

How do Transformer Embeddings Represent Compositions? A Functional Analysis

Aishik Nagar^{1*} Ishaan Rawal^{2,3,4} Mansi Dhanania^{2,3,5} Cheston Tan^{2,3}

¹ASUS Intelligent Cloud Services

²Institute of High Performance Computing (IHPC),
Agency for Science, Technology and Research (A*STAR)

³Center for Frontier AI Research (CFAR),
Agency for Science, Technology and Research (A*STAR)

⁴Texas A&M University ⁵McGill University

Abstract

Compositionality is a key aspect of human intelligence, essential for reasoning and generalization. While transformer-based models have become the de facto standard for many language modeling tasks, little is known about how they represent compound words, and whether these representations are compositional. In this study, we test compositionality in Mistral, OpenAI Large, and Google embedding models, and compare them with BERT. First, we evaluate compositionality in the representations by examining six diverse models of compositionality (addition, multiplication, dilation, regression, etc.). We find that ridge regression, albeit linear, best accounts for compositionality. Surprisingly, we find that the classic vector addition model performs almost as well as any other model. Next, we verify that most embedding models are highly compositional, while BERT shows much poorer compositionality. We verify and visualize our findings with a synthetic dataset consisting of fully transparent adjective-noun compositions. Overall, we present a thorough investigation of compositionality.

1 Introduction

Large Language Models (LLMs) have emerged as the de facto standard for language generation and modeling tasks (Zhao et al., 2023), largely due to their superior generalization abilities compared to earlier models like BERT (Devlin et al., 2019a). One key factor contributing to this generalizability is **compositionality** – the ability to form new concepts by combining known ones (Fodor and Lepore, 2002). However, testing the compositionality of these models is challenging. Previous works on transformer models have mostly relied on performance on benchmarks as proxies for compositional generalization (Johnson et al., 2017; Yu et al., 2022), but such numbers can be often misleading (Rawal et al., 2024). Thus, representation

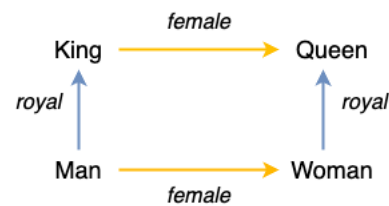


Figure 1: Linear concept transformation functions (*female* and *royal*) used to transform the representations of nouns (Ethayarajh et al., 2019).

analysis, like that in Fig. 1, can provide insights about the inner workings of a model. Thus, we conduct representation-centric analyses to verify the compositionality of transformer models.

While a substantial body of literature exists on the functional compositionality of Vector Space Models (VSMs) (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010), such understanding of transformer models, which also rely on the distributional hypothesis (Firth, 1957), remains limited. Additionally, exploring differences in how current models comprehend compositionality is crucial. To address these, we adopt a functional approach to investigate the compositionality of transformer-based embedding models. We focus on embedding models, due to their widespread use in applications like Retrieval-Augmented Generation (Lewis et al., 2020) and vector databases (Han et al., 2023). The primary goal of this work is specifically to analyze how transformer models represent constituents of compound words, and how these token-level representations relate to the embedding generated for the entire compound word, taken as context. Transformer models excel at generating contextual embeddings, which differ fundamentally from static embeddings traditionally used in vector space models (VSMs). This context-dependent embedding generation allows transformer models to represent higher-order se-

*Correspondence to: aishiknagar@gmail.com

semantic features, such as semantic dominance and transparency, which emerge naturally in language usage. For example, the model might have learned that the term "Red Cross," although literally composed of "red" and "cross," frequently carries an organization-like meaning distinct from a purely additive combination of its constituents, resulting in a lower similarity between their constituent representations and the compound, regardless of the method of tokenization. While it may seem intuitive that transformer models represent compound words as weighted averages of their constituent parts, the non-linear and multi-layered nature of these models makes composition far less predictable (Templeton et al., 2024). Tokenized compounds (e.g., "blue" + "tooth") are not always interpreted in a compositional way, and may instead be mapped to distinct, non-compositional meanings based on learned context (Haber and Poesio, 2024; Grindrod, 2024). This ambiguity—especially for polysemous or rare compounds—motivates a closer examination of whether transformer embeddings respect compositionality in practice.

Compound words are a significant aspect of natural language. In English, compound words account for approximately 4% of the total vocabulary in large corpora like the British National Corpus (Biber et al., 1999). Language models often fail to generalize to novel compounds or properly encode existing ones (Cordeiro et al., 2016) and if a model struggles with word-level composition, it may face challenges with more complex structures (Lake et al., 2017). Understanding how models handle lexical composition is therefore foundational for scaling up to phrases and sentences. Adjective-noun compositions and compound words have been widely studied as tests of compositionality (Reddy et al., 2011; Baroni and Zamparelli, 2010). They allow studying semantic composition in a controlled setting, minimizing syntactic variability. Although seemingly simple, two-word compounds and compositions can present significant challenges. For example, *ladybird* is a beetle unrelated to female birds. *Red ball* refers to a red-colored ball, whereas *chocolate ball* refers to a ball made of chocolate. Moreover, humans often create novel compositions, such as *deepfake*, which first appeared in 2018. As such, we focus on two-word compounds and adjective-noun compositions.

We extend the study of functional compositionality in VSMs to embedding models, by testing six compositionality models for combining constituent

word embeddings to generate the embedding of their composition. We find that (i) ridge regression can effectively explain compositionality in these models, and (ii) current models are significantly more compositional than BERT. We also test the generalizability of our method on novel compound words created through data augmentation. Finally, we test the properties of the models on the LADEC dataset and on our Simple Adjective-Noun Compositions (SANC) dataset to (iii) investigate and visualize the underlying mechanics.

2 Related Works

Coil and Schwartz (2023) questioned whether language models possess the ability to understand noun compounds. They inferred that GPT-3 was able to correctly understand the implicit relation between two nouns in compound words such as *chocolate crocodile*, and conclude that it denotes a *crocodile-shaped chocolate* due to the large amount of data it has access to through the web corpus.

Another study, by Buijtelar and Pezzelle (2023), utilized human judgement to evaluate two psycholinguistic measures of compound semantic analysis, namely, lexeme meaning dominance and semantic transparency. This work demonstrated that BERT resembled human intuition in terms of semantic measures, especially when provided sufficient context to the data inputted. Turney and Pantel (2010) and Bullinaria and Levy (2012)'s work on Distributional Semantic Models (DSMs) is discussed in appendix A.1.

Compositional Distributional Models. Mitchell and Lapata (2008) explored various compositional distributional models that derive vectors for phrases by composing the vectors of their constituent words. Their work on additive and multiplicative models provided a foundational understanding of how simple algebraic operations could approximate the meaning of phrases. Baroni and Zamparelli (2010) extended this by introducing the lexical function model, treating adjectives as functions mapping noun vectors to phrase vectors.

Word Embeddings and Neural Networks. The advent of word embeddings, such as Word2Vec (Mikolov et al., 2013) and BERT (Devlin et al., 2019b), provided dense, context-sensitive representations of words. These models have been instrumental in capturing subtle semantic nuances and have been widely used for various NLP tasks. Recent works on variational autoencoders for seman-

tic analysis, such as Kingma and Welling (2013), have expanded understanding of embedding spaces by identifying latent dimensions representing semantic attributes like “smiling” and “frowning”.

Compositionality in Neural Language Models. Carvalho et al. (2022) proposed a methodology for measuring compositional behavior in contemporary neural language models, focusing on adjectival modifier phenomena in adjective-noun compositions. They introduced novel tests inspired by Montague semantics to evaluate whether neural embeddings can reflect theoretical compositional properties, such as intersectivity and non-subsectivity. Their findings indicated that while neural language models capture some aspects of these properties, they often fail to do so consistently, particularly in handling non-intersective adjectives.

While existing studies have shed light on the compositional abilities of both transformer and non-transformer models, they overlook the detailed exploration of latent embeddings and specific embedding models. By examining specific embedding models provided by Google, OpenAI, and Mistral alongside traditional embeddings like BERT, we provide a comprehensive analysis that uncovers nuanced semantic behaviors and enhances our understanding of embedding-driven compositionality.

3 Testing Compositionality Models

We test 6 compositionality models (both simple mathematical models and ones with learned parameters). These are tested on 4 common embeddings.

3.1 Methods

3.1.1 Definitions

For a given 2-word compound c , the first constituent word is c_1 , while c_2 is the second constituent word. We define the embeddings of c_1 , c_2 and c to be vectors u , v and w respectively.

3.1.2 Compositionality Models

A model of compositionality is a function $f(u, v) = \hat{w}$, which produces a predicted embedding \hat{w} by combining u and v .

- **Simple Addition:** $f(u, v) = u + v$
Straightforward sum of the embeddings of the constituent words, as the simplest mathematical relation possible between two vectors (Park et al., 2023).
- **Weighted Addition:** $f(u, v) = \alpha u + \beta v$
Following Mitchell and Lapata (2008, 2009,

2010), embeddings are combined using a weighted sum.

- **Multiplication:** $f(u, v) = u \circ v$
Element-wise multiplication, capturing feature intersections where both constituent embeddings have high values (Baroni and Zamparelli, 2010).
- **Dilation:** $f(u, v) = (u \cdot u)v + (\lambda - 1)(u \cdot v)u$

Used by Mitchell and Lapata (2008, 2009, 2010) and Baroni and Zamparelli (2010), dilation involves stretching one vector towards another by dilation parameter λ .

- **Ridge Regression:**
The target values w are regressed against the concatenation of u and v (Baroni and Zamparelli, 2010; Vecchi et al., 2017), with parameter $\alpha = 1.0$. Linear regression gave very similar results.
- **Multi-Layer Perceptron (MLP):**
A simple MLP with two hidden layers (256 neurons and 128 neurons respectively), ReLU activation, and cosine embedding loss function. The train-test split was 80-20.

3.1.3 Optimization

For the Weighted Addition and Dilation models, parameters were tuned using grid search and gradient descent. Since results using grid search were consistently worse than gradient descent, we omit them. More details are in Supplementary Materials.

3.2 Experimental Setup

3.2.1 LADEC Dataset

The LADEC dataset (Gagné et al., 2019a) is a set of 8,956 compound words and their constituent words. For each compound word, semantic ratings and other psycholinguistic markers are provided. These markers include measures of semantic transparency, which reflect how much of the meaning of the compound can be predicted from its parts, and meaning retention, which indicates how much each constituent retains its meaning within the compound. These ratings were obtained from human participants, who were asked to evaluate compounds based on specific criteria in controlled laboratory settings. Examples of compounds in LADEC are: bookstore, brainstorm, copycat, dumbbell, hamstring, hotdog, toothache, turtleneck and waterfall.

3.2.2 LADEC–NC Dataset

We created an extension of LADEC to assess how embedding models represent novel out-of-dictionary compounds words, providing insights into robustness and generalization. For example, a compound word like ‘deepfake’ only came into existence in the era of generative vision models, but people easily grasped the gist of this novel (back then) compound from its constituent words ‘deep’ (from *deep learning*) and ‘fake’. **To what degree do the compositional capabilities of embedding models extend to novel compounds?** In the absence of human ratings, it can be postulated that these novel compounds have moderate to high semantic transparency – human interpretations of terms like “zoodough” would naturally combine meanings of their constituents (e.g. "dough used in a zoo") in the absence of contextual cues or prior lexical knowledge indicating that the compound carries an idiomatic or non-compositional meaning.

Starting with the compounds in LADEC, we generated lists of unique c_1 (first constituent word) and c_2 (second constituent word), e.g. ‘deep’ and ‘fake’ respectively. We then generated all pairwise combinations (i.e. compounds) from c_1 and c_2 , filtering out any existing words to ensure novelty. Finally, 10,000 of these were randomly selected to form the **LADEC – Novel Compounds (LADEC–NC)** dataset. Example novel compounds include: widowrise, throwlooms, hemnut and zoodough.

3.2.3 Embedding Models

We examine four different embedding models [dimensionality in brackets]: Google¹ [768], Mistral² [1024], OpenAI Large³ [3072], and BERT [768] (Devlin et al., 2019b). For BERT, we consider output embeddings corresponding to the CLS token 12th layer, because it holds maximum semantic information (Mohebbi et al., 2021) while there are proprietary APIs to obtain the embeddings for the other models. Our selection of the primary models was motivated by their widespread adoption in industry and academic research (for example, the usage of BERT in metrics like BERTScore (Zhang et al., 2019) and the usage of the proprietary models

¹<https://cloud.google.com/vertex-ai/generative-ai/docs/embeddings/get-text-embeddings>

²<https://docs.mistral.ai/capabilities/embeddings/>

³<https://platform.openai.com/docs/guides/embeddings/frequently-asked-questions>

in applications such as generating document representations and tasks such as retrieval-augmented-generation in LLMs). Resource and space constraints necessitated selecting a representative subset of models that clearly illustrate architectural differences (bidirectional vs. encoder-decoder embeddings) while being widely used for a diverse range of tasks which rely on the models to generate compositional and meaningful representations from text.

3.2.4 Metrics

To evaluate the different compositionality methods, we compute the cosine similarity between the predicted (\hat{w}) and actual (w) embeddings for each compound. These cosine similarities are then compared against a baseline distribution of cosine similarities between 1,000 random pairs of LADEC compounds (e.g. between ‘bookstore’ and ‘seashell’; between ‘hamstring’ and ‘turtleneck’). If random compound embeddings showed similarities comparable to actual compounds and their constituents, tasks relying on embedding uniqueness – such as document retrieval or text similarity (e.g. BERTScore (Zhang et al., 2019)) – would be infeasible. Thus, comparing actual compound similarities against a random baseline confirms whether embeddings encode meaningful semantic distinctions or are merely noise. The purpose of the random baseline is not to capture variance across transparency levels, but rather to assess whether compound–constituent pairs lie meaningfully apart from random pairs at the distribution level as a sanity check for our approach.

To quantify the difference between a method’s distribution and the baseline distribution, we calculate the Jensen-Shannon (JS) Divergence (Lin, 1991). This measure of dissimilarity between two distributions is symmetric, and always gives finite values between 0 and 1. Two distributions are similar when their JS Divergence is close to 0. In this work, a larger divergence indicates that a method better captures semantic compositionality, i.e. larger difference from the baseline distribution.

3.2.5 Visualization

We use Kernel Density Estimation (KDE) plots to visualize the distribution of cosine similarities. Peaks in the plots indicate typical similarity levels, while the curve spread shows variability.

3.3 Results

From Figure 2 and Table 1, the most striking results at first glance are that: (i) the Multiplication model does extremely poorly across all embeddings, with the blue distributions lower than the baselines in red; (ii) BERT does not performed well, with large overlaps between the blue and red distributions. Detailed analyses is below and Appendix B.

3.3.1 How do modern transformer-based embedding models compare to BERT?

From Figure 2, BERT is much less compositional than Google, Mistral and OpenAI Large. Apart from the Multiplication model (which performs poorly across all embeddings), all models of compositionality show large separations between the blue and red plots for all embedding models other than BERT. This is also reflected quantitatively in Table 1; for all compositionality models, BERT has much lower JS Divergence than all others.

3.3.2 Which compositionality model best predicts compound embeddings?

From Table 1, for all embedding models, Ridge Regression results in the highest JS Divergence values (by a small margin) and highest cosine similarities. MLP produces JS Divergence values close to those of Ridge Regression, but produces generally lower cosine similarities. Given that Ridge Regression is a linear model, it is somewhat surprising that it outperforms the non-linear MLP.

3.3.3 How does the classic simple vector addition model perform?

For the modern transformer-based embeddings (Google, Mistral and OpenAI Large), this simplest model of compositionality performs surprisingly well. The JS Divergence values are only marginally lower than those for Ridge Regression and NLP, while the mean cosine similarities are near (89%) to those for Ridge Regression: 0.772 vs. 0.868 (average over 3 embedding models). Interestingly, the performance of Simple Addition is virtually the same as Weighted Addition and Dilation. Overall, this classic simple model holds up surprisingly well for predicting modern compound embeddings.

3.3.4 Does compositionality arise from architecture or training data?

From Figure 2 and Table 1, a number of embeddings and compositionality models show good re-

sults, with both high JS Divergence and cosine similarities. Could the existence of compounds words and their constituents (e.g. ‘deepfake’, ‘deep’ and ‘fake’) in training corpora give rise to compositionality? **In other words, is compositionality innate or learnt?** From Table 2, the metrics for LADEC versus LADEC-NC are very similar. This means that embeddings of never-seen-before compounds (e.g. ‘zoodough’) can be predicted from the embeddings of their constituents (‘zoo’ and ‘dough’) to the same extent as existing compounds. The results suggest that compositionality arises from the architecture of embedding models, rather than their training data or recipe.

Embed. Model	Composition Model	Cosine Sim.	JS Div.
Google	Simple Addition	0.852	0.824
Google	Weighted Addition	0.854	0.824
Google	Multiplication	-0.038	n/a
Google	Dilation	0.854	0.824
Google	Ridge Regression	0.922	0.829
Google	MLP	0.865	0.827
Mistral	Simple Addition	0.854	0.829
Mistral	Weighted Addition	0.856	0.829
Mistral	Multiplication	-0.042	n/a
Mistral	Dilation	0.856	0.829
Mistral	Ridge Regression	0.914	0.831
Mistral	MLP	0.854	0.829
OpenAI	Simple Addition	0.609	0.819
OpenAI	Weighted Addition	0.609	0.819
OpenAI	Multiplication	0.001	n/a
OpenAI	Dilation	0.607	0.818
OpenAI	Ridge Regression	0.767	0.830
OpenAI	MLP	0.673	0.825
BERT	Simple Addition	0.891	0.289
BERT	Weighted Addition	0.891	0.294
BERT	Multiplication	-0.610	n/a
BERT	Dilation	0.891	0.293
BERT	Ridge Regression	0.932	0.501
BERT	MLP	0.932	0.496

Table 1: Comparison of compositionality models for each type of embedding model, for the LADEC dataset. Numbers in blue and magenta are the highest and second-highest for each embedding model.

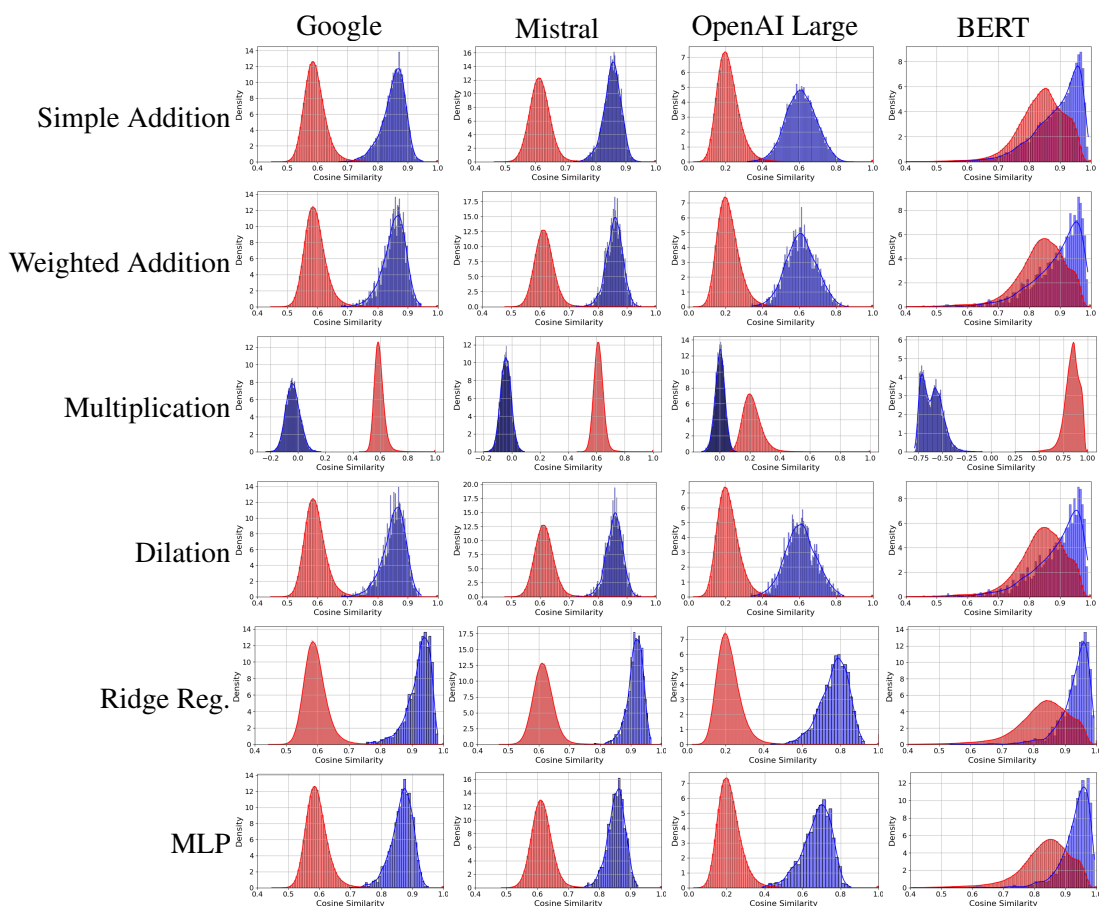


Figure 2: Comparison across compositionality models (rows) and embedding models (columns). **Blue plots** are the KDEs of cosine similarity values between the predicted (\hat{w}) and ground truth (w) embeddings, while the **red plots** are the baseline distributions (KDEs of cosine similarities between embeddings of random pairs of compounds).

4 Generalization to Systematic Compositions

Comp. Model	Dataset	Cosine Sim.	JS Div.
Simple Add.	LADEC	0.854	0.829
Simple Add.	LADEC-NC	0.862	0.831
Weighted Add.	LADEC	0.856	0.829
Weighted Add.	LADEC-NC	0.863	0.832
Multiplication	LADEC	-0.042	n/a
Multiplication	LADEC-NC	-0.037	n/a
Dilation	LADEC	0.856	0.829
Dilation	LADEC-NC	0.863	0.832
Ridge Reg.	LADEC	0.914	0.831
Ridge Reg.	LADEC-NC	0.925	0.832
MLP	LADEC	0.854	0.828
MLP	LADEC-NC	0.827	0.831

Table 2: Comparing LADEC vs. LADEC-NC, existing and novel compounds show very similar compositionality metrics. Mistral embeddings are used here, but other embeddings give very similar results.

The LADEC dataset contains compound words like ‘bookstore’, ‘dumbbell’ and ‘waterfall’, which are all single-word concepts composed from two constituent words. These compounds have varying levels of semantic transparency, and have arbitrary semantic relationships between the constituents. For example, ‘bookstore’ is a “store for books”, but ‘dumbbell’ is not a “bell for dumbs”. ‘Dumbbell’ originates from it looking like a church bell, but is dumb or silent (i.e. the noun is the second constituent), whereas ‘waterfall’ relates to water that falls (i.e. noun is first constituent).

Do the results in the previous section **generalize to two-word compositions that are composed more systematically**? To investigate this, we turn to adjectives and nouns, because in adjective-noun compositions, there is a clear semantic relationship between the constituents.

4.1 Simple Adjective-Noun Compositions (SANC) Dataset

We design a dataset that comprises all pairwise combinations of 25 common adjectives with 25 common nouns. The 25 adjectives are from 3 common categories (12 colors, 7 materials and 6 size/shape attributes). The full set of adjectives and nouns can be seen in Figure 5.

Inspired by Baroni and Zamparelli (2010) and Gagné et al. (2019b), we constrain our dataset to simple adjective-noun compositions, because common nouns and adjectives are exhaustively composable, meaning that any adjective can be combined with any noun, regardless of commonness or plausibility (e.g. *metal papaya* or *magenta watermelon*). Furthermore, the adjective-noun compositions are semantically transparent, i.e. *red ball* is clearly both *red* and a *ball* (unlike *flea market*, for instance).

4.2 Methods and Experimental Setup

For this section on generalization, we use only the classic **Simple Addition** compositionality model, due to its simplicity (no learning or parameters), and the fact that its performance is close to the best model (Ridge Regression).

4.2.1 Semantically-Similar Phrases

We want to gauge how the Simple Addition model fares in relation to semantically-similar phrases. For example, where $S(x, y)$ is the cosine similarity between x and y , and $\langle p \rangle$ is the embedding vector of phrase p , how does Simple Addition compare to semantically similar variations?

The comparisons include several formulations of S : ($\langle \text{red ball} \rangle$, $\langle \text{red} \rangle + \langle \text{ball} \rangle$), ($\langle \text{red ball} \rangle$, $\langle \text{red-colored ball} \rangle$), ($\langle \text{red ball} \rangle$, $\langle \text{ball is red-colored} \rangle$), and ($\langle \text{red ball} \rangle$, $\langle \text{color of ball is red} \rangle$). Additionally, we evaluate against baselines such as ($\langle \text{red ball} \rangle$, $\langle \text{red} \rangle$), ($\langle \text{red ball} \rangle$, $\langle \text{ball} \rangle$), and ($\langle \text{red ball} \rangle$, $\langle \text{color of door is green} \rangle$). From the SANC dataset, we cross the 12 color adjectives with all 25 nouns, then plot the KDEs (using $12 * 25 = 300$ compositions) separately for each variation or baseline (see Figure 4). BERT was not used, due to poor compositionality. We use only colors for this experiment, in order to automatically generate semantically-similar phrases (e.g. ‘red-colored ball’).

4.3 Results

4.3.1 Generalization to Adjective-Noun Compositions

From Figure 3, we see that the Simple Addition model also works well on the SANC dataset for Google, Mistral and OpenAI Large, with fairly high JS Divergence values and cosine similarities (details in Supplementary Materials). The good performances on LADEC, LADEC-NC and SANC point to the generality of the Simple Addition model across different types of compositions.

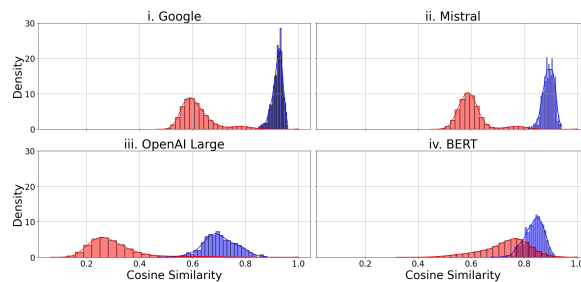


Figure 3: KDEs across models for the SANC dataset. Blue plots are the cosine similarity distributions using Simple Addition. Red plots are baseline distributions.

4.3.2 Comparison to Semantically-Similar Phrases

From Figure 4, for both Google and Mistral, the order of the distributions (from highest to lowest) is roughly: blue, red, orange/green, cyan/purple, black. Using ‘red ball’ to illustrate, this means that the similarity of the various embeddings to $\langle \text{red ball} \rangle$, from most to least similar, are:

- $\langle \text{red-colored ball} \rangle$
- $\langle \text{red} \rangle + \langle \text{ball} \rangle$
- $\langle \text{ball is red-colored} \rangle$ or $\langle \text{color of ball is red} \rangle$
- $\langle \text{red} \rangle$ or $\langle \text{ball} \rangle$
- $\langle \text{color of door is green} \rangle$

These results not only match the rough intuition of how semantically similar the phrases are to ‘red ball’, but show that the Simple Addition model $\langle \text{red} \rangle + \langle \text{ball} \rangle$ represents ‘red ball’ almost as well as ‘red-colored ball’ (which is semantically almost identical to ‘red ball’).

Results for OpenAI Large are qualitatively similar, except that the red distribution is no longer distinctly higher than the orange and green ones.

5 Deeper Performance Analysis

Our findings suggest some insights for new models:

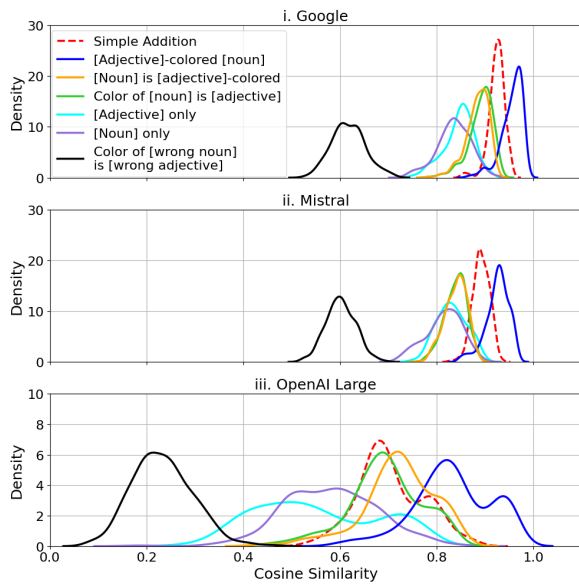


Figure 4: Compositionality of Simple Addition in relation to 3 semantically-similar phrases and 3 baselines.

- The strong performance of simple linear models suggests that transformer embeddings support linear additive compositionality. This aligns with previous findings and indicates that **embedding spaces could be optimized for linear operations**.
- BERT’s poorer performance compared to models like Mistral and OpenAI suggests that **architectural choices** (e.g., bidirectional encoder vs. autoregressive models) and training objectives impact compositionality.
- Models can be trained with **optimization objectives** that promote linear separability and **semantic transparency**, potentially improving performance on downstream tasks requiring compositional understanding.

A complete discussion on this can be found in Appendix B.

6 Qualitative Analyses

6.1 Methods

6.1.1 Constituent dominance

For each adjective-noun composition in SANC, we analyzed whether it is more dominated by (similar to) the adjective or the noun, i.e. we compare S (<red ball>, <red>) to S (<red ball>, <ball>).

6.1.2 Visualization using UMAP

We used UMAP – Uniform Manifold Approximation and Projection (McInnes et al., 2018) to visu-

alize the embedding space of adjective-noun compositions in SANC.

6.2 Results

6.2.1 Constituent dominance

From Figure 5, we see that there is a diversity among adjective-dominance and noun-dominance; it is not the case the adjectives always dominate, or nouns always dominate. More importantly, we see that the embeddings are quite different, with Google, OpenAI Large and BERT increasingly different from Mistral. Despite these different representations, it is interesting that Google, Mistral and OpenAI Large give fairly similar results in Figure 2 across compositionality models.

6.2.2 UMAP visualizations

Figure 6 (top and middle) shows the UMAP representations of Mistral embeddings predicted by the Ridge Regression model, labelled either adjective and noun constituent. We can see that when a constituent dominates the representation (e.g. ‘beige’, ‘watermelon’), its compounds are well-clustered near it. However, for constituents that are not dominant (e.g. ‘brown’, ‘tree’), its compounds are dispersed. It is important to note that that dominance is defined based on cosine similarity on the ground-truth embedding, while the visualization is based on UMAP (using euclidean distance) on predictions of Ridge Regression. These independent analyses suggest that Ridge Regression indeed performs semantic compositionality.

Figure 6 (bottom) depicts a few example compositions (circles) and their constituents (stars). Qualitatively, it is interesting to note that several – though not all – compositions (e.g. ‘maroon watermelon’, ‘metal cup’, etc) are located between their constituents, consistent with linear models (such as Ridge Regression and Weighted Addition) being able to account for compositionality well.

These visualizations demonstrate that compound words tend to cluster around one of their constituent parts, confirming that semantic contributions within compounds are influenced by factors like semantic richness and contextual salience, rather than fixed syntactic rules (Libben, 1998). This finding aligns with the *dual coding theory*, which highlights the deeper semantic processing of concrete nouns due to their imageability, leading to stronger clustering effects (Paivio, 1990). Additionally, dominant adjectives in UMAP visualizations support the *qualia structure* in generative lexicon

theory, where adjectives play a crucial role in shaping the semantic interpretation of nouns (Pustejovsky, 1998). The observed clustering behavior underscores the dynamic construction of compound meanings based on the semantic properties of their parts, resonating with *contextualism* in semantic theory and *semantic frame theory* (Recanati, 2004; Fillmore et al., 2006).

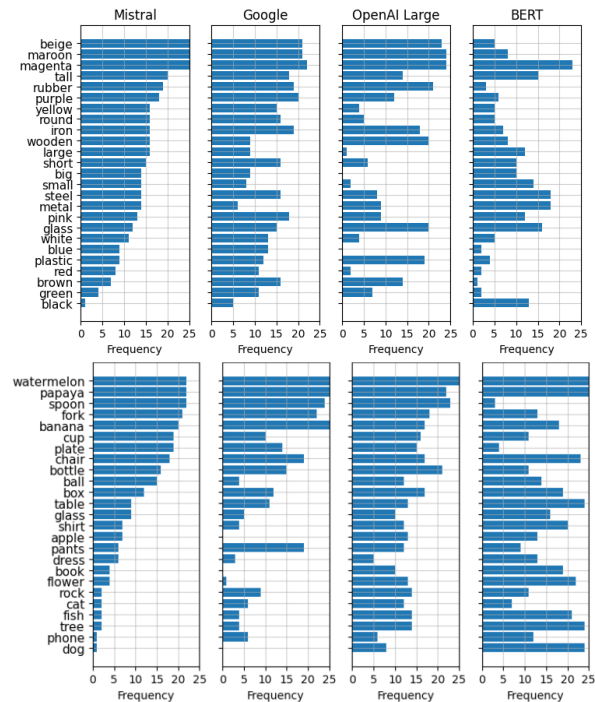


Figure 5: Constituent dominance of **adjectives** and **nouns**, ordered by Mistral dominance rank.

7 Conclusion

In this study, we explored the compositionality of various transformer-based embedding models. For the first time, we comprehensively analyse modern embedding models and evaluate their compositionality over several metrics and factors. Our results showed that modern transformer models exhibit strong compositionality compared to BERT. Functional analysis revealed that ridge regression, a linear model, best explained compositionality in these embeddings. We also found that these embeddings are highly compositional regardless of whether the compound words were seen during training, suggesting that compositionality arises from the embedding models' architectures, rather than training process or data. Our work significantly advances our understanding of compositionality in modern language models.

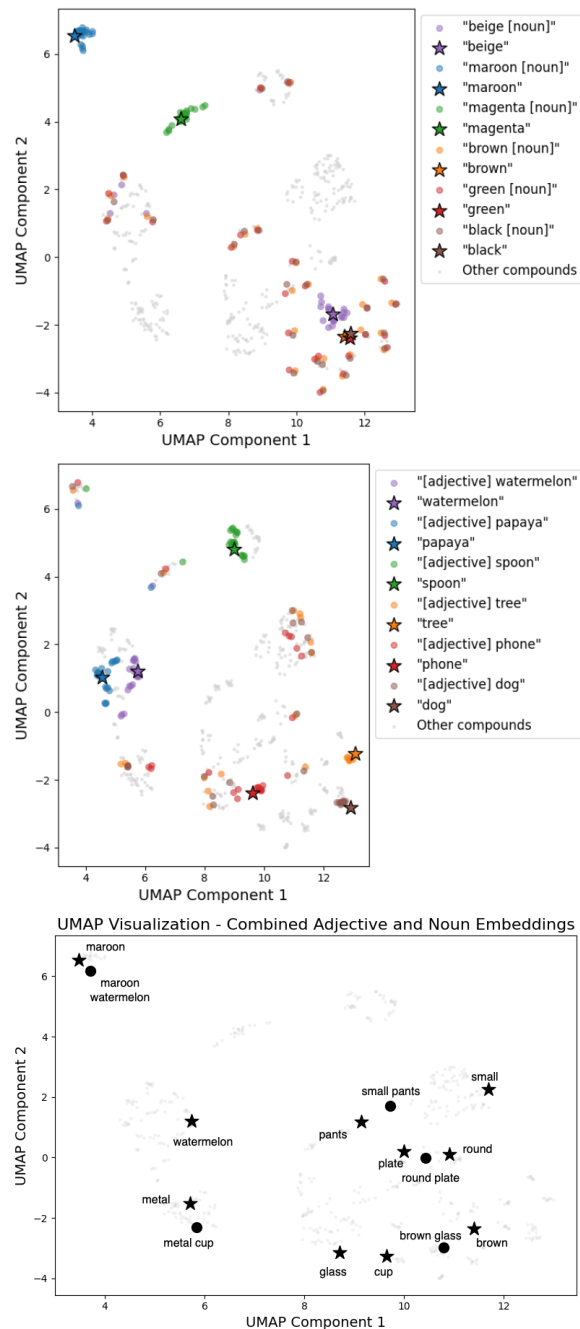


Figure 6: UMAP plot of Mistral embeddings using Ridge Regression. The top figure shows the UMAP color-coded by **adjective**. The most dominant adjectives are 'beige', 'maroon' and 'magenta'; least are 'brown', 'green' and 'black'. The middle figures shows the UMAP colored by **noun**. The most dominant nouns are 'watermelon', 'papaya' and 'spoon'; the least are 'tree', 'phone' and 'dog'. The bottom-most UMAP plots the nouns, adjectives and their corresponding compositions on the same plot.

8 Limitations

While our study provides valuable insights into the compositionality of transformer-based embeddings, several limitations must be acknowledged:

- **Scope of Models:** We focused on a limited set of embedding models, specifically Google, Mistral, OpenAI Large, and BERT. Other prominent embeddings (e.g., GPT-4 embeddings or smaller, domain-specific models) were not included and may yield different results.
- **Simple Adjective-Noun Compositions:** Our experiments are constrained to two-word compositions, with an additional focus on adjective-noun pairs. This was done to control for syntactic complexity and precisely measure compositional properties. While this setup provides a controlled evaluation, it does not account for more complex compositional structures involving multiple modifiers or sentence-level contexts.
- **Dataset Bias:** The LADEC and SANC datasets, though effective for testing compositionality, may not fully capture real-world variability in compound words and phrases. Additionally, LADEC-NC consists of synthetically generated compounds, which may lack linguistic or cultural plausibility.
- **BERT-Specific Challenges:** BERT’s poor performance, attributed to subword tokenization and bidirectionality, highlights specific architectural limitations. However, our analysis does not explore potential improvements for BERT (e.g., fine-tuning on compositional tasks) or the effects of alternative training strategies.
- **Linear Assumptions:** Our reliance on linear models (e.g., Ridge Regression and Simple Addition) assumes that compositionality can be approximated linearly. While this works well empirically, it may oversimplify the nuanced, non-linear interactions present in transformer embeddings.
- **Evaluation Metrics:** Cosine similarity and JS divergence were used as primary metrics to assess compositionality. While these metrics are widely used, they may not fully capture

the semantic nuances or compositional relationships in all contexts.

- **Model Training Data:** We do not analyze the role of specific training data in enabling compositionality. The presence or absence of certain compounds during pretraining could influence the models’ ability to generalize to novel compositions.
- **More models and architectures:** While we provide additional models in section A.10, including more models and newer architectures can further improve the comprehensiveness of the findings.
- **Evaluation metrics:** additional semantic evaluation metrics beyond cosine similarity and JS divergence could provide richer insights into the compositionality nuances. We chose these metrics due to their interpretability and widespread acceptance in previous literature but exploring more precise and specific evaluation metrics could lead to richer understanding of the embedding representations.

Future work can address these limitations by including a broader range of embedding models, exploring complex multi-word compositions, and investigating the impact of alternative training objectives and evaluation frameworks.

References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org.
- Marco Baroni and Roberto Zamparelli. 2010. *Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space*. In *Empirical Methods in Natural Language Processing*, pages 1183–1193.
- Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, and Edward Finegan. 1999. Longman of

- spoken and written english. *Harlow: Pearson Education*.
- Marc Brysbaert and Boris New. 2009. [Subtlex-us: A new measure for american english word frequency based on film subtitles](#). *Behavior Research Methods*, 41(4):978–990.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. [Concreteness ratings for 40 thousand generally known english word lemmas](#). *Behavior Research Methods*, 46(3):904–911.
- Lars Buijelaar and Sandro Pezzelle. 2023. [A psycholinguistic analysis of BERT’s representations of compounds](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2230–2241.
- John A Bullinaria and Joseph P Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior Research Methods*, 44:890–907.
- Danilo S Carvalho, Edoardo Manino, Julia Rozanova, Lucas Cordeiro, and André Freitas. 2022. [Montague semantics and modifier consistency measurement in neural language models](#). *arXiv preprint arXiv:2212.04310*.
- Albert Coil and Vered Shwartz. 2023. [From chocolate bunny to chocolate crocodile: Do language models understand noun compounds?](#) In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2698–2710.
- Silvio Cordeiro, Carlos Ramisch, Marco Idiart, and Aline Villavicencio. 2016. Predicting the compositionality of nominal compounds: Giving word embeddings a hard time. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1986–1997. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.
- Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. [Towards understanding linear word analogies](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262.
- Charles J Fillmore et al. 2006. Frame semantics. *Cognitive Linguistics: Basic Readings*, 34:373–400.
- Jr Firth. 1957. A synopsis of linguistic theory, 1930–1955. *Studies in Linguistic Analysis*.
- Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.
- Christina L Gagné, Thomas L Spalding, and Daniel Schmidtke. 2019a. [Ladec: The large database of english compounds](#). *Behavior Research Methods*, 51:2152–2179.
- Christina L Gagné, Thomas L Spalding, and Daniel Schmidtke. 2019b. [LADEC: The large database of english compounds](#). *Behavior Research Methods*, 51(5):2152–2179.
- Jumbly Grindrod. 2024. [Transformers, contextualism, and polysemy](#). *arXiv preprint arXiv:2404.09577*.
- Janosch Haber and Massimo Poesio. 2024. [Polysemy—Evidence from linguistics, behavioral science, and contextualized language models](#). *Computational Linguistics*, 50(1):351–417.
- Yikun Han, Chunjiang Liu, and Pengfei Wang. 2023. [A comprehensive survey on vector database: Storage and retrieval technique, challenge](#). *arXiv preprint arXiv:2310.11703*.
- Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. [CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.
- Barbara J Juhasz, Yun-Hsuan Lai, and Michelle L Woodcock. 2015a. [A database of 629 english compound words: ratings of familiarity, lexeme meaning dominance, semantic transparency, age of acquisition, imageability, and sensory experience](#). *Behavior research methods*, 47:1004–1019.
- Barbara J. Juhasz, Yun-Hsuan Lai, and Michelle L. Woodcock. 2015b. [A database of 629 english compound words: Ratings of familiarity, lexeme meaning dominance, semantic transparency, age of acquisition, imageability, and sensory experience](#). *Behavior Research Methods*, 47(4):1004–1019.
- Say Young Kim, Melvin J. Yap, and Winston D. Goh. 2019. [The role of semantic transparency in visual word recognition of compound words: A megastudy approach](#). *Behavior Research Methods*, 51(6):2722–2732.
- Diederik P Kingma and Max Welling. 2013. [Auto-encoding variational bayes](#). *arXiv preprint arXiv:1312.6114*.
- Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. [Building machines that learn and think like people](#). *Behavioral and brain sciences*, 40:e253.

- Thomas K. Landauer and Susan T. Dumais. 1997. [A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge](#). *Psychological Review*, 104(2):211–240.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Gary Libben. 1998. Semantic transparency in the processing of compounds: Consequences for representation, processing, and impairment. *Brain and Language*, 61(1):30–44.
- Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.
- Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244.
- Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Empirical Methods in Natural Language Processing*, pages 430–439.
- Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.
- Hosein Mohebbi, Ali Modarressi, and Mohammad Taher Pilehvar. 2021. [Exploring the role of BERT token representations to explain sentence probing results](#). In *Empirical Methods in Natural Language Processing*, pages 792–806.
- Allan Paivio. 1990. *Mental Representations: A Dual Coding Approach*. Oxford University Press.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. [The linear representation hypothesis and the geometry of large language models](#). In *Causal Representation Learning Workshop at NeurIPS 2023*.
- James Pustejovsky. 1998. *The Generative Lexicon*. MIT Press.
- Ishaan Singh Rawal, Alexander Matyasko, Shantanu Jaiswal, Basura Fernando, and Cheston Tan. 2024. [Dissecting multimodality in VideoQA transformer models by impairing modality fusion](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42213–42244. PMLR.
- François Recanati. 2004. *Literal Meaning*. Cambridge University Press.
- Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218.
- Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. [Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet](#).
- Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.
- Walter J. B. van Heuven, Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. [Subtlex-uk: A new and improved word frequency database for british english](#). *Quarterly Journal of Experimental Psychology*, 67(6):1176–1190.
- Eva M Vecchi, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2017. Spicy adjectives and nominal donkeys: Capturing semantic deviance using compositionality in distributional spaces. *Cognitive Science*, 41(1):102–136.
- Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. [Norms of valence, arousal, and dominance for 13,915 english lemmas](#). *Behavior Research Methods*, 45(4):1191–1207.
- Xinyan Velocity Yu, Sewon Min, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. [Crepe: Open-domain question answering with false presuppositions](#). *arXiv preprint arXiv:2211.17257*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, et al. 2023. [A survey of large language models](#). *arXiv preprint arXiv:2303.18223*.

A Appendix

A.1 Further Related works

Distributional Semantic Models (DSMs). Turney and Pantel (2010) introduced DSMs, which derive vector-based word representations from patterns of word usage in large corpora. These models have proven effective for numerous applications, such as word similarity and analogy tasks. The seminal work by Bullinaria and Levy (2012) highlighted the importance of pointwise mutual information (PMI) scores in constructing effective DSMs, laying the groundwork for subsequent models.

A.2 Compute Details

1. The experiments were carried out on a compute server using an NVIDIA RTX 3060 with 12GB of VRAM, and an AMD Ryzen 5500 CPU with 16GB of RAM. For proprietary models, a standard API interface was used which has been provided in the code submission.
2. **Software and libraries used.** Tensorflow (Abadi et al., 2015) was used for model loading and management. The complete set of modules, packages and software used is described in the code submission.
3. A random seed of 42 was used for all sampling purposes.

A.3 Adjective-Noun Compounds Dataset

Table 3: List of 25 adjectives and 25 nouns used for curating adjective-noun compounds.

ADJECTIVES		NOUNS	
BEIGE	PLASTIC	APPLE	FORK
BIG	PURPLE	BALL	GLASS
BLACK	RED	BANANA	PANTS
BLUE	ROUND	BOOK	PAPAYA
BROWN	RUBBER	BOTTLE	PHONE
GLASS	SHORT	BOX	PLATE
GREEN	SMALL	CAT	ROCK
IRON	STEEL	CHAIR	SHIRT
LARGE	TALL	CUP	SPOON
MAGENTA	WHITE	DOG	TABLE
MAROON	WOODEN	DRESS	TREE
METAL	YELLOW	FISH	WATERMELON
PINK		FLOWER	

A.4 Ladec-NC Dataset

The Ladec-NC dataset as described in the main paper is included in the supplemental in the file `ladec_nc.jsonl`

A.5

Mathematical Formulations for Techniques

A.5.1 Simple Addition

Simple Addition is a simple un-weighted addition of the constituent vector embeddings.

$$p = \mathbf{u} + \mathbf{v}$$

A.5.2 Additive Model

The Additive Model sums the embeddings of the constituent words. It is expressed as:

$$p = \alpha \mathbf{u} + \beta \mathbf{v}$$

where α and β are scalar weights of the vectors \mathbf{u} and \mathbf{v} .

A.5.3 Multiplicative Model

This model involves an element-wise multiplication of the embeddings:

$$p_i = u_i v_i$$

for each component i of vectors \mathbf{u} and \mathbf{v} .

A.5.4 Dilation

Dilation modifies the interaction between two vectors by stretching one towards another.

$$p = (\mathbf{u} \cdot \mathbf{u})\mathbf{v} + (\lambda - 1)(\mathbf{u} \cdot \mathbf{v})\mathbf{u}$$

This model accounts for both direct and scaled vector interactions, where λ is a dilation parameter.

A.6 Final Learned Parameters for Gradient Descent Based Parameter Learning in Additive and Dilation Methods

This appendix summarizes the final learned parameters for each model after training via gradient descent over 100 epochs. Each model employed either the additive or dilation composition method, as indicated.

Additive Gradient Descent Method

- **Google:** Weights = [4.0680, 2.9924]
- **OpenAI Large:** Weights = [4.3841, 3.9983]
- **Mistral:** Weights = [5.3313, 3.9911]
- **BERT:** Weights = [5.0875, 5.5244]
- **Mistral Novel Compound Words:** Weights = [4.3603, 3.7515]
- **BERT Novel Compound Words:** Weights = [5.4968, 4.7989]

Dilation Gradient Descent Method

- **Google:** $\lambda = 3.1064$
- **OpenAI Large:** $\lambda = 4.5377$
- **Mistral:** $\lambda = 2.9842$
- **BERT:** $\lambda = 2.0609$
- **Mistral Novel Compound Words:** $\lambda = 2.8168$
- **BERT Novel Compound Words:** $\lambda = 2.3679$

Each parameter vector or scalar λ was optimized to minimize the loss function specific to each model, reflecting how well the gradient descent approach could approximate the theoretical composition method.

A.7 Grid Search Based Results for Additive and Dilation Models

Table 4 provides grid search based results for Additive and Dilation Methods, and Figure 7 provides the corresponding Density plots.

Table 4: Metrics for Additive and Dilation Methods Across Different Models

Model	Method	Cos. Dist.	JS Div.
Google	Additive	0.1460	0.8242
OpenAI Large	Additive	0.3910	0.8190
Mistral	Additive	0.1442	0.8289
BERT	Additive	0.1087	0.2893
Mistral Novel Compound Words	Additive	0.1381	0.8314
BERT Novel Compound Words	Additive	0.08055	0.2744
Google	Dilation	0.2493	0.7225
OpenAI Large	Dilation	0.5148	0.7287
Mistral	Dilation	0.2381	0.7524
BERT	Dilation	0.1252	0.2088
Mistral Novel Compound Words	Dilation	0.2309	0.8134
BERT Novel Compound Words	Dilation	0.1073	0.1036

A.7.1 Optimal Parameters from Grid Search

In our grid search for tuning the model parameters, we explored a range of weight combinations for the Additive method and a set of lambda values for the Dilation method. The weight combinations tested included values such as (0.5, 0.5), (0.6, 0.4), (0.2,0.8), (0.1,0.9) and others, while the lambda values ranged from -1 to 1.

The following table summarizes the final parameters selected for each model after the completion of the grid search process:

Table 5: Final Parameters Selected from Grid Search

Model	Additive Weights	Dilation Lambda
Google	(0.6, 0.4)	1
OpenAI Large	(0.5, 0.5)	1
Mistral	(0.6, 0.4)	1
BERT	(0.5, 0.5)	1
Mistral Wrong Phrases	(0.5, 0.5)	1
BERT Wrong Phrases	(0.5, 0.5)	1

These parameters were selected based on their performance in optimizing the respective loss functions used in each model’s evaluation, aiming to achieve the best results for the compositionality of embeddings in both the Additive and Dilation composition methods. An alternate Grid search was also performed for dilation where the lambda could take up to a max value of 40, at intervals of 1. The grid search suggested that lambda always optimized to the maximum possible value of available grid parameters (ie, 20 if the max value was 20 and 40 if the grid space ranged till 40 and so on). The density plots and metrics in all cases was similar to the provided results.

A.8 Correlation of compound embedding cosine similarities with Semantic Metrics

Tables 6 and 7 provide the complete correlation tables for the compound embeddings created by Google Embeddings and OpenAI Large Embeddings and their correlation with semantic metrics provided in the LADEC Dataset.

Compositionality is the dataset’s meaning-

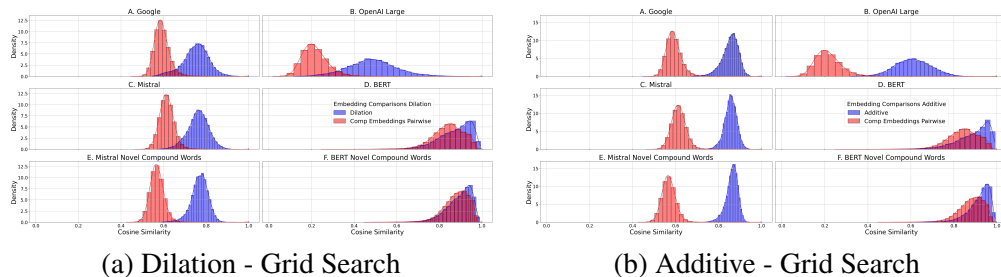


Figure 7: Cosine similarity distribution densities across different composition models. Each subfigure shows the result from a different method or optimization approach, providing a comprehensive overview of the embedding model performances.

predictability score (`ratingcmp`); it shows how easily a reader can guess a compound’s sense from its two parts. Juhasz_tran repeats semantic-transparency ratings collected by Juhasz, Lai and Woodcock for 629 English compounds (Juhasz et al., 2015a,b). `ratingC1` and `ratingC2` give meaning-retention judgments for the first and second constituents; `st_c1_mean` and `st_c2_mean` come from an update by Kim et al. (2019). The three LSA values—`LSAc1stim`, `LSAc2stim`, `LSAc1c2`—are cosine similarities from Latent Semantic Analysis: first part versus whole compound, second part versus whole compound, and the two parts together (Landauer and Dumais, 1997). Concreteness scores (`concreteness_stim`, `concreteness_c1`, `concreteness_c2`) originate in the Brysbaert et al. 40 k norms; higher numbers indicate richer perceptual content (Brysbaert and New, 2009; Brysbaert et al., 2014). Valence scores (`valence_stim`, `valence_c1`, `valence_c2`) come from Warriner et al. affective ratings; larger values point to a more pleasant tone (Warriner et al., 2013). Sentiment variables (`sentimentprobpos_`, `sentimentprobneg_`, `sentimentratioposneg_`) are outputs of a Mathematica sentiment classifier for the compound (`_stim`) and for each part (`_c1`, `_c2`). `Zipfvalue` presents frequency on the 1–7 Zipf scale (\log_{10} occurrences per billion words + 3) (van Heuven et al., 2014). `stim_SLLg10wf`, `c1_SLLg10wf` and `c2_SLLg10wf` store SUBTLEX-US \log_{10} word-frequency values for the compound and its two constituents (Brysbaert and New, 2009).

While the correlation values (Pearson) reported in Tables 6 and 7 are moderate, they are positive for well-performing models like Google and Mistral across key semantic factors—such as the predictability of the compound from its constituents (semantic transparency), meaning retentions of constituent 1 and constituent 2, and relatedness of the

compound to each constituent. In contrast, BERT shows correlations that are close to zero or even negative for some metrics.

Note that the correlation values are always higher for the head (constituent 2) than the modifier (constituent 1). This observation aligns with human ratings. In LADEC, the mean normalized meaning retention based on human scores for constituent 2 is 0.693, while for constituent 1 it is 0.644. This appears to reflect the natural behavior of English compounds, where the head typically carries more semantic weight, and the models appear to capture this tendency. Our deeper analysis through the UMAP projections in section 6.2 dives deeper into this phenomenon through examination of the existing linguistic theories around this subject.

Tables 6 and 7 show positive correlations between embedding-based compositionality scores and human-annotated semantic transparency ratings. These findings confirm our expectation: transparent compounds ("red ball") exhibit high similarity to the sum of their constituents’ embeddings ("red" + "ball"), whereas opaque compounds ("bluetooth") do not. Thus, our results demonstrate that the models behave consistently with human semantic judgments.

A.9 SANC Simple Addition Numerical Results

Embedding Model	Cos. Sim.	JS Div.
Google	0.921	0.824
Mistral	0.893	0.822
OpenAI Large	0.703	0.779
BERT	0.838	0.542

Table 10: Metrics for Simple Addition on the SANC dataset.

Metric	Correlation
compositionality	1.0000
ratingC2	0.4352
Juhasz_tran	0.3888
st_c2_mean	0.3218
LSAc2stim	0.2868
ratingcmp	0.2804
st_c1_mean	0.2398
LSAc1stim	0.2282
concreteness_c2	0.2194
LSAc1c2	0.1676
ratingC1	0.1616
concreteness_stim	0.1360
concreteness_c1	0.08303
c2_SLlg10wf	0.07585
valence_stim	0.06658
sentimentratioposneg_c2	0.05392
valence_c2	0.04755
sentimentprobneg_stim	0.04322
sentimentprobpos_c1	0.01612
sentimentprobpos_c2	0.01163
sentimentratioposneg_c1	0.007423
sentimentprobneg_c1	-0.01365
sentimentprobpos_stim	-0.01573
valence_c1	-0.03590
sentimentprobneg_c2	-0.03883
sentimentratioposneg_stim	-0.04266
c1_SLlg10wf	-0.06150
Zipfvalue	-0.07600
stim_SLlg10wf	-0.07600

Table 6: Correlation values for Google model.

Metric	Correlation
compositionality	1.0000
ratingC2	0.3627
LSAc2stim	0.3193
st_c2_mean	0.3095
Juhasz_tran	0.2553
ratingcmp	0.2238
concreteness_stim	0.2232
LSAc1stim	0.2120
concreteness_c2	0.1699
st_c1_mean	0.1599
LSAc1c2	0.1503
ratingC1	0.08918
valence_stim	0.08348
concreteness_c1	0.05163
stim_SLlg10wf	0.04702
Zipfvalue	0.04702
sentimentprobneg_c1	0.01716
sentimentprobneg_c2	0.01496
sentimentratioposneg_c2	0.01179
sentimentprobpos_c1	0.002194
sentimentprobneg_stim	-0.003203
sentimentratioposneg_stim	-0.003465
sentimentprobpos_stim	-0.009310
valence_c2	-0.01035
sentimentratioposneg_c1	-0.01967
sentimentprobpos_c2	-0.02335
c2_SLlg10wf	-0.03055
valence_c1	-0.07380
c1_SLlg10wf	-0.1286

Table 8: Correlation values for Mistral model.

Metric	Correlation
compositionality	1.0000
Juhasz_tran	0.3449
ratingC2	0.2851
st_c2_mean	0.2823
st_c1_mean	0.2713
LSAc1stim	0.2284
ratingcmp	0.2279
LSAc2stim	0.2104
LSAc1c2	0.1658
concreteness_stim	0.1639
ratingC1	0.1596
valence_stim	0.1375
concreteness_c2	0.1099
concreteness_c1	0.08663
sentimentratioposneg_c2	0.03800
sentimentratioposneg_c1	0.02216
sentimentprobpos_c1	0.02207
sentimentprobpos_c2	0.01885
sentimentratioposneg_stim	0.008305
sentimentprobpos_stim	0.008204
valence_c2	-0.002929
sentimentprobneg_stim	-0.005037
sentimentprobneg_c2	-0.01864
sentimentprobneg_c1	-0.02530
valence_c1	-0.04324
Zipfvalue	-0.05018
stim_SLlg10wf	-0.05018
c2_SLlg10wf	-0.1207
c1_SLlg10wf	-0.1856

Table 7: Correlation values for OpenAI Large model.

Metric	Correlation
compositionality	1.0000
c2_SLlg10wf	0.1528
Zipfvalue	0.1195
stim_SLlg10wf	0.1195
valence_stim	0.07592
valence_c2	0.06451
sentimentprobpos_c1	0.04225
sentimentratioposneg_stim	0.03315
sentimentratioposneg_c1	0.02905
sentimentprobneg_c2	0.01631
st_c1_mean	0.008733
valence_c1	0.007264
sentimentprobpos_stim	0.006798
sentimentratioposneg_c2	0.003231
sentimentprobpos_c2	0.002738
concreteness_c2	0.002198
c1_SLlg10wf	-0.003704
LSAc1c2	-0.02588
LSAc1stim	-0.02968
sentimentprobneg_c1	-0.03122
LSAc2stim	-0.03632
sentimentprobneg_stim	-0.04079
concreteness_c1	-0.04906
Juhasz_tran	-0.05242
ratingC2	-0.05843
st_c2_mean	-0.06931
ratingC1	-0.07454
ratingcmp	-0.1124
concreteness_stim	-0.1471

Table 9: Correlation values for BERT model.

A.10 Additional Models

While our main analysis included four embedding models (Google, OpenAI, Mistral, and BERT), we also conducted preliminary evaluations on several additional smaller-scale models and other architectures. We provide JS divergence results for these models and architectures on the SANC dataset using the additive method in table 11, which align with our previous findings.

B Result Analysis

Our findings suggest some insights for new models:

- The strong performance of simple linear models suggests that transformer embeddings support linear additive compositionality. This aligns with previous findings and indicates that **embedding spaces could be optimized for linear operations.**

Model	JS Divergence
OpenAI Small	0.770269
CLIP	0.809656
FastText	0.555278
Comp-CLIP	0.808932

Table 11: Performance of Additional models on SANC dataset and additive method

- BERT’s poorer performance compared to models like Mistral and OpenAI suggests that **architectural choices** (e.g., bidirectional encoder vs. autoregressive models) and training objectives impact compositionality.
- Models can be trained with **optimization objectives** that promote linear separability and **semantic transparency**, potentially improving performance on downstream tasks requiring compositional understanding.

The poor performance of BERT can be explained by the following factors:

Bidirectionality in architecture:

- BERT’s bidirectional nature means that embeddings are context-dependent and influenced by both left and right contexts.
- This can lead to embeddings that are less stable and less amenable to simple additive composition, as they may capture complex interactions that are not easily linearized.

Masked Language Modeling (MLM) as training objective:

- BERT is trained using MLM, where tokens are masked, and the model predicts them based on context.
- This training may not encourage the model to learn representations that facilitate linear compositionality, as the focus is on contextual prediction rather than encoding compositional semantics.

Subword Tokenization in BERT:

- BERT’s WordPiece tokenizer consistently splits words into subword units, which can fragment compound words.
- This fragmentation may result in embeddings that don’t capture the full semantic meaning of the compound word.

- For example, a novel compound like “deep-fake” might be split into “deep” and “##fake,” and the model might not have sufficient context to combine these subwords effectively.
- This can affect BERT’s ability to generate meaningful embeddings for compound words, hindering compositionality.
- The other embedding models are also trained on a larger number of tokens which means a lot more compound words and complex phrases are natively present in their embedding representations.

Difference in Scale:

- The difference in the model size of BERT and the other embedding models could also be a potential factor in the relatively poor embedding quality for BERT.

In contrast, models like Mistral and OpenAI embeddings may be trained with objectives or architectures that better preserve compositional properties, or they might utilize tokenization methods that maintain whole-word representations more effectively.

Presence of Target Compounds in Model Vocabularies and Tokenization Effects

An important point to note in this analysis is the impact of tokenization and presence or absence of target compounds from the model vocabulary for each of the models. While intuitively one might assume that compounds not in the model vocabulary might be split into subparts which could ultimately yield some sort of weighted average, in practice this is not necessarily true.

The highly non-linear and multi-layered nature of these models makes the representation and the impact of tokenization more ambiguous than trivial—A compound represented as a single token (e.g., “bluetooth”) may still be mapped to a compositional meaning internally, while two separate tokens (“blue” + “tooth”) may be jointly mapped to a distinct, possibly non-compositional, representation. The crucial point to emphasize is that here is that even when a compound like “bluetooth” is tokenized into “blue” and “tooth,” transformer models do not necessarily represent the meaning of the compound as a weighted sum of its token embeddings as the implicit composition function. These models may learn that specific token sequences

correspond to non-compositional meanings during their training phase and encode this context in their representation phase. For instance, in pretraining data, "blue" and "tooth" might co-occur frequently in technological contexts, and the model may associate the entire phrase with a concept of technology rather than just with either word alone.

This variability makes it difficult to predict how meaning is constructed, and highlights why analyzing whether the relevance of composition functions, linguistic theories, and priors from VSM studies for transformer-based embeddings becomes even more important. This sort of emergent behavior has been documented in recent interpretability studies (Templeton et al., 2024). Such tokenization makes the composition task less trivial for the model, warranting an investigation into composition representation of such transformer models. This is especially true when we account for the fact that compound words may appear in multiple forms (e.g., "bluetooth" vs. "blue tooth") in the training corpora, where their meanings shift across contexts and impact model representations (Haber and Poesio, 2024; Grindrod, 2024). The impact of tokenization on such polysemous constituents or compounds can introduce ambiguity in compositional interpretation, particularly when the model's pretraining fails to capture dominant or contextually appropriate senses—an effect compounded by recent findings on superposition (Templeton et al., 2024), where overlapping semantic features are encoded in shared subspaces, making disambiguation more difficult.