of number of independent channels on performance. Results are taken across four random initialisations. Models are trained on English. Top half of the table represents average performance, the bottom half contains the best performing initialisation. # Params is the number of non-embedding parameters.

ceptible to noise (e.g. the word 'the' may occur frequently in the batch, but each repeated instance will be treated as a false negative), and under such conditions the objective has been shown to lead to feature suppression (Robinson et al., 2021).

Summary: We find that combining cross entropy and contrastive loss leads to better representations than applying each objective individually, and consequently use this approach going forward.

3.3 How many channels?

Each embedding in Self-StrAE is treated as consisting of k independent channels of size u. This is intended to allow the representations to capture different senses of meaning. However, in the original paper the number of channels is set to be the square root of u, and not explored further. Consequently, we wanted to see what the optimal balance between the number of channels and their size was. Results are shown in 2. Surprisingly, we found that as the number of channels increased (and consequently u decreased) performance improves quite dramatically, even to the limit of treating each value in the embedding as independent. Furthermore, because the number of non-embedding parameters (i.e. the composition and decomposition functions) is directly tied to the channel size u, *decreasing model complexity improves embedding quality*.

However, it should be noted that this decrease in complexity comes with a tradeoff in terms of reliability. The smaller the size of the channel, the more variance we observed between random initialisations, with some initialisations failing to learn any meaningful representations whatsoever. We have found a solution that is able to maintain performance and ensure stability between seeds, but we leave discussion of this to the appendix, as we do not yet have a clear picture of what exactly is causing instability and wish to avoid speculation. We do however wish to emphasise that the problem is tractable and there is ample scope for further development, and direct the interested reader to A for more information.

Summary: Increasing the number of channels while decreasing their size leads to significant improvements in performance, though at the cost of some instability between seeds. For our submisson to SemRel we used the setting $k = 128$, $u = 2$ as this allowed for an acceptable failure rate while not compromising performance (roughly 1 in 4 seeds fail). Consequently, our system utilises only 22 non-embedding parameters.

3.4 Performance Across Languages

So far our experiments have only considered English. We now examine whether the framework is language agonistic, and pre-train Self-StrAE on both Spanish and Afrikaans. As before we pretrain on a small scale data (described in 3.1).

Results are in 3. We can see that the improvements to Self-StrAE hold across different languages and are not the result of some quirk in our English pre-training set. In fact performance is either comparable or better than on English. The

Language	NL STS-B (Dev)	NL STS-B (Test)	Afr SemRel (Dev)	Afr SemRel (Test)	Competition Rank
Afrikaans	52.8	64.5	23.4	76.5	2
Language	ESP STS-B (Test)	ESP SemRel (Train)	ESP SemRel (Dev)	ESP SemRel (Test)	Competition Rank
Spanish	61.5	58.5	68.7	63.5	6

Table 3: Self-StrAE Performance on Spanish and Afrikaans. Results correspond to those of the submitted systems, which we selected using the best run from four random initialisations.

results on Afrikaans are particularly interesting as the model performs significantly better on this language. Whether this is due to how the test set was created or to underlying features of the language provides an interesting question for future work. Moreover, the Afrikaans model, despite never having been trained on Dutch, is able to generalise fairly well to it, shown by the results on the translated STS-B sets.

4 Related Work

Recursive Neural Networks: Self-StrAE belongs to the class of recursive neural networks first popularised by (Socher et al., 2011, 2013). Recursive neural networks are extremely similar to recurrent neural networks, they differ because they process inputs hierarchically rather than sequentially (e.g. going up a parse tree).

Learning Structure and Representations: Recursive neural networks require structure as input. An alternative approach is to train a model that learns structure and the network at the same time. Recent unsupervised examples include Drozdov et al. (2019, 2020); Hu et al. (2021). However, these mechanisms generally use search to determine structure making them highly memory intensive. Self-StrAE differs from these as it asks the representations to define their own structure, making it much more resource efficient, though less flexible in certain aspects.

Contrastive Loss: Contrastive loss is an objective which optimises the representation space directly. In broad terms this objective requires the representations of a positive pair to be as similar to each other as possible, while minimising similarity to a set of negative examples. The closest examples of this objective, for the approach employed in this paper, are Chen et al. (2020); Shi et al. (2020); Radford et al. (2021).

5 Conclusion

We show that two simple changes can make Self-StrAE significantly more performant: adding a discrete reconstruction objective and increasing the number of independent channels. The latter also has the added benefit of reducing the number of parameters in the model, and surprisingly means that simpler is better. More broadly, we believe these findings demonstrate the potential of an inductive biases towards explicit structure. Self-StrAE, at present, is a very simple model. The only thing it really has going for it is the inductive bias which tasks embeddings with organising themselves hierarchically. While the gap between Self-StrAE and SoTA systems still remains, the fact that it is able to perform at all demonstrates the promise. Moreover, the fact that the two simple changes demonstrated in this paper can lead to such improvements indicates that the full potential of the inductive bias has yet to be reached, and it is likely that further refinements can lead to even more substantial benefits. Finally, because this model does not require significant scale to optimise pursuing further improvements may provide substantial benefits for low resource languages where pre-training data is scarce.

6 Limitations

The results in this paper represent steps towards an improved model rather than a complete picture. We still do not fully understand what causes the instability in training when the number of channels increased, and though we can provide a solution (see A), further analysis is needed. The performance of contrastive loss can depend quite heavily on how positive and negative examples are defined and it is likely that the explanation rests there. Secondly, while we have shown that Self-StrAE can be applied to languages other than English the results are limited to Indo-European languages. An interesting avenue for future work would be investigating a broader spectrum of languages, and whether specific characteristics can be identified which influence how well the model performs.

7 Acknowledgements

MO was funded by a PhD studentship through Huawei-Edinburgh Research Lab Project 10410153. We thank Victor Prokhorov, Ivan Vegner and Vivek Iyer for their valuable comments and helpful suggestions during the creation of this work.

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 *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 19–27.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, page 385–393, USA. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Workshop on Semantic Evaluation (SemEval)*, pages 1–14.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Learning Representations (ICLR)*, volume 119, pages 1597–1607.

Andrew Drozdov, Subendhu Rongali, Yi-Pei Chen, Tim O'Gorman, Mohit Iyyer, and Andrew McCallum. 2020. Unsupervised parsing with S-DIORA: Single tree encoding for deep inside-outside recursive autoencoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4832–4845, Online. Association for Computational Linguistics.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. 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 1129–1141, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Xiang Hu, Haitao Mi, Zujie Wen, Yafang Wang, Yi Su, Jing Zheng, and Gerard de Melo. 2021. R2D2: Recursive transformer based on differentiable tree for interpretable hierarchical language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4897–4908, Online. Association for Computational Linguistics.

Álvaro Huertas-García, Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2021. Countering misinformation through semantic-aware multilingual models. In *Intelligent Data Engineering and Automated Learning – IDEAL 2021*, pages 312–323, Cham. Springer International Publishing.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Mattia Opper, Victor Prokhorov, and Siddharth N. 2023. StrAE: Autoencoding for pre-trained embeddings using explicit structure. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7544–7560, Singapore. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said

Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. 2021. Can contrastive learning avoid shortcut solutions? *CoRR*, abs/2106.11230.

Yuge Shi, Brooks Paige, Philip Torr, and N. Siddharth. 2020. Relating by contrasting: A data-efficient framework for multimodal generative models. In *International Conference on Learning Representations (ICLR)*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–161.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *CoRR*, abs/2005.10242.

A Stabilising High Channel Self-StrAE

One solution we have found to the instability issue is modifying the objective. This formulation, loosely inspired by SimCSE (Gao et al., 2021), runs the same input through the model twice, with different dropout masks applied each time. The objective is cross entropy reconstruction for the leaves, and contrastive loss between the two different sets of decoder embeddings for the non-terminals. Currently we have two theories as to why this might work:

- Better negatives: because the decoder embeddings represent the contextualised meaning of node rather than it's local one, the issue of false negatives is somewhat mitigated.
- Encoder consistency: because we ask the two sets of decoder embeddings to be similar to each other the encoder is encouraged to produce the same structure regardless of dropout mask. It may be that this pressure towards regularity leads to the improved consistency.

Results are shown in 4. For lack of a better term we refer to this alternative objective as StrCSE. In its current form we do not consider this objective to be well formed, and solely provide it here as a possible starting point for further research.

Objective	Simlex	Wordsim S	Wordsim R	STS-12	STS-16	STS-B	SemRel (Dev)
Contrastive	13.80 ± 0.41	54.33 ± 0.78	52.40 ± 0.87	31.93 ± 1.03	52.48 ± 0.44	40.05 ± 2.01	50.13 ± 0.88
CE	13.77 ± 9.43	46.43 ± 24.00	51.23 ± 23.04	17.68 ± 4.88	25.40 ± 15.60	22.43 ± 15.12	32.95 ± 14.93
CECO	**19.15 ± 2.39**	**58.33 ± 3.31**	**62.65 ± 2.76**	41.20 ± 4.04	58.40 ± 1.35	48.35 ± 1.36	54.40 ± 0.81
CECO k=128 u=2	17.33 ± 7.12	52.85 ± 19.33	55.15 ± 19.85	39.63 ± 20.83	50.38 ± 31.92	46.63 ± 27.95	47.78 ± 30.92
StrCSE k=128 u=2	**21.68 ± 1.88**	**59.06 ± 2.38**	**64.08 ± 0.91**	**49.46 ± 0.59**	**66.18 ± 0.24**	**61.30 ± 0.76**	**62.88 ± 0.42**

Table 4: StrCSE compared with other objectives. Results are taken over four random intialisations. Training data is English.