Banyan: Improved Representation Learning with Explicit Structure

Mattia Opper [1] **N. Siddharth** [1]

Abstract

We present Banyan, a model that efficiently learns semantic representations by leveraging explicit hierarchical structure. While transformers excel at scale, they struggle in low-resource settings. Conversely recent structured models have shown promise as efficient learners, but lack performance. Banyan bridges this gap with two key innovations: an entangled hierarchical tree structure and diagonalised message passing, enabling it to outperform larger transformer models with just 14 non-embedding parameters. It excels in low-resource settings, offering a viable alternative for under-represented languages and highlighting its potential for efficient, interpretable NLP in resource-constrained environments.

1. Introduction

Semantic representations are foundational for various NLP applications, such as retrieval-augmented generation (RAG) (Lewis et al., 2020), question answering, and summarisation (Abdalla et al., 2023; Wang et al., 2022). They are also crucial for clustering and organising textual data when labelled training data is unavailable. Typically, such representations are generated by large-scale transformer models (Vaswani et al., 2017); highly effective but needing substantial amounts of data and computational resources to train.

An alternative approach draws inspiration from linguistics and cognitive science, incorporating structured compositions—a principle that posits that understanding the semantics of a whole requires knowing the meanings of its parts and the structural rules that determine how they assemble (Chomsky, 1956; Crain & Nakayama, 1987; Pallier et al., 2011; de Marneffe et al., 2006). This principle is highly efficient because novel utterances can be decomposed into familiar components using systematic rules, minimising the need to store individual meanings. It allows humans to learn efficiently from relatively little data and enables effective and efficient learners (Lake et al., 2016; Ito et al., 2022; Wiedemer et al., 2023).

To incorporate such inductive biases into models, the traditional information flow within neural networks needs altering. Instead of relying on implicit processing alone, models must learn representations for atomic components and an explicit computation graph that dictates how these components combine. Additionally, models must learn functions to govern information flow through this graph. Such approaches have demonstrated improved language modelling perplexity at cognitively plausible scales (Hu et al., 2021; 2022), better systematic generalisation (Sartran et al., 2022; Murty et al., 2023), and, especially relevant here, enhanced efficiency in acquiring semantics (Opper et al., 2023b).

The SELF-STRAE model (Opper et al., 2023b) learns representations that explicitly model compositional semantics, and achieves promising performance while requiring minimal resources, both in terms of data and model size. It opened the door to exploring more compute-efficient solutions—particularly valuable for low-resource languages where scaling is often infeasible. However, while innovative, it still falls short compared to large-scale pre-trained transformers, even in languages outside standard pre-training corpora. Here, we introduce BANYAN, a model which significantly outperforms SELF-STRAE while achieving greater resource efficiency. Our approach involves modifying the structural optimisation process to induce an *entangled* graph that models global relations between nodes and employs a message passing mechanism using diagonal functions, reducing parameters while enhancing expressiveness.

BANYAN, achieves performance comparable to transformer-based baselines and represents a low-cost, viable alternative to transformers for producing representations in low-resource languages, as measured by semantic textual similarity (STS) tasks. By leveraging cognitively inspired inductive biases, our work enables semantic representation learning that rivals or surpasses large-scale pre-trained LLMs—using only 14 non-embedding parameters. Our model, BANYAN, offers a new direction for efficient and effective semantic understanding in resource-constrained environments. [1]

[1]School of Informatics, University of Edinburgh, UK. Correspondence to: Mattia Opper <m.opper@ed.ac.uk>, N. Siddharth <n.siddharth@ed.ac.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]Code available at: github.com/exlab-research/Banyan

2. Background and Related Work

Banyan is a graph neural network, specifically a recursive neural network (RvNN) that learns both structure and representations. We unpack these components below.

Recursive Neural Networks (RvNNs): Like regular recurrent neural networks (RNNs), RvNNs process data by repeatedly applying a function to update their state in sequence. However, instead of relying on temporal ordering (like the sequence of words in a sentence), RvNNs use hierarchical structures, often provided as input—most commonly as a binary tree—and can be applied either bottom-up (from leaves to root) or top-down (root to leaves). They were popularised in the deep learning era by Socher et al. (2011; 2013), inspiring many successor models that vary in how they define the recursive function, including Tree-LSTMs (Tai et al., 2015) and IORNN (Le & Zuidema, 2014).

Learning Structure: RvNNs often require structural input, which limits their flexibility since this structure may not always be available or easily obtainable. To address this, researchers have developed methods to induce structure within the model during recursive computation; using differentiable chart parsing (Drozdov et al., 2019; 2020; Hu et al., 2021; 2022), beam search (Ray Chowdhury & Caragea, 2023), continuous relaxation (Chowdhury & Caragea, 2021; Soulos et al., 2024), and reinforcement learning (Havrylov et al., 2019). However, these methods can struggle with memory issues and require careful tuning of hyperparameters. Here, we adopt a method from Opper et al. (2023b) that uses representation similarity to determine how nodes should be merged during computation, which is both computationally efficient and surprisingly effective.

Semantic Representations of Text: Systems like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) use the distributional hypothesis (Harris, 1954) to model word semantics, which posits that words are defined by the context in which they appear. To learn representations, these models use a fixed context window and predict a missing word in a sequence. While initially effective, this approach is limited because representations for higher level objects (i.e. phrases, sentences etc.) are computed by simply averaging word embeddings. However, some notable follow on works attempted to improve upon this. Arora et al. (2017); Ethayarajh (2018) introduce a more sophisticated form of taking an average over word embeddings by using SVD, while Rücklé et al. (2018) use the power mean. These approaches yielded improvements, but relied on representations pre-trained at scale which limited their applicability to specialised domains or low resource languages. Wieting et al. (2021) attempt to refine the sentence representation through the use of paraphrase corpora, looking to increase alignment between language pairs, again improving performance, but requiring large scale parallel corpora. Finally, Pagliardini et al. (2017) realised that if the average of the embeddings was going to be used to create the sentence representation, it makes sense to optimise it directly. Consequently, they modified the pre-training objective in order to have the average predict a missing word from a sentence. At scale this proved tremendously effective. However, despite offering substantial improvements, all these methods require scale and more importantly do not tackle the central limitation of word embeddings - the inability to handle changes in meaning dependent on context.

On the other hand, transformers, through self-attention, are able to represent contextualised meanings. However, early encoder-only transformer models produced poor representations (Reimers & Gurevych, 2019), especially compared to the more sophisticated approaches based on word embeddings. This was largely due to the anisotropy issue (Godey et al., 2024), which required the development of techniques using contrastive fine-tuning (Gao et al., 2021) to finally remedy. These approaches eventually surpassed word embeddings, and have become the method of choice for producing semantic representations. However, they still rely heavily on scale for success, as contrastive refinement is a final fine-tuning step applied to pre-trained models rather than directly incorporated within pre-training.

Semantic Representation Learning through Structure: Transformer embeddings have become more successful than static word embeddings due to their ability to handle varying contextual influences. Unlike attention mechanisms in transformers, which route information based on token relationships, some approaches use explicit graphs or structures, such as dependency (Levy & Goldberg, 2014; Vashishth et al., 2019) or constituency parses (Pham et al., 2015), to determine the focus of context windows. These models have the potential to bridge the gap between the efficiency of word embeddings and the contextualisation offered by transformers. This is because the discrete structure provides an input specific routing order which dictates interactions between atoms and consequently determines their influence on higher level representations - allowing for more flexibility than simple averaging. Most related to our work, Opper et al. (2023b) introduce two models. StrAE, which use constituency parsers to learn sentence-level embeddings alongside word embeddings, and SELF-STRAE, which learns its own structure using representations. This latter model, SELF-STRAE, serves as the foundation for BANYAN and is described next.

3. Preliminary: Self-StrAE

SELF-STRAE involves three main components that act over a sequence of tokens $\mathbf{w} = \langle w_n \rangle_{n=1}^{N}$: (a) an algorithm for merging tokens based on their similarity, (b) functions for composition and decomposition of embeddings, and (c) an

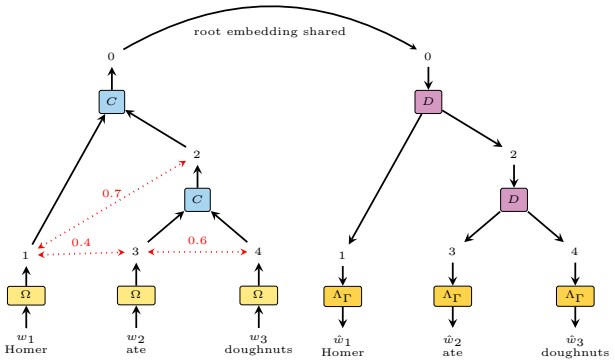


Figure 1: Self-StrAE operation. Red lines indicate cosine similarity. Shared colours imply shared parameters.

objective that leverages both the induced structure and embeddings. While full details are available in Opper et al. (2023b), we provide a brief overview to establish context for our model development (§ 4).

At a high level, SELF-STRAE learns representations that define their own structure while being shaped by it. Starting with an initial embedding matrix Ω_Ψ, tokens are merged into single embeddings using a composition function C_Φ based on best cosine similarity (e.g., see Figure 1). This process reduces the sequence to a single root embedding while capturing semantic relationships. The resulting merge history forms a binary tree structure, over which the model then operates in reverse by decomposing embeddings at each node using a decomposition function D_Θ, to reconstruct the leaf embeddings. Optionally, it can further predict tokens ($\widehat{w}_n$) using a dembedding function Λ_Γ. Figure 1 illustrates the autoencoding process. During training, tokens that are frequently merged together develop correlated representations, leading the model to learn meaningful compositional semantics. This results in embeddings that reflect both their own structure and the semantic patterns they encode.

More formally, one denotes tokens as the vertices $w_i \in \Delta^V$ in a V-simplex for vocabulary size V, and note that the models generates two sets of embeddings—one going up ($\bar{e}$: leaves $\to$ root) and one coming down ($\underline{e}$: root $\to$ leaves). The embeddings are viewed as $e \in \mathbb{R}^{U\times K}$ allowing the composition and decomposition functions to act independently over K channels, and be defined as

$$C_\Phi(\bar{e}_i, \bar{e}_{i+1}) = \textsc{hcat}(\bar{e}_i, \bar{e}_{i+1})\,\Phi + \phi, \quad \Phi \in \mathbb{R}^{2U\times U} \quad (1)$$

$$D_\Theta(\underline{e}_i) = \textsc{hsplit}(\underline{e}_i\,\Theta + \theta), \qquad \Theta \in \mathbb{R}^{U\times 2U} \quad (2)$$

To learn this model from data, Opper et al. (2023b) derive two objectives. The first is straightforward cross-entropy over reconstructed tokens, for sentence $\mathbf{w}$ and prediction $\widehat{\mathbf{w}}$ as $\mathcal{L}_{\text{CE}}(\mathbf{w}, \widehat{\mathbf{w}}) = -\frac{1}{N}\sum_{n=1}^{N} w_n \cdot \log \widehat{w}_n$. This however, places little constraint on the intermediate nodes in the hierarchical model. To address this, an alternate structural contrastive objective is formulated over a batch of sentences. As up and down trees are structurally identical (modulo edge reversal), it draws together an embedding and its dual on the other tree, while pushing away all other embeddings across the batch, using cosine similarity. Denoting pairwise similarity matrix $A \in \mathbb{R}^{M\times M}$ between up and down embeddings over M nodes in the batch, the objective is: $\mathcal{L}_{\text{CO}}(\bar{e}, \underline{e}) = \frac{-1}{2M}\sum_{i=1}^{M} \log\left(\sigma_\tau(A_{i\cdot})\,\sigma_\tau(A_{\cdot i})\right)$ with $\sigma_\tau(\cdot)$ the tempered softmax over the unspecified $(\cdot)$ dimension.

4. Banyan

Given their construction, the upward embeddings are always *locally-contextual*: only encapsulating the context of the span they cover. For example, in Figure 1, the upward embedding $\bar{e}$ for the span "ate doughnuts" is always the same regardless of context, no matter who did the eating. In contrast, downward embeddings are always *globally-contextual*: necessarily encapsulating surrounding context, being decomposed from embeddings of larger spans. In our example, this implies multiple downward embeddings $\underline{e}^y$, one for each $y \in \{\text{"Lisa"}, \text{"Homer"}, \dots\}$. Learning effective embeddings requires amortisation over these differences to ensure meaning resolves over all these contexts.

4.1. From trees to entangled trees

Algorithm 1 BANYAN: Entangled Compose

Input: Global frontier $\langle (s_n, e_n)\rangle_{n=1}^{N}$, compose ($\circ$), concat ($\diamond$), similarity $\textsc{csim}(e, e')$

$\mathcal{A} \leftarrow \langle (s_n, e_n)\rangle_{n=1}^{N}$ ▷ initialise frontier
$(\mathcal{V}, \mathcal{E}) \leftarrow (\varnothing, \varnothing)$ ▷ initialise graph
while $\exists i : s_i \diamond s_{i+1} \notin_s \mathcal{V}$ **do**
  $i^\star \leftarrow \arg\max_i \textsc{csim}(e_i, e_{i+1})$ ▷ locate closest pair
  $e_p = \circ(e_{i^\star}, e_{i^\star+1})$ ▷ compose
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{(s_{i^\star} \diamond s_{i^\star+1}, e_p)\}$
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{p \sim i^\star, p \sim (i^\star + 1)\}$
  $\mathcal{J} \leftarrow \{j : (s_j, s_{j+1}) = (s_{i^\star}, s_{i^\star+1})\}$
    ▷ locate all occurrences of this pair
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\forall_{j\in\mathcal{J}}\, \mathcal{A}_j, \mathcal{A}_{j+1}\}$
    ▷ delete occurrences from those locations
  $\mathcal{A} \leftarrow \mathcal{A} \cup_{\mathcal{J}} \{(s_{i^\star} \diamond s_{i^\star+1}, e_p)\}$
    ▷ insert composition into those locations

return: Graph $(\mathcal{V}, \mathcal{E})$

We wish to have composition embeddings amortise over all possible contexts, and simultaneously, all decompositions embeddings to resolve to the same thing. The representation of an entity "Lisa" should encapsulate everything she could possibly eat. Simultaneously, the average of everything she could eat we should get back to "Lisa". Self-StrAE does not explicitly model this behaviour in its structure. Decomposition embeddings of the same entity only interact

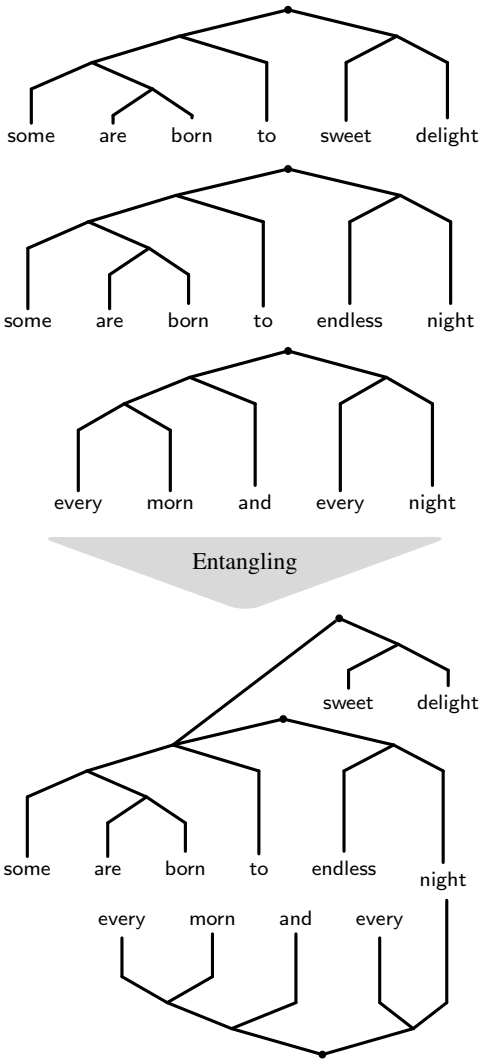
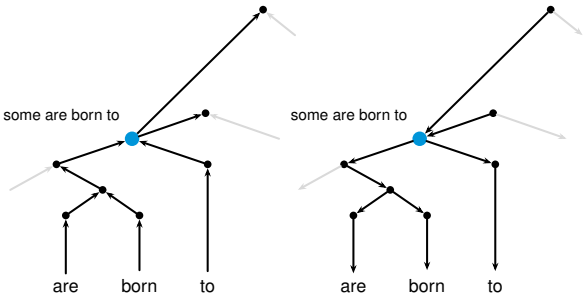


Figure 2: Entangled trees: Example of disjoint trees being transformed into an entangled tree. Internal functions ($C_\Phi, D_\Theta, \ldots$) are elided to avoid clutter.

when we calculate the loss. On top of this, because the loss is taken over the batch, they are actually treated as false negatives to each other. Even though they are terms that ought not be pushed away, the objective ask them to be.

Our innovation is to address both these issues by formulating the process in terms of entangled trees—where entangling describes the transformation of disjoint tree structures into a conjoined graph structure. An example is shown in Figure 2. Here, all instances of "night" and "some are born to" are captured by a single node representing that constituent. We call our model BANYAN on account of this entangling, because, like the tree, it can have many roots—consisting of nodes frequently reused across contexts.

Entangling: Constructing an entangled tree given a set of disjoint trees is a relatively straightforward process and is formally specified in Algorithm 1. In contrast to the agglomerative clustering employed in SELF-STRAE, here we employ a global frontier spanning all leaf nodes across the given data. The key differences to the prior methods are mainly to do with constructing a graph jointly with progressing the frontier and ensuring that new nodes are never duplicated, for which we employ a node identity s_n in addition to the node embedding e_n.



(a) upward composition (b) downward decomposition

Figure 3: Upward and downward traversals for a section of the entangled tree from Figure 2.

Incorporating context: Following the entangling of trees described, the model proceeds in a similar vein to SELF-STRAE, by composing upwards from leaves to roots (multiple roots corresponding to multiple trees), and then decomposing downwards back to the leaves. With entangled trees, while traversing upwards each node is always composed from the same two children, but on the way back down, things are different as each separate context for a given node provides a different downward embedding. This is shown in Figure 3 focussing on a subgraph of the entangled tree from Figure 2(right). Note that the node in question (in blue) corresponds to the span "some are born to", and has downward embeddings that incorporate context both from "endless night" and "sweet delight". This is exactly as desired, as BANYAN allows explicit aggregation to derive the downward embedding that resolves over the contexts. For any upward embedding $\bar{e}$ whose span occurs in different contexts $y \in \mathcal{Y}$, the corresponding downward embedding is derived by simply averaging over the different contextual down embeddings; i.e., $\underline{e} = 1/|\mathcal{Y}| \sum_y \underline{e}^y$.

Effectiveness and efficiency: Beyond the ability to explicitly incorporate context across data, entangled trees also help the contrastive objective by avoiding false negatives since they do not admit duplicate nodes by construction. Furthermore, the lack of duplicate nodes also drastically impacts the memory footprint of the model as one deals with the *set* of all nodes rather than counting each instance as its own node. These effects becomes more pronounced when entangling a larger set of instances as the likelihood of false negative and duplicates goes up together.

Practical estimation: Given the advantages of entangled trees, one would ideally want to construct it over *all* the

available data—not practically feasible with the exponential growth in dataset sizes. To address this, we construct our model to estimate the given objective by taking steps over *batches* of data that are of a more manageable size, noting that this estimator is unbiased. To see this is the case, note that entangled trees only affects the downward embeddings directly, and that batching simply means that the resolved embedding is an average over *samples* instead of over all the data (*population*)—the sample mean is always an unbiased estimator of the population mean.

4.2. Simplified Message Passing

Complementary to the development of entangled trees to incorporate context, we also explore avenues to improve the message passing with the composition (C) and decomposition (D) functions. The original formulations (1, 2) concatenate or split using simple single-layer linear neural networks. These were found to lead to better representations than e.g., Tree-LSTM cells, because they forced the model to conform to the compression order of the structure.

But if all that was required for success is to respect the compression order, then one could possibly do better with a simpler solution that exploits diagonalised functions (Ba et al., 2016)—a crucial component in the resurgence of recurrent neural networks (Peng et al., 2023; Orvieto et al., 2023; De et al., 2024) introducing decayed memory across time. Thus, rather than using linear layers, we now define:

$$C(\bar{e}_i, \bar{e}_{i+1}) = (\bar{e}_i \cdot \sigma(\Phi_l) + \bar{e}_{i+1} \cdot \sigma(\Phi_r)) + \phi \quad (3)$$

$$D(\underline{e}_i) = \left(\underline{e}_i \cdot \sigma(\Theta_l) + \theta_l,\ \ \underline{e}_i \cdot \sigma(\Theta_r) + \theta_r\right) \quad (4)$$

$$\Phi_l, \Phi_r, \phi, \Theta_l, \Theta_r, \theta_l, \theta_r \in \mathbb{R}^U$$

with sigmoid non-linearity (σ) applied to parameters both for numerical stability and to enforce a decayed memory over structure depth. Repeated application of the diagonal composition function will decay the influence of nodes further down in the tree, thereby respecting its compression order. During composition representations can increase in magnitude as they are the sum of the two children. During decomposition representations will, by necessity, reduce back down in magnitude towards the leaves. Further mimicking the information flow specified by the entangled trees. Finally, they restrict encoder (comp) and decoder (decomp) embeddings to remain in the same space. Which makes amortisation required for successful reconstruction, letting us switch objective to **cross entropy** over the vocabulary. We provide analysis to support this claim later in § 7.

These relatively simple changes have a pretty drastic effect, both in terms of performance (see experiments), as well as efficiency, with parameters now reduced by a factor of U compared to the functions from (1, 2).

5. Experiments: English Evaluation

Goal: We wish to test whether BANYAN can efficiently learn semantics. We start by evaluating on English, which is well resourced and has a wide array of test sets available with which we can measure the efficacy of our embeddings. This is crucial to establish, because when we turn to low resource languages later on, the amount of reliable evaluation sets will become limited. We want to make use of broad spectrum of tests available for English to reliably demonstrate embedding quality before moving forward.

5.1. Experimental Setup and Evaluation:

We want to evaluate how well BANYAN learns effective semantic representations. Ideally we want to probe this at different levels of hierarchy, because it allows us to test whether structured models can do what they are supposed to i.e., seamlessly transfer semantic knowledge across different levels of hierarchy via composition. Our evaluation is unsupervised, both to directly probe the effect of the inductive bias, and for greater parity with what may be expected in a low-resource domain. It consists of three parts:

Correlation with human judgements: We compare the cosine similarity of embedding pairs produced by the model with human judgements of their semantic correspondence. On the word level, we use Simlex-999 (Hill et al., 2015) and WordSim-S/R (Agirre et al., 2009). All tasks measure semantics, but do so on differing axes. Simlex measures similarity at the exclusion of relatedness. Wordsim S measures similarity without penalising relatedness. And Wordsim R measures relatedness. On the sentence level, we use STS-12 through 16 (Agirre et al., 2012; 2013; 2014; 2015; 2016), the STS-B (Cer et al., 2017), SICK-R (Marelli et al., 2014) and SemRel (Ousidhoum et al., 2024) - which combined cover a wide array of semantic correspondence.

Retrieval: This is a cornerstone of Retrieval-Augmented Generation (RAG)-based systems and perhaps the most important use case for embedding models. We use two retrieval datasets from the BEIR suite (Thakur et al., 2021). Quora: evaluates success of matching questions to answers and capturing the response relation. Arguana: evaluates matching arguments to counter arguments, testing if our semantic space captures the notion of dialectical opposition.

Classification: We also include two test sets from the GLUE benchmark (Wang et al., 2019). Sentiment classification (SST-2) tests whether the representation space captures semantic polarity. Paraphrase detection (MRPC): tests whether our representation space capture semantic equivalence. While our other evaluation is applied to the embeddings zero-shot, for the classification tasks we train a GeLU MLP with a 512D hidden size, though we leave models frozen as a direct test of representation quality.

Table 1: Sentence level results for models pretrained on English. Higher is better. Results represent mean and standard deviation across four random initialisations. Spearman's ρ is * 100 following convention.

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	SemRel	Score
SELF-STRAE	31.98 ± 0.58	53.88 ± 0.68	37.73 ± 0.70	55.23 ± 0.58	55.55 ± 0.47	39.53 ± 1.61	51.78 ± 0.29	50.05 ± 0.92	46.59 ± 0.43
GLOVE	31.61 ± 0.31	21.69 ± 0.12	27.37 ± 0.10	40.42 ± 0.09	29.27 ± 0.12	28.25 ± 0.08	50.20 ± 0.25	41.20 ± 0.43	33.75 ± 0.04
+ stopword rm	39.00 ± 0.57	41.61 ± 0.19	39.31 ± 0.18	51.06 ± 0.35	45.14 ± 0.14	48.40 ± 0.07	52.80 ± 0.04	42.37 ± 0.13	44.96 ± 0.10
Sent2Vec	38.14 ± 0.29	51.37 ± 0.48	48.64 ± 0.09	67.28 ± 0.02	56.26 ± 0.06	53.39 ± 0.11	**59.67 ± 0.02**	51.47 ± 0.03	53.28 ± 0.11
ROBERTA	42.77 ± 1.27	51.70 ± 1.30	45.67 ± 1.42	63.97 ± 0.81	59.60 ± 0.61	39.97 ± 0.95	52.93 ± 0.23	52.73 ± 0.58	51.08 ± 0.61
+ SimCSE	**50.63 ± 1.45**	62.23 ± 2.51	54.17 ± 2.10	68.77 ± 3.00	**66.67 ± 1.40**	53.53 ± 1.18	56.87 ± 1.16	59.27 ± 0.93	59.02 ± 1.45
BANYAN	**51.38 ± 0.15**	**69.60 ± 0.37**	**63.20 ± 0.28**	**73.08 ± 0.26**	**67.18 ± 0.56**	**61.90 ± 0.63**	55.23 ± 0.13	**61.88 ± 0.22**	**62.97 ± 0.03**

Table 2: Word level results analogous to Table 1.

Model	Simlex	Wordsim-S	Wordsim-R	Score
SELF-STRAE	13.80 ± 0.41	54.38 ± 0.78	52.85 ± 1.27	40.34 ± 0.66
GLOVE	27.47 ± 0.25	62.53 ± 0.42	51.00 ± 0.56	47.00 ± 0.38
Sent2Vec	**28.88 ± 0.42**	**68.32 ± 1.28**	54.49 ± 1.51	**50.56 ± 0.79**
ROBERTA	**29.23 ± 0.64**	61.97 ± 2.38	46.00 ± 2.13	45.73 ± 1.71
BANYAN	14.65 ± 2.90	63.23 ± 2.21	**67.73 ± 0.3**	**48.53 ± 1.33**

Baselines: We compare against the SELF-STRAE, GLOVE (Pennington et al., 2014), Sent2Vec (Pagliardini et al., 2017) and a ROBERTA (Liu et al., 2019) in the medium configuration from (Turc et al., 2019; Opper et al., 2023a). SELF-STRAE, the closest point of comparison to BANYAN, indicates where the current performance level of structured representation learning lies. GLOVE lets us compare to traditional static embeddings, and tests whether our model is learning anything more than just simple bag of word features. To obtain sentence embeddings, we report results using both the simple average of the word embeddings and the average with stopwords removed following (Reimers & Gurevych, 2019). The latter is generally stronger. An even more powerful variant is Sent2Vec, which directly optimises the averaged representation by using it to represent the context. This comparison measures the utility BANYAN's flexible and parametrised composition process, compared to an optimised average. Finally, for ROBERTA, we report results using both the standard model, and again after enhancing ROBERTA through an extra round of contrastive SIMCSE training (Gao et al., 2021), as a further non-lexical baseline. Our pooling strategy is mean. To produce static embeddings from ROBERTA to use in lexical evaluation, we follow Bommasani et al. (2020) and average the contextualised representations of all occurrences of the word in the training set. The ROBERTA is intended as a stronger baseline. It has significantly more parameters than BANYAN and can model meaning in context unlike GLOVE and Sent2Vec.

Hyperparameters and Pre-Training Details: For all models we set the embedding size to 256. For SELF-STRAE we use the configuration of (Opper et al., 2023b) and set embeddings as square matrices (i.e., K=16 and U=16). For BANYAN we set these values to K=128 and U=2, because the more independent channels we allowed the better the model seemed to perform. We refer the reader to Appendix A for ablations. We also note that because we can perform this reduction in channel size, the number of non-embedding parameters for BANYAN drops to just 14, as these are directly proportional to U. The configuration for RoBERTa medium is 8 layers, 8 attention heads, 2048 dimensional feed forward layers, and relative positional embeddings. For SELF-STRAE, BANYAN and Sent2Vec we pre-train on a uniform subsample of English Wikipedia consisting of circa 10 million tokens. This represents the lower end of how many tokens might be available in a low resource setting, and allows us to test whether these methods are efficient. Meanwhile for GLOVE and ROBERTA we pre-train on Wiki-103 (Merity et al., 2016), to ensure that they are not penalised by insufficient scale. Wiki-103 comprises circa 100 million tokens and therefore represents the very upper end of what might be available in a low resource setting. Further training details are in Appendix B.

5.2. Results:

Results are shown in Tables 1 to 3. On both the word level and sentence level BANYAN does much better than SELF-STRAE. We ablate the reasons for this in more detail later in the manuscript. Both models suffer on SimLex because they need to model both similarity and relatedness as the latter dictates merge (related concepts often compose together). However, the important thing to note is that the structured models effectively transfer the same performance from the word level to the sentence level. They can take advantage of composition, and transfer the meaning of the parts to understanding the meaning of the whole. The GLOVE baseline is good on the word level, but does not generalise to the sentence level as well as the transformer, even when we give it stopword removal. Similarly, Sent2Vec is extremely strong on the word level, and while more effective than GLOVE, neither approach *seamlessly* transfers semantic knowledge to different levels of complexity. BANYAN can, and is able to achieve comparable or better performance than the SimCSE ROBERTA despite being much smaller and exposed to 10x less pre-training data. This means we have a structured model that remains efficient and cheap, and also effective at representation learning.

Table 3: Sentence level results on retrieval and classification tasks for models pretrained on English.

	Quora				Arguana				SST-2	MRPC
Model	NDCG@1	NDCG@10	R@1	R@10	NDCG@1	NDCG@10	R@1	R@10	Acc	F1
Self-StrAE	32.88 ± 0.28	40.02 ± 4.94	29.59 ± 0.23	44.77 ± 0.28	09.96 ± 0.11	15.48 ± 0.13	09.96 ± 0.11	21.48 ± 0.22	74.67 ± 0.52	80.34 ± 0.42
GloVe	29.99 ± 0.14	35.71 ± 0.15	26.08 ± 0.15	43.17 ± 0.25	06.19 ± 0.19	12.77 ± 0.24	06.18 ± 0.19	24.68 ± 7.40	75.83 ± 0.62	81 ± 0
+ stopword rm	44.41 ± 0.13	52.54 ± 0.17	38.78 ± 0.15	62.15 ± 0.25	09.89 ± 0.19	20.27 ± 0.09	09.89 ± 0.19	33.00 ± 0.26	76.50 ± 1.08	81 ± 0
Sent2Vec	36.12 ± 0.21	43.26 ± 0.15	31.33 ± 0.21	52.38 ± 0.05	09.60 ± 0.31	23.24 ± 0.15	09.60 ± 0.31	39.73 ± 0.89	76.53 ± 0.98	81 ± 0
RoBERTa	43.26 ± 0.76	49.97 ± 0.72	37.67 ± 0.68	58.78 ± 0.78	08.18 ± 0.43	17.60 ± 0.36	08.18 ± 0.43	28.85 ± 0.94	75.68 ± 0.96	81 ± 0
+ SimCSE	51.79 ± 2.12	59.30 ± 2.10	45.09 ± 1.60	68.74 ± 2.01	10.06 ± 1.27	21.84 ± 2.23	10.06 ± 1.27	37.36 ± 2.16	75.97 ± 1.08	80.83 ± 0.24
Banyan	**57.74 ± 0.10**	**65.71 ± 0.14**	**50.14 ± 0.10**	**75.71 ± 0.14**	**12.42 ± 0.40**	**28.28 ± 0.17**	**12.42 ± 0.40**	**48.19 ± 0.18**	75.96 ± 0.57	79.48 ± 0.42

Table 4: Multilingual Results. BANYAN performance over four random seeds. Baselines marked † finetuned on supervised semantic similarity datasets. FT–unsupervised finetuning using masked language modelling on same corpora as BANYAN.

Model	Indonesian	Arabic	Telugu	Marathi	Hausa	Afrikaans	Spanish	Amharic	Hindi	Score
XLM-R	46.7	31.6	46.3	55.7	4.1	56.2	68.9	57.3	52.7	46.61
Llama-3.1 (8B)	**53.4**	31.1	65.6	63.4	6.1	65.4	66.7	64.1	**61.7**	53.06
Mistral Nemo	50.7	20.1	57	52.3	1.8	58.3	66.2	53.2	55.8	46.16
MiniLM-L12†	39	16.1	34.8	39.5	32.7	74.1	58.8	9.6	43.8	38.71
Paraphrase XLM-R†	46.1	**61**	58.1	**79.6**	22.5	76.8	71.7	64.6	52	59.16
XLM-R (FT)	47.9	33.6	68.8	75.1	14.6	72.6	**72.8**	59.6	57.6	55.84
BANYAN	41.90 ± 0.56	42.28 ± 1.57	**71.58 ± 1.24**	66.38 ± 0.84	**49.68 ± 0.75**	**79.35 ± 0.65**	60.88 ± 0.86	**66.40 ± 0.90**	**61.63 ± 0.43**	**60.01 ± 0.35**

6. Experiments: Multilingual Evaluation

Goal: We've established that BANYAN is an efficient learner. This implies potential use for languages that are not well covered by current NLP approaches. Here we test that.

6.1. Experimental Setup and Evaluation:

Tasks: Learning semantic representations for low-resource languages remains an ongoing challenge. A core problem is not just the lack of training data, but also the lack of evaluation datasets. Recent work by Ousidhoum et al. (2024) has sought to address this issue, providing semantic relatedness test sets for several low resource Asian and African languages, evaluated by comparing embedding similarity to human judgements. This means we can BANYAN's ability on Indonesian, Arabic, Telugu, Marathi, Hausa, Afrikaans, Spanish, Amharic and Hindi—covering a broad spectrum of resource extent. For example, Spanish is generally well represented, while Hausa is considerably less so.

Baselines: We select XLM-R (Conneau et al., 2019): a transformer encoder trained on 2TB of multilingual data. Llama 3.1 8B (Dubey et al., 2024): a decoder only LLM trained on 15 trillion tokens. Mistral Nemo 12B: a decoder only LLM designed with multi-lingual capacities in mind. In addition we also compare against two specialised embedding models from the sentence transformers range (Reimers & Gurevych, 2019): Mini-LM-L12-V2 and Paraphrase-XLM-R-Multilingual-V1. These are pre-trained encoders that have been finetuned on supervised datasets designed to produce high quality semantic representations. The baselines we select here are emblematic of the kind of models one might reach for in order to embed a corpus. For all baselines we use mean pooling following (Reimers & Gurevych, 2019). Finally, for parity we include an XLM-R baseline which is finetuned on the same corpora as BANYAN.

Banyan Pre-training and Hyperparameters: For Afrikaans, Spanish and Amharic we obtained corpora from Leipzig Corpora Collection (Goldhahn et al., 2012). For Amharic we utilised a MIT licenced pre-training set of 1 million sequences available at this link. Hausa data was sourced from Opus (Nygaard & Tiedemann, 2003). Each dataset consists of roughly 10 million tokens. We utilise a pre-trained BPE tokeniser for each language from the BPEMB Python package (Heinzerling & Strube, 2018). BANYAN's hyperparameters remain the same as before. For XLM-R we finetune for 100k steps with early stopping, using a linearly scheduled learning rate of 5e-5 with 10% of steps as warmup. XLM-R runs at batch size 128 across 4xA40 cards.

6.2. Results

See Table 4. In Spanish, a well-resourced language with high coverage, the transformer baselines almost all outperform BANYAN. However, as languages become lower resourced the picture changes, and BANYAN outperforms or is comparable to the baselines. This even includes the multilingual XLM-R that has undergone supervised training. While

Table 5: Number of non-embedding parameters.

Model	BANYAN	SELF-STRAE	ROBERTA (M)	All-MiniLM-L12-V2	XLM-R	Llama 3.1	Mistral Nemo
Params	14	1072	≈10M	≈21M	≈85M	≈8B	≈12B

Table 6: Ablations of modelling changes made for Banyan. Higher is better. Results averaged across four random initialisations. Bolded results indicate no standard deviation overlap. Spearman's ρ is * 100 following convention.

Model	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	SemRel	Score
Standard Trees	31.98 ± 0.58	53.88 ± 0.68	37.73 ± 0.70	55.23 ± 0.58	55.55 ± 0.47	39.53 ± 1.61	51.78 ± 0.29	50.05 ± 0.92	46.59 ± 0.43
+ diag functions	35.13 ± 0.33	56.05 ± 0.24	40.58 ± 0.05	58.83 ± 0.10	56.78 ± 0.21	44.10 ± 0.14	53.35 ± 0.17	52.65 ± 0.17	49.68 ± 0.06
++ CE loss	47.10 ± 1.04	61.85 ± 1.44	58.60 ± 1.34	70.45 ± 0.57	62.45 ± 0.70	59.50 ± 0.53	**59.00 ± 0.26**	60.33 ± 0.26	59.91 ± 0.54
Entangled Trees	38.98 ± 0.39	61.75 ± 0.14	43.65 ± 0.46	58.21 ± 0.41	55.29 ± 0.23	46.15 ± 0.71	53.93 ± 0.16	52.53 ± 0.09	51.31 ± 0.13
+ diag functions	44.15 ± 0.002	62.80 ± 0.002	48.30 ± 0.001	64.60 ± 0.002	60.30 ± 0.001	49.80 ± 0.002	55.14 ± 0.001	57.70 ± 0.001	55.23 ± 0.01
++ CE loss	**51.38 ± 0.15**	**69.60 ± 0.37**	**63.20 ± 0.28**	**73.08 ± 0.26**	**67.18 ± 0.56**	**61.90 ± 0.63**	55.23 ± 0.13	**61.88 ± 0.22**	**62.97 ± 0.03**

finetuning XLM-R improves performance the amount of benefit it provides is not uniform and is insufficient to prove viable in the very low resource cases. BANYAN is able to learn competitive representations consistently across languages, unsupervised and with very little data, providing a viable alternative for cheaply and efficient embeddings for low resource languages.

7. Improvements and Ablations

7.1. Efficiency

Other than its embedding matrix, BANYAN only has composition and decomposition functions. Diagonalising these makes them easier to compute and more lightweight than standard weight matrices, ($2U$ rather than $2U \times U$), achieving a further order of magnitude reduction in parameters compared with the already minimal SELF-STRAE. Table Table 5 shows the difference, including a comparison to the various baselines used throughout the paper. Despite its size BANYAN remains competitive.

Secondly, by exploiting entangled tree structure the number of nodes grows at a significantly reduced rate with batch size compared with standard sentential trees (see Figure 4). This is because the number of reused constituent nodes also grows as batch size increases, and entangled trees capture the set of all constituents, which consequently does not grow as drastically. In practical terms, because entangled trees requires fewer nodes, and each node requires two distinct embeddings ($\bar{e}$ and $\underline{e}$) to be held for it, reducing the number of nodes required leads to radical improvements in memory efficiency. Put together, these changes mean that we can train BANYAN very quickly as we can use large batches and its small number of parameters ensure quick convergence. On a single Nvidia A40 GPU with a batch size of 1024, Banyan trains from scratch in under 50 minutes, meaning that the total cost of pretraining a BANYAN model sits at around 30 cents[2]. Free-tier Google Colab users can achieve similar results in about two hours with a smaller batch size. Inference can also be performed on CPU on typical laptops, because the model is so small. Combined with its data efficiency, we believe this provides a promising alternative for low resource languages and communities.

[2]Cloud computing costs from: https://www.runpod.io/pricing

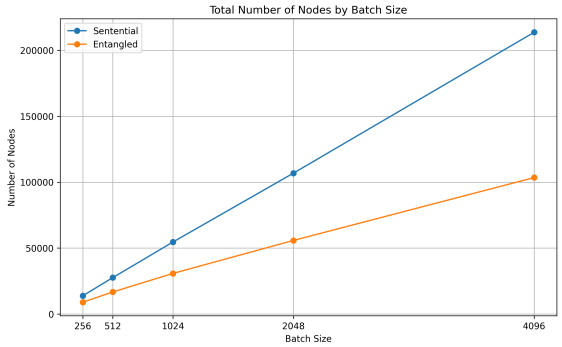
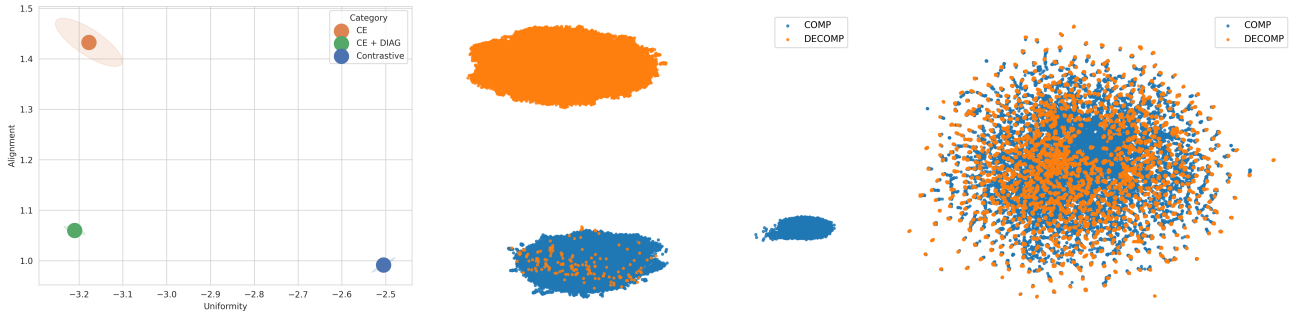


Figure 4: Total number of nodes in entangled vs. sentential trees as batch size grows.

7.2. Ablations

Why is BANYAN so much more effective than its SELF-STRAE predecessor? To probe the impact of each change, we perform ablations using the English STS tasks (Table 6).

Beyond improving efficiency, changing to entangled trees yields some benefits in terms of performance. The effect is significantly more pronounced when using the contrastive objective, as it removes the issue of false negatives as discussed in Section 4. However, it also yields some slight benefit with the CE leaf reconstruction objective. Entangling explicitly allows the model to take advantage of shared constituency structure between complex sequences, as it combines the information from all incoming parent messages. The slight edge this provides indicates that explicitly allowing the model to take advantage of such systematicity may be useful. However, in terms of performance, we find that the choice of functions and objective plays a much bigger role than entangling.



(a) Uniformity (↓) and Alignment loss (↓) analysis of BANYAN's representations using: CONTRASTIVE, CE and CE+DIAG.

(b) UMAP analysis of COMP and DECOMP embeddings for a 30k node entangled tree learned by CE w/o diagonal functions.

(c) UMAP analysis of COMP and DECOMP embeddings for a 30k node entangled tree learned by CE with diagonal functions.

Figure 5: Representation Analysis - UMAP Parameters in Appendix C.

Diagonal Functions: Perhaps the clearest benefits come from the introduction of the diagonalised composition and decomposition functions. These are bounded scalar values (sigmoid) multiplied with embeddings to mimic the time mixing blocks of SSMs (Gu & Dao, 2024). Hierarchically decaying in the influence of embeddings further down the structure through repeated application. This means that the representations are restricted to conform to the compression order it dictates, and we know from (Opper et al., 2023b) that the more we enforce this constraint the better our representations. Secondly, such simple message passing functions bias the representation space towards informative separability. There has to be some signal from which to perform reconstruction, and all the pressure is now on the representations. This is beneficial with the contrastive loss, but really shines when we combine them with cross entropy.

Changing Objective: Our more instructive finding is that cross entropy now outperforms the contrastive loss used by Opper et al. (2023b), contrary to the earlier result that the contrastive objective was critical. Figure 5 provides an analysis with Figure 5a showing uniformity vs. alignment metrics (↓) from Wang & Isola (2020). These measure (a) how evenly spread embeddings are (uniformity) and (b) the proximity of locally contextual composition embeddings to their globally contextualised counterparts (alignment). Success requires having low scores on both. We can see that the contrastive loss does well, while cross entropy achieves high uniformity, but not alignment without diagonal functions. This is further confirmed when we look at the UMAP (McInnes et al., 2018) analysis of BANYAN's embeddings with (5c) and without (5b) diagonal functions. Without diagonal functions, composition and decomposition embeddings largely occupy separate subspaces, failing to amortise over context. Introducing diagonal functions results in incredibly high overlap between the two, indicating effective amortisation. We believe that this is due to their simplicity and scaling constraints, restricting complex transformations during message passing. This *implicitly* forces representations to optimise the same useful qualities as the contrastive loss, without its propensity for shortcut solutions (Robinson et al., 2021). As a result, we can switch objective without compromising representation quality.

8. Conclusion, Limitations and Future Work

We introduce BANYAN, a Self-Structuring AutoEncoder. BANYAN's focus on global, entangled structure and simplified message passing exploits the benefits of structured compositions inherent in language data. It is more effective and efficient than prior work from which we draw three central conclusions.

Firstly, explicitly modelling structured compositions is an effective inductive bias. Table 5 shows the parameters for the structured models versus the baselines. The structured models are far smaller, with tens or thousands of parameters instead of millions or billions. And nonetheless, BANYAN is still competitive across several metrics, indicating we have found an efficient learning procedure.

Secondly, we have not yet fully exploited the potential of the inductive bias. BANYAN still relies on greedy agglomerative clustering to induce structure. This is effective, but sub-optimal. Future work could learn the structure induction procedure. The type of structure models are exposed to impacts the quality of learnt semantic representations (Opper et al., 2023b). So if *how* we induce structure improves, the model should learn better representations.

Finally, good and cheap embedding models are useful for many applications. For example, the digital humanities need to organise corpora of ancient languages, making it easier for researchers to access texts they need. But these corpora are small, and these languages are unlikely to be present in pretraining corpora of larger models. BANYAN provides an efficient solution for producing representations for both these use cases and low resource languages and under represented communities more generally. To conclude, Banyan addresses the problem of efficient learning in low-resource settings.

Acknowledgements

MO was funded by a PhD studentship through Huawei-Edinburgh Research Lab Project 10410153. We also wish to thank Henry Conklin, Seraphina Goldfarb-Tarrant, Vivek Iyer, Ivan Vegner, Sahil Verma, Su Kara, Ivan Titov and Edoardo Ponti for their valuable comments and feedback, throughout the research and development of this work. We'd also like to thank the ICML reviewers for their suggestions to help improve the paper. Finally, a special thank you goes to James Morrison, for helping to think through the ideas and providing invaluable feedback and edits to the manuscript.

Impact Statement

This paper represents an effort to enable resource efficient embedding techniques. We hope that our model makes a positive step towards making AI research and technologies more fair, equitable and accessible.

References

Abdalla, M., Vishnubhotla, K., and Mohammad, S. What makes sentences semantically related? a textual relatedness dataset and empirical study. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 782–796, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.55. URL https://aclanthology.org/2023.eacl-main.55.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. A study on similarity and relatedness using distributional and WordNet-based approaches. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 19–27, 2009.

Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. 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, pp. 385–393, USA, 2012. Association for Computational Linguistics.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. *SEM 2013 shared task: Semantic textual similarity. In Diab, M., Baldwin, T., and Baroni, M. (eds.), *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pp. 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/S13-1004.

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., and Wiebe, J. SemEval-2014 task 10: Multilingual semantic textual similarity. In Nakov, P. and Zesch, T. (eds.), *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2010. URL https://aclanthology.org/S14-2010.

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., and Wiebe, J. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In Nakov, P., Zesch, T., Cer, D., and Jurgens, D. (eds.), *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2045. URL https://aclanthology.org/S15-2045.

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., and Wiebe, J. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 497–511, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1081. URL https://aclanthology.org/S16-1081.

Arora, S., Liang, Y., and Ma, T. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SyK00v5xx.

Ba, J., Hinton, G., Mnih, V., Leibo, J. Z., and Ionescu, C. Using fast weights to attend to the recent past, 2016. URL https://arxiv.org/abs/1610.06258.

Bommasani, R., Davis, K., and Cardie, C. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4758–4781, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.431. URL https://aclanthology.org/2020.acl-main.431.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. SemEval-2017 task 1: Semantic textual similar-

ity multilingual and crosslingual focused evaluation. In *Workshop on Semantic Evaluation (SemEval)*, pp. 1–14, 2017.

Chomsky, N. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956. doi: 10.1109/TIT.1956.1056813.

Chowdhury, J. R. and Caragea, C. Modeling hierarchical structures with continuous recursive neural networks. *CoRR*, abs/2106.06038, 2021. URL https://arxiv.org/abs/2106.06038.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL http://arxiv.org/abs/1911.02116.

Crain, S. and Nakayama, M. Structure dependence in grammar formation. *Language*, 63(3):522–543, 1987.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., Desjardins, G., Doucet, A., Budden, D., Teh, Y. W., Pascanu, R., Freitas, N. D., and Gulcehre, C. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/440_pdf.pdf.

Drozdov, A., Verga, P., Yadav, M., Iyyer, M., and McCallum, A. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. *CoRR*, abs/1904.02142, 2019. URL http://arxiv.org/abs/1904.02142.

Drozdov, A., Rongali, S., Chen, Y.-P., O'Gorman, T., Iyyer, M., and McCallum, A. 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)*, pp. 4832–4845, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.392. URL https://aclanthology.org/2020.emnlp-main.392.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ethayarajh, K. Unsupervised random walk sentence embeddings: A strong but simple baseline. In Augenstein, I., Cao, K., He, H., Hill, F., Gella, S., Kiros, J., Mei, H., and Misra, D. (eds.), *Proceedings of the Third Workshop on Representation Learning for NLP*, pp. 91–100, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-3012. URL https://aclanthology.org/W18-3012/.

Gao, T., Yao, X., and Chen, D. SimCSE: Simple contrastive learning of sentence embeddings. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL https://aclanthology.org/2021.emnlp-main.552.

Godey, N., Clergerie, É., and Sagot, B. Anisotropy is inherent to self-attention in transformers. In Graham, Y. and Purver, M. (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35–48, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.3/.

Goldhahn, D., Eckart, T., and Quasthoff, U. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S. (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 759–765, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/327_Paper.pdf.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Harris, Z. S. Distributional structure. *Word*, 10:146–162, 1954.

Havrylov, S., Kruszewski, G., and Joulin, A. Cooperative learning of disjoint syntax and semantics. In Burstein, J., Doran, C., and Solorio, T. (eds.), *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)*, pp. 1118–1128, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1115. URL https://aclanthology.org/N19-1115.

Heinzerling, B. and Strube, M. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T. (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

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

Hu, X., Mi, H., Wen, Z., Wang, Y., Su, Y., Zheng, J., and de Melo, G. 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)*, pp. 4897–4908, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.379. URL https://aclanthology.org/2021.acl-long.379.

Hu, X., Mi, H., Li, L., and de Melo, G. Fast-R2D2: A pre-trained recursive neural network based on pruned CKY for grammar induction and text representation. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2809–2821, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.181. URL https://aclanthology.org/2022.emnlp-main.181.

Ito, T., Klinger, T., Schultz, D. H., Murray, J. D., Cole, M. W., and Rigotti, M. Compositional generalization through abstract representations in human and artificial neural networks, 2022. URL https://arxiv.org/abs/2209.07431.

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

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *CoRR*, abs/1604.00289, 2016. URL http://arxiv.org/abs/1604.00289.

Le, P. and Zuidema, W. Inside-outside semantics: A framework for neural models of semantic composition. In *NIPS 2014 Workshop on Deep Learning and Representation Learning*, 2014.

Levy, O. and Goldberg, Y. Dependency-based word embeddings. In *Association for Computational Linguistics (ACL)(Volume 2: Short Papers)*, pp. 302–308, 2014.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401, 2020. URL https://arxiv.org/abs/2005.11401.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

McInnes, L., Healy, J., Saul, N., and Grossberger, L. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016. URL http://arxiv.org/abs/1609.07843.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*, 2013.

Murty, S., Sharma, P., Andreas, J., and Manning, C. D. Pushdown layers: Encoding recursive structure in transformer language models, 2023. URL https://arxiv.org/abs/2310.19089.

Nygaard, L. and Tiedemann, J. Opus—an open source parallel corpus. In *Proceedings of the 13th Nordic Conference on Computational Linguistics*, 2003.

Opper, M. and Siddharth, N. Self-strae at semeval-2024 task 1: Making self-structuring autoencoders learn more with less. *arXiv preprint arXiv:2404.01860*, 2024.

Opper, M., Morrison, J., and Siddharth, N. On the effect of curriculum learning with developmental data for grammar acquisition. *arXiv preprint arXiv:2311.00128*, 2023a.

Opper, M., Prokhorov, V., and Siddharth, N. Strae: Autoencoding for pre-trained embeddings using explicit

structure. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7544–7560, 2023b.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences, 2023.

Ousidhoum, N., Muhammad, S. H., Abdalla, M., Abdulmumin, I., Ahmad, I. S., Ahuja, S., Aji, A. F., Araujo, V., Beloucif, M., De Kock, C., Hourrane, O., Shrivastava, M., Solorio, T., Surange, N., Vishnubhotla, K., Yimam, S. M., and Mohammad, S. M. 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, 2024.

Pagliardini, M., Gupta, P., and Jaggi, M. Unsupervised learning of sentence embeddings using compositional n-gram features. *CoRR*, abs/1703.02507, 2017. URL http://arxiv.org/abs/1703.02507.

Pallier, C., Devauchelle, A.-D., and Dehaene, S. Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, 108 (6):2522–2527, 2011. doi: 10.1073/pnas.1018711108.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., He, X., Hou, H., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., Mantri, K. S. I., Mom, F., Saito, A., Tang, X., Wang, B., Wind, J. S., Wozniak, S., Zhang, R., Zhang, Z., Zhao, Q., Zhou, P., Zhu, J., and Zhu, R.-J. Rwkv: Reinventing rnns for the transformer era, 2023.

Pennington, J., Socher, R., and Manning, C. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

Pham, N. T., Kruszewski, G., Lazaridou, A., and Baroni, M. Jointly optimizing word representations for lexical and sentential tasks with the C-PHRASE model. In Zong, C. and Strube, M. (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 971–981, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1094. URL https://aclanthology.org/P15-1094/.

Ray Chowdhury, J. and Caragea, C. Beam tree recursive cells. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28768–28791. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ray-chowdhury23a.html.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Robinson, J., Sun, L., Yu, K., Batmanghelich, K., Jegelka, S., and Sra, S. Can contrastive learning avoid shortcut solutions? *CoRR*, abs/2106.11230, 2021. URL https://arxiv.org/abs/2106.11230.

Rücklé, A., Eger, S., Peyrard, M., and Gurevych, I. Concatenated p-mean word embeddings as universal cross-lingual sentence representations. *CoRR*, abs/1803.01400, 2018. URL http://arxiv.org/abs/1803.01400.

Sartran, L., Barrett, S., Kuncoro, A., Stanojević, M., Blunsom, P., and Dyer, C. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439, 2022. doi: 10.1162/tacl_a_00526. URL https://aclanthology.org/2022.tacl-1.81.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 151–161, 2011.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1631–1642, 2013.

Soulos, P., Conklin, H., Opper, M., Smolensky, P., Gao, J., and Fernandez, R. Compositional generalization across distributional shifts with sparse tree operations. *arXiv preprint arXiv:2412.14076*, 2024.

Tai, K. S., Socher, R., and Manning, C. D. Improved semantic representations from tree-structured long short-term memory networks. In *Association for Computational Linguistics (ACL)*, pp. 1556–1566, 2015.

Thakur, N., Reimers, N., R"uckl'e, A., Srivastava, A., and Gurevych, I. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

Turc, I., Chang, M., Lee, K., and Toutanova, K. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962, 2019. URL http://arxiv.org/abs/1908.08962.

Vashishth, S., Bhandari, M., Yadav, P., Rai, P., Bhattacharyya, C., and Talukdar, P. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Association for Computational Linguistics (ACL)*, pp. 3308–3318, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019.

Wang, B., Kuo, C.-C. J., and Li, H. Just rank: Rethinking evaluation with word and sentence similarities. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6060–6077, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.419. URL https://aclanthology.org/2022.acl-long.419.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., and Zhang, Z. Deep graph library: A graph-centric, highly-performant package for graph neural networks, 2020.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *CoRR*, abs/2005.10242, 2020. URL https://arxiv.org/abs/2005.10242.

Wiedemer, T., Mayilvahanan, P., Bethge, M., and Brendel, W. Compositional generalization from first principles, 2023. URL https://arxiv.org/abs/2307.05596.

Wieting, J., Gimpel, K., Neubig, G., and Berg-Kirkpatrick, T. Paraphrastic representations at scale. *CoRR*, abs/2104.15114, 2021. URL https://arxiv.org/abs/2104.15114.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

A. The *k* and *u* balance

The change to diagonal composition functions allows us to reduce the number of total parameters while maintaining performance. This is because the number of parameters is directly proportional to channel size *u*. We show ablations for this finding in Table 7. Our findings are similar to those of (Opper & Siddharth, 2024) the smaller the channel size the better the model performs, although in our case we keep things stable between seeds whereas before simplification caused issues with extreme instability during training. This is thanks to the new message passing functions.

Table 7: Performance Depending on *k* and *u* values using new functions. Scores are the average of four random seeds.

k	*u*	Lex Score	STS Score
4	64	47.83 ± 0.2	55.50 ± 0.22
8	32	47.41 ± 1.0	62.58 ± 0.11
16	16	48.01 ± 1.1	62.73 ± 0.1
32	8	47.65 ± 1.1	62.79 ± 0.07
64	4	48.48 ± 0.7	62.63 ± 0.16
128	2	48.53 ± 1.33	62.97 ± 0.23
256	1	49.15 ± 0.6	62.61 ± 0.23

B. Hyperparameters:

We trained SELF-STRAE and BANYAN for 15 epochs (circa 15k steps and sufficient for convergence) using the Adam optimiser (Kingma & Ba, 2015), with a learning rate of 1e-3 for BANYAN and 1e-4 for SELF-STRAE using a batch size of 512. To process the graphs we used DGL (Wang et al., 2020). The GLOVE baseline was trained for 15 epochs with a learning rate of 1e-3, and a window size of 10. We used the official C++ implementation. ROBERTA medium was trained for 200,000 steps, (10% of which were used for warmup). We used a learning rate of 5e-5, and a linear schedule. Positional embeddings are relative key-query. We used the Transformers library to implement and train the model (Wolf et al., 2020). For SimCSE training, we used the default parameters and the official implementation for unsupervised ROBERTA training from Gao et al. (2021). For Sent2Vec we used their official implementation and recommend hyperparameters.

C. UMAP Parameters:

For the UMAP visualisations, we set number of neighbours to 100, minimum distance to 0.3, the metric to cosine and local connectivity to 3. However, the same patterns can we observed through a wide array of hyperparameters, and when changing the metric to euclidean distance. The behavioural changes induced by the diagonal functions remain clear. We selected the above purely based on aesthetic preference for the resulting plots.