Quantifying Compositionality of Classic and State-of-the-Art Embeddings

Zhijin Guo[1,2*] **Chenhao Xue**[1*] **Zhaozhen Xu**[2] **Hongbo Bo**[2] **Yuxuan Ye**[2]
Janet B. Pierrehumbert[1] **Martha Lewis**[3]

[1]University of Oxford [2]University of Bristol [3]ILLC, University of Amsterdam

Abstract

For language models to generalize correctly to novel expressions, it is critical that they exploit compositional meanings when this is justified. Even if we don't know what a "pelp" is, we can use our knowledge of numbers to understand that "ten pelps" makes more pelps than "two pelps". Static word embeddings such as Word2vec made strong, indeed excessive, claims about compositionality. The SOTA generative, transformer models and graph models, however, go too far in the other direction by providing no real limits on shifts in meaning due to context. To quantify the additive compositionality, we formalize a two-step, generalized evaluation that (i) measures the linearity between known entity attributes and their embeddings via canonical correlation analysis, and (ii) evaluates additive generalization by reconstructing embeddings for unseen attribute combinations and checking reconstruction metrics such as L2 loss, cosine similarity, and retrieval accuracy. These metrics also capture failure cases where linear composition breaks down. Sentences, knowledge graphs, and word embeddings are evaluated and tracked for the compositionality across all layers and training stages. Stronger compositional signals are observed in later training stages across data modalities, and in intermediate layers of the transformer-based model before a decline at the top layer. Code is available at: https://github.com/Zhijin-Guo1/quantifying-compositionality.

1 Introduction

The representation of entities, concepts, and relations as vector embeddings is a foundational technique in machine learning (Mikolov et al., 2013a,b). Despite their broad success, embeddings often lack interpretability. One approach to improving interpretability involves examining the *compositionality* of embeddings: if the embeddings of complex things can be understood as a function of known, interpretable representations, the information they encode becomes more accessible.

Compositionality also enables models to understand and generate novel combinations of familiar components (Hofmann et al., 2025). For instance, in natural language, we can infer the meaning of a phrase like *eco-friendly transport* even if we have never seen it before, simply by combining the meanings of *eco-friendly* and *transport*. In the context of embeddings, if a model can construct representations for unseen combinations by composing the embeddings of known parts, it suggests that the learned representation space is structured and generalizable.

Recent work has explored compositionality across word embeddings (Mikolov et al., 2013a), sentence embeddings (Hewitt and Manning, 2019; Xu et al., 2023), and graph embeddings (Bose and Hamilton, 2019), often highlighting examples like *king* - *man* + *woman* $\approx$ *queen* as evidence of linear semantic structure. However, these claims, particularly for static embeddings such as Word2Vec, have faced strong criticism. Notably, Church (2017) and others have argued that such findings are often cherry-picked and fail to generalize, thereby overstating the role of compositionality. In contrast, Transformer-based models such as BERT (Devlin et al., 2018), GPT (Brown, 2020), and Llama (Touvron et al., 2023) offer more expressive and flexible architectures. These models typically rely on last-layer representations to encode sentence meaning, but often in ways that are less constrained by additive composition. In addition, many linguistic phenomena such as compound nouns (*hot dog*), phrasal verbs (*give up*), and sarcastic expressions (*Great job, you broke it again*) cannot be accurately captured through additive mechanisms alone, as their meanings often depend on context rather than the meanings of individual components.

[*] Joint first author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 22130–22146
November 4-9, 2025 ©2025 Association for Computational Linguistics

This contrast between early enthusiasm for additive structure and the complexity of meaning in modern models motivates a more rigorous investigation. It is essential to evaluate how much linear additive composition is actually retained across different architectures, modalities, and stages of training. Our work formalizes such an evaluation framework and applies it systematically to word, sentence, and graph embeddings. Specifically, we formalize a two-step, modality-agnostic evaluation: (i) measuring the linear alignment between known attributes and their embeddings using canonical correlation analysis (CCA), and (ii) assessing additive generalization by reconstructing embeddings for unseen attribute combinations and checking distance loss, cosine similarity, and retrieval accuracy.

Our study makes several novel contributions:

- Quantifying linearity and generalization: We formalize a unified evaluation pipeline by integrating canonical correlation analysis (CCA) with additive reconstruction to assess compositional generalization.

- Evaluation generalized across modalities: Our evaluation is applied to Transformer-based sentence embeddings (e.g., BERT, GPT, Llama), static embeddings (Word2Vec), and knowledge graph embeddings used in recommender systems. We track compositional signals across layers and training stages.

- Empirical insights: Embeddings show increasing additive compositionality during training, with improved generalization in Multi-BERT and a 1.5× rise in attribute–embedding correlation in graph models. Word embeddings align with semantic and syntactic structure and decompose into roots and suffixes.

- Showing compositional limits: We quantify failure cases where additive methods fall short, revealing residuals that signal non-linear structure and unexplained semantics, guiding future research.

2 Related Works

The classical principle that "the meaning of a whole is built from its parts" underlies linguistic theory and modern distributional semantics, motivating decades of work on whether and how vector embeddings encode such structure. Moving beyond intuition, Elmoznino et al. (2024) define *representational compositionality* information-theoretically: a representation is compositional if it can be recoded as discrete symbol strings whose semantics is delivered by a *low-complexity* mapping. In this paper we instantiate, and quantitatively test, the **linear additive** case of that criterion across words, sentences and knowledge-graph entities.

When compositionality fails. Not all language is built compositionally: idiomatic compounds (*hot dog*), phrasal verbs (*take off*), sentiment of expressions (*yeah right*), and sarcasm invert or ignore the literal sum of their parts (Jackendoff et al., 2011; **?**; Dankers and Lucas, 2023). Early computational work detects this breakdown by treating non-compositional expressions as a *statistical outlier*: unusual association or semantic-asymmetry scores flag verb–object or noun compounds that resisted substitution tests (Tapanainen et al., 1998; Lin, 1999). A parallel strand relied on syntactic and selectional-preference cues (Cook et al., 2007; McCarthy et al., 2007), while neural phrase-embedding models merge frequent idioms into single atomic tokens (Mikolov et al., 2013b). Non-compositionality also manifests in vector spaces: analogy tests misrank many compounds that contradict the conceptual priors of compositionality (Church, 2017), and work detects systematic non-compositionality in multiword expressions (Yazdani et al., 2015). Even if compositionality exists, the mapping of conceptual components could be nonlinear, as quadratic or polynomial projections, complexity-theoretic definitions, or dynamic, role-dependent nonlinear combinations (Yazdani et al., 2015; Elmoznino et al., 2024; Wang and Xu, 2024). The set of conceptual components could also be incomplete or defined inaccurately (Dalpiaz et al., 2018; Wei et al., 2008). These realities frame our scope: we analyze to what extent the conceptual field where compositionality presents and, we only test its linear-additive form defined over a fixed set of conceptual components.

Linear structure across modalities. Early word-analogy work established that semantic relations align with linear directions in Skip-gram vectors (Mikolov et al., 2013b; Levy and Goldberg, 2014; Gittens et al., 2017); similar structures appear in word (Mikolov et al., 2013a,c; Goldberg and Levy, 2014; Seonwoo et al., 2019; Fournier et al., 2020), sentence embeddings (Bowman et al., 2016; Hewitt and Manning, 2019; Li et al., 2020), and representations of LLMs, vision–language models, and other deep learning models (Meng et al., 2022; Hernandez et al., 2024; Trager et al., 2023; Lepori et al., 2023). Park et al. (2024) formalise this “linear representation hypothesis”, tying linear probes to steerable directions via a causal inner product.

Measuring compositionality. Empirical work measures compositional structure in three main ways. (i) *Geometry-based measures:* vector arithmetic in Skip-gram embeddings (Mikolov et al., 2013a), formal additive-offset analyses (Gittens et al., 2017; Seonwoo et al., 2019), linear decomposition for subject, verb and object in a sentence (Xu et al., 2023), graded TRE approximation scores (Andreas, 2019), and higher-order ICA correlations (Oyama et al., 2024) all ask whether representations decompose as (near)linear sums of parts. (ii) *Probing and perturbation:* simple classifiers or representational-similarity scores reveal which linguistic factors a layer encodes (Hewitt and Manning, 2019; Ettinger et al., 2016; Tenney et al., 2019a; Chrupała and Alishahi, 2019; Lepori and McCoy, 2020). (iii) *Behavioral generalization:* model outputs on nonce or zero-shot combinations expose the underlying mechanism, e.g. analogy vs. rules in derivational morphology (Hofmann et al., 2025). Our contribution is a unified and statistically robust two-step diagnostic consisting of linearity quantification and additive generalization. It applies across word, sentence, and knowledge graph embeddings, and supports layer- and time-resolved analysis of compositional structure.

3 Tasks and Formulations

Previous work has explored methods to detect compositionality in embeddings by examining the relationships between entity attributes and their representations, including Linear or Syntactic Probing, Linear Representation Hypothesis, Correlation-Based Linearity Detection, Embedding of Labels and Additive Compositionality Detection (Hewitt and Manning, 2019; Thilak et al., 2024; Xu et al., 2023; Park et al., 2024; Guo et al., 2023). Building on the geometric and behavioral strands surveyed in Section 2, we formalize a two-step, modality-agnostic diagnostic to quantify compositional generalization. First assesses linearity between the attributes matrix and their embedding spaces. Second, tests whether those linear parts combine to predict embeddings for unseen attribute mixes. The pipeline is shown in Figure 1. For each entity, we consider two representations:

- A *binary attribute matrix* $\mathbf{A}$ (e.g., demographics or syntactic properties), where each row is an entity and each column represents an attribute. Entries $a_{ij} \in \{0, 1\}$ indicate the absence or presence of attribute j in entity i.

- A *continuous embedding matrix* $\mathbf{U}$ (e.g., user behaviour embeddings or word embeddings), where each row corresponds to an entity's embedding and each column represents a dimension of the embedding space.

3.1 Assessing Linearity

We employed the Canonical Correlation Analysis (CCA) (Shawe-Taylor et al., 2004; Xu et al., 2023) to uncover latent correlations between the binary attribute matrix $\mathbf{A}$ and the continuous embedding matrix $\mathbf{U}$.

Given the binary attribute matrix $\mathbf{A} \in \{0, 1\}^{q \times n}$ and the embedding matrix $\mathbf{U} \in \mathbb{R}^{q \times m}$, where q is the number of entities, n is the number of attributes, and m is the dimensionality of the embeddings, the goal is to find transformation matrices $\mathbf{W}_A \in \mathbb{R}^{n \times k}$ and $\mathbf{W}_U \in \mathbb{R}^{m \times k}$ that maximize the correlation between the projected data: $\rho = \max_{\mathbf{W}_A, \mathbf{W}_U} \mathrm{corr}\left(\mathbf{A}\mathbf{W}_A, \mathbf{U}\mathbf{W}_U\right)$. Here, corr denotes the Pearson Correlation Coefficient (PCC) between the projected representations, and k is the number of canonical components. The transformations $\mathbf{A}\mathbf{W}_A$ and $\mathbf{U}\mathbf{W}_U$ capture the most correlated aspects of the attributes and embeddings, respectively. We provide a detailed formal exposition of all settings necessary to define a CCA component, including the precise definition, the underlying vector space and mapping of semantic attributes, as well as illustrative examples. A complete derivation and examples are included in the Appendix A.

3.2 Quantifying Additive Generalization

An attribute embeddings matrix $\mathbf{X}$ quantifies how much each attribute influences the embeddings (Kumar et al., 2019; Xu et al., 2023). It assumes that an entity's embedding can be approximated as a linear combination of attribute embeddings (Seonwoo et al., 2019). Given $\mathbf{A}$ and $\mathbf{U}$, the goal is to solve the linear system $\mathbf{A}\mathbf{X} = \mathbf{U}$ where $\mathbf{X} \in \mathbb{R}^{n \times m}$ is the matrix of attribute embeddings to be learned.

We employed the Leave-One-Out (LOO) experiment to quantify the ability of inferring the embedding vector of an unseen entity's attributes. A LOO is performed for each entity i. We firstly exclude

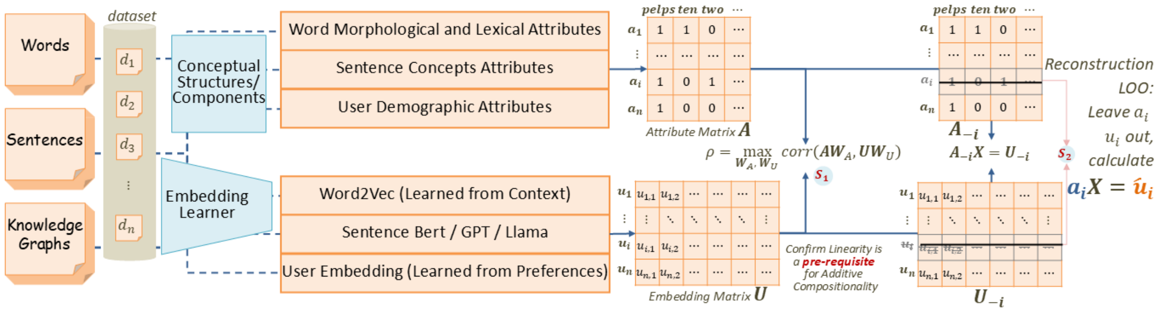

Words
Sentences
Knowledge Graphs
dataset
d_1
d_2
d_3
d_n
Conceptual Structures/ Components
Embedding Learner
Word Morphological and Lexical Attributes
Sentence Concepts Attributes
User Demographic Attributes
Word2Vec (Learned from Context)
Sentence Bert / GPT / Llama
User Embedding (Learned from Preferences)
pelps ten two ...
Attribute Matrix A
Embedding Matrix U
$\rho = \max_{W_A, W_U} corr(AW_A, UW_U)$
s_1
Confirm Linearity is a pre-requisite for Additive Compositionality
A_{-i}
$A_{-i}X = U_{-i}$
U_{-i}
s_2
Reconstruction LOO: Leave a_i u_i out, calculate
$a_iX = \hat{u}_i$

Figure 1: The pipeline of evaluating additive compositionality consists of quantifying linearity and generalizability.

the i-th row from $\mathbf{A}$ and $\mathbf{U}$ to obtain $\mathbf{A}_{-i}$ and $\mathbf{U}_{-i}$, then solve $\mathbf{A}_{-i}\mathbf{X} = \mathbf{U}_{-i}$ using the pseudo-inverse to obtain $\mathbf{X}$. We then estimate the left-out embedding using $\hat{\mathbf{u}}_i = \mathbf{a}_i\mathbf{X}$, where $\mathbf{a}_i$ is the i-th row of $\mathbf{A}$, and finally compare $\hat{\mathbf{u}}_i$ with the actual embedding $\mathbf{u}_i$ using the following metrics: (more details are given in Appendix B)

1. *L2 Loss (Linear System Loss)*: $L_2 = \|\hat{\mathbf{u}}_i - \mathbf{u}_i\|^2$, measuring the reconstruction error.

2. *Embedding Prediction (Cosine Similarity)*: $\cos(\theta) = \dfrac{\hat{\mathbf{u}}_i \cdot \mathbf{u}_i}{\|\hat{\mathbf{u}}_i\| \, \|\mathbf{u}_i\|}$, assessing the alignment between the predicted and actual embeddings.

3. *Identity Prediction (Retrieval Accuracy)*: Determines if $\hat{\mathbf{u}}_i$ correctly identifies entity i by checking if $\mathbf{u}_i$ is the nearest neighbor to $\hat{\mathbf{u}}_i$ among all embeddings.

Hypothesis Testing To determine statistical significance, a non-parametric hypothesis test is performed by directly estimating the p-value through Monte Carlo sampling. First, the test statistic T_{real} is computed for the real pairing: for CCA, this is the canonical correlation ρ_{real}; for additive compositionality, they are L2 loss, cosine similarity and retrieval accuracy based on the real pairing. Next, permuted pairings are generated by randomly shuffling the row order of $\mathbf{A}$ to disrupt the alignment, resulting in permuted datasets $\{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}\}$.

4 Experiments

Before presenting results, we first illustrate how our framework is applied in practice. For sentences, attributes correspond to concepts such as genre, theater name, and location. For knowledge graphs, they correspond to demographic attributes such as gender and age groups. For words, they correspond to morphological components including roots and suffixes. The experiments then test whether embeddings align linearly with these attributes and whether additive combinations can reconstruct unseen entities.

4.1 Dataset and Experiment Setup

Sentence We evaluate the extent to which sentence embeddings from SBERT, GPT, and Llama can be additively decomposed into concepts expressed in the sentence, using a dataset consisting of sentences annotated with concepts, derived from the Schema-Guided Dialogue (SGD) dataset (Rastogi et al., 2020)[1]. An example pair of sentence and concept labels is: **Sentence:** Can you find me an Adventure movie playing at AMC NewPark in Newark? **Concepts:** [*location*, *theater name*, *genre*].

We select 2,458 sentences, each annotated with minimum 3 and maximum 4 concepts from a total set of 47 concepts. The mean number of concepts per sentence is 3.16. There are 90 unique combinations of concepts used. The sentences used comprise the test set of the Schema-Guided Dialogue (SGD) dataset. This split was chosen because it is annotated with denser labels, providing more comprehensive concept coverage.

We first look at the compositionality of sentence embeddings from the last layer of each model. We then examine how compositionality develops during training using the MultiBERTs, which are open reproductions of BERT released with intermediate checkpoints, making them suitable for tracking changes over training. Finally, we analyze compositionality across different layers of SBERT..

Embeddings are generated using pretrained sentence models, producing matrices $\mathbf{U} \in \mathbb{R}^{2,458 \times d}$, where d is the dimensionality of the model. For

[1] https://github.com/google-research-datasets/dstc8-schema-guided-dialogue

SBERT[2] and Llama[3], embeddings are generated by mean pooling over the token embeddings, excluding padding tokens. For GPT[4], sentence embeddings are obtained from the OpenAI API, which provides precomputed embeddings representing the entire sentence. Each row of the matrix $\mathbf{U}$ consists of a sentence embedding for the corresponding sentence. We construct attribute matrices $\mathbf{A} \in \{0,1\}^{2,458\times 47}$ with each row being a binary vector indexing the relevant concepts for the corresponding sentence. In linearity quantification experiments, all sentences are used to compute correlations. In additive compositional generalization experiments, sentences are grouped by their concept combinations, and mean embeddings are computed for each group.

Knowledge Graph We use the MovieLens 1M dataset (Harper and Konstan, 2015) with 6040 users, 3900 movies, and 1 million ratings (1–5). Similar to sentence embedding, we pair each entity, i.e. user, with two descriptions: a binary vector representing demographic attributes (gender, age and occupation) and a user embedding learned from movie preferences (details in Appendix H).

For linearity quantification experiments, the continuous embedding matrix $\mathbf{U}_{corr} \in \mathbb{R}^{6040\times 50}$ is populated with these embeddings. We also generate a binary attribute matrix $\mathbf{A}_{corr} \in \{0,1\}^{6040\times 9}$ containing one 9-dimensional binary attribute vector for each user, including attributes based on gender, age, and occupation. For additive compositional generalization experiments, we generate a binary attribute matrix $\mathbf{A}_{add} \in \{0,1\}^{14\times 9}$ containing one 9-dimensional binary attribute vector for each user including attributes based on gender and age (2 genders × 7 age groups). The continuous embedding matrix $\mathbf{U}_{add} \in \mathbb{R}^{14\times 50}$ is generated by taking the mean across embeddings associated with the same attribute combination.

Table 1: Suffix presence (indicated by '1') for selected words from the MorphoLex dataset

Word	er	t	y	est	ly	ness	less
weightier	1	1	1	0	0	0	0
weightiest	0	1	1	1	0	0	0
weightily	0	1	1	0	1	0	0
weightiness	0	1	1	0	0	1	0
weightlessly	0	1	0	0	1	0	1

Word We use WordNet embeddings as semantic representation, and MorphoLex for syntactic structures to examine semantic and syntactic information contained in word2vec embeddings. By comparing the word2vec embeddings against both WordNet and MorphoLex, we are able to disentangle the semantic and syntactic aspects of the word2vec representation. Training details for WordNet are provided in Appendix F.1. using vectors from GoogleNews-vectors-negative300, We build two continuous embedding matrices $\mathbf{U}_{w2v-wn} \in \mathbb{R}^{25,781\times 300}$ and $\mathbf{U}_{wn-w2v} \in \mathbb{R}^{25,781\times 20}$.

MorphoLex (Sánchez-Gutiérrez et al., 2018) provides a standardized morphological database derived from the English Lexicon Project, encompassing 68,624 words with nine variables for roots and affixes. An example extract from the dataset is given in Table 1. In this paper, we focus specifically on words with one root and multiple suffixes. For linearity quantification experiments, 15,342 words and 81 suffixes are selected by intersecting the MorphoLex and word2vec (GoogleNews-vectors-negative300) vocabularies and filtering out suffixes occurring fewer than 10 times. We build a continuous embedding matrix $\mathbf{U}_{w2v-morpho} \in \mathbb{R}^{15,342\times 300}$ using vectors from GoogleNews-vectors-negative300 and a binary embedding matrix $\mathbf{X}_{morpho} \in \{0,1\}^{15,342\times 81}$.

To examine the additive compositionality of word2vec embeddings across multiple suffixes, we select 278 words as follows. We filter words that have exactly 3 suffixes, each of which occurs 10 times or more[5]. This results in 17 suffixes, but some words end up with fewer than 3 suffixes because certain suffixes occur less than 10 times and are filtered out. We build a continuous embedding matrix $\mathbf{U}_{w2v-add} \in \mathbb{R}^{278\times 300}$ using vectors from GoogleNews-vectors-negative300. We build a binary attribute matrix $\mathbf{A}_{morpho-add} \in \mathbb{R}^{278\times 45}$ indicating the presence or absence of each root word and morphemes.

4.2 Quantifying Linearity

Sentence For SBERT and GPT embeddings[6], we apply CCA to assess whether sentence embeddings correlate with their binary concept vectors. Figures 2a and 2b show significant difference in correlation scores between real and permuted pairings (p-value < 0.01) for both GPT and SBERT embeddings,

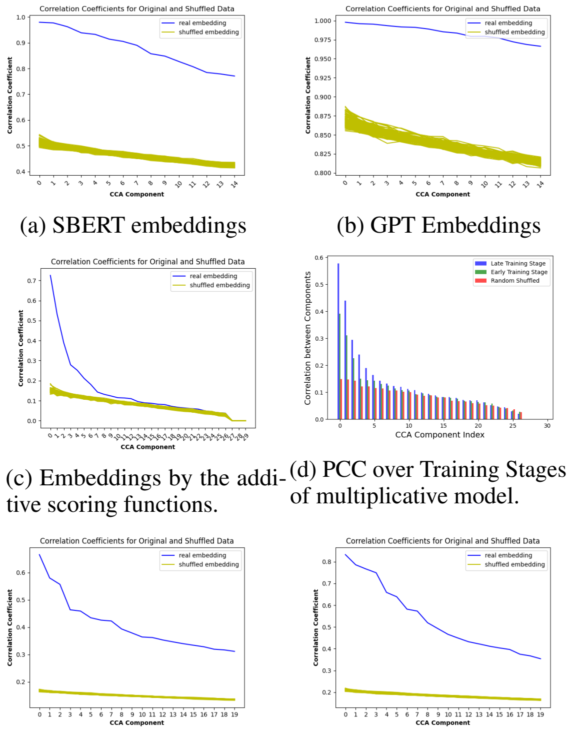
[2] sentence-transformers/all-MiniLM-L6-v2

[3] meta-llama/Llama-2-7b-hf

[4] text-embedding-3-small

[5] These words are listed in Appendix G

[6] Correlation for Llama embeddings is not assessed due to high dimensionality; see Limitations.



(a) SBERT embeddings

(b) GPT Embeddings

(c) Embeddings by the additive scoring functions.

(d) PCC over Training Stages of multiplicative model.

(e) WordNet-word2vec

(f) MorphoLex-word2vec

Figure 2: PCC (y-axis) between real (blue) and permuted (yellow) pairings across canonical components k (x-axis). **Panels**: (a,b) sentence–concept (100 perms); (c) user–attribute (100 perms); (d) blue-late, green-early, red-permuted (e,f) 50 perms. PCC is computed between projected vectors **A** and **U**; **real embedding always higher than the permuted baselines**.

indicating that sentence embeddings are correlated with their binary concept vectors.

Knowledge Graph Results of the correlation-based linearity analysis is reported in Figures 2c for additive (TransE) scoring functions computed embeddings. As measured by CCA, the PCC for true user-attribute pairings substantially exceeds that for randomly permuted pairings, especially in the early canonical components (approximately three times). Moreover, we extract embeddings from both early (5 epochs) and late (300 epochs) stages of training to compare correlation-based linearity over time. The results show that the Pearson correlation coefficient (PCC) at the later stage is approximately 1.5 times higher than at the early stage.

Word As shown in Figures 2e and 2f, the Pearson Correlation Coefficient (PCC) between word2vec, WordNet embeddings, and MorphoLex binary vectors substantially exceeds the randomized baseline, indicating that word2vec embeddings, trained on contextual co-occurrences, implicitly capture both semantic and morphological information. Results are reported in Figure 2d. We see that as training progresses, the correlation coefficient increases, showing that more demographic information becomes encoded into the embeddings.

Table 2: Generalizability metrics for SBERT, GPT, and Llama embeddings. Figures in **Real** columns are the mean across all leave-one out experiments. Figures in **Permuted** columns are the mean across 100 permutations of the sentence-concept pairs. All increases of Real over Permuted values are significant with p<0.01.

Embed -ding	**Cosine Real**	**Cosine Permuted**	**Hits@5 Real**	**Hits@5 Permuted**
SBERT	0.7761	0.4865	0.59	0.0405
GPT	0.7753	0.4941	0.57	0.0399
Llama	0.9355	0.8153	0.52	0.0468

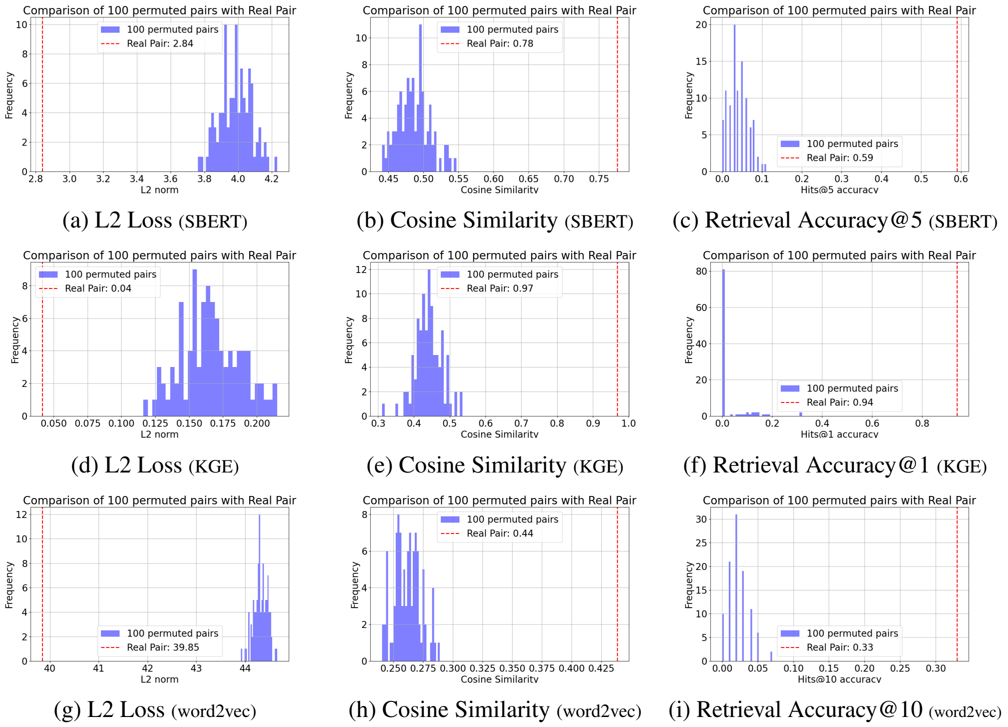
4.3 Quantifying Additive Generalization

Once we confirm the conceptual components are linearly correlated with the embedding matrix, we can then quantify the additive generalizability.

Sentence Results are reported in Table 2. Cosine similarity and retrieval accuracy for the true pairings are all significantly and substantially higher than for the permuted pairings, suggesting the additive compositionality in sentence embeddings can be generalized to unseen combination of concepts. The behaviour of SBERT and GPT embeddings are very similar, whereas Llama embeddings have a less substantial increase over permuted pairings in embedding prediction and retrieval accuracy (Hits@5). The distribution of the metric values for permuted pairs vs. real pairs for SBERT embeddings are plotted in Figures 3a, 3b, 3c. Results for GPT and Llama embedding are provided in Appendix D, Figures 5 and 6.

Knowledge Graph Results are reported in Figures 3d, 3e, 3f, showing L2 loss of 0.04, cosine similarity of 0.97, and Hits@1 of 0.94, each outperforming the random baseline with $p = 0.01$. These findings reject the null hypothesis, demonstrating that user embeddings generalize additive relationships to new users' gender and age attributes.

Word Results are presented in Figures 3g, 3h, 3i. We see that word2vec embeddings can be decomposed into root and multiple suffixes fairly well, and be generalized to unseen root-suffix combinations. The linear system loss is 39.85, lower than the minimum loss of the random system (43.91). Cosine similarity is 0.44, greater than all instances of the random baseline, and retrieval accuracy @10



(a) L2 Loss (SBERT) (b) Cosine Similarity (SBERT) (c) Retrieval Accuracy@5 (SBERT)

(d) L2 Loss (KGE) (e) Cosine Similarity (KGE) (f) Retrieval Accuracy@1 (KGE)

(g) L2 Loss (word2vec) (h) Cosine Similarity (word2vec) (i) Retrieval Accuracy@10 (word2vec)

Figure 3: Test statistics (y-axis) for embedding decompositions. Dashed line = mean scores of the predicted $\hat{\mathbf{u}}$ learned from real embedding pairs; bars = scores from 100 random permutations. Rows: SBERT, Knowledge Graph Embeddings, and word2vec. **Real pairings achieve higher cosine similarity and retrieval accuracy with lower L2 loss than permuted baselines**.

is greater than that of the random system. However, overall these values are low, showing that there is still a fair bit of information that is not captured by this representation.

Table 3: Additive compositionality metrics across training steps of BERT. Number of training steps is given by the suffix to the model, e.g. cls_20k indicates the performance of the model after 20k training steps. Rel. Diff expresses this difference as a percentage of the permuted similarity, showing proportional improvement. Figures in Real columns are the mean across all leave-one out experiments. Figures in Permuted columns are the mean across 100 permutations of the sentence-concept pairs.

Model	Cosine Real	Cosine Permuted	Cosine Rel. Diff	Hits @5 Real	L2 Loss Real
cls_0k	0.9884	0.9882	0.02%	0.44	29.39
cls_20k	0.8787	0.7722	13.79%	0.55	65.46
cls_40k	0.8773	0.7724	13.59%	0.48	62.48
cls_100k	0.9201	0.8323	10.54%	0.55	47.16
cls_1000k	0.9545	0.9149	4.33%	0.44	21.16
cls_2000k	0.9538	0.9094	4.89%	0.48	21.07

4.3.1 Comparison across Training Stages

Sentence We further compare additive compositional generalization across different training stages of the MultiBERTs (Sellam et al., 2022). We again use the same binary attribute matrix $\mathbf{A}$ at each training stage. We build continuous embedding matrices $\mathbf{U}_{steps}$ for 0, 20k, 40k, 100k, 1000k, and 2000k training steps. We use the [CLS] token to represent sentences. As in Table 3, at 0k steps, there is no substantial differences between the values of the cosine similarity metrics for real and permuted despite high overall values (0.9884). This indicates that the embeddings lack differentiation and do not capture conceptual relationships. After 20k training steps, generalization metrics (Cosine Real, L2 loss) constantly improve, showing generalization of additive compositionality emerges from training.

4.3.2 Comparison between Different Layers

Sentence We assess the additive compositional generalization of sentence embeddings derived from each layer of SBERT. At each layer, the binary attribute matrix $\mathbf{A}$ remains the same as in the last layer experiment. Separate continuous embedding matrices $\mathbf{U}_i$ are built for each layer i. To generate sentence embeddings at each layer, we extract hidden states for all layers from the SBERT model and apply the same pooling method as used for the last layer (mean pooling over token embeddings). Each embedding is normalized to length 1.

The results, reported in Table 4, show that com-

Table 4: Additive compositionality metrics across SBERT layers. Figures in **Real** columns are the mean across all leave-one out experiments. Figures in **Permuted** columns are the mean across 100 permutations of the sentence-concept pairs. All Real vs. Permuted increases are significant at $p < 0.01$.

Layer	**Cosine Real**	**Cosine Permuted**	**Hits@5 Real**	**L2 Loss Real**
0	0.8889	0.7808	0.59	2.23
1	0.9366	0.8671	0.57	1.80
2	0.9397	0.8576	0.61	1.76
3	0.9403	0.8330	0.64	1.73
4	0.9408	0.8298	0.66	1.72
5	0.9409	0.8273	0.62	1.71
6	0.7761	0.4865	0.59	2.84

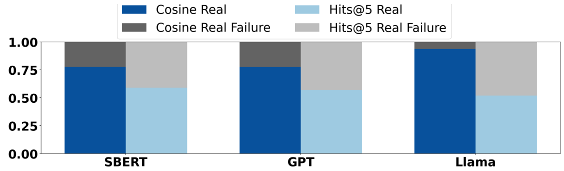


Figure 4: Comparison between SBERT, GPT, Llama Failures in Compositionality, failures are the cases that embeddings cannot be decomposed into components or vice versa.

positionality generalization increases through the layers, peaking at layer 4 or 5. However, an abrupt drop in compositionality generalization, as measured by cosine similarity, is observed at the last layer. This is in line with the phenomenon that semantic information is better encoded at earlier layers in the model, see for example experiments in the original BERT paper (Devlin et al., 2018). Even though the cosine metric is fairly high for the random baseline, the discrimination between sentence meanings is still high as can be seen by fact that the Hits@5 for real pairings is consistently substantially higher than that for permuted pairings.

5 Discussion and Conclusion

Linearity and Compositionality Our paper adopted a two-step approach to explore the inherent structure of embeddings. In the first step, quantifying linearity confirmed a robust linear correlation between embeddings and candidate semantic features. This step established the necessary foundation for the second step, quantifying generalizability, where we rigorously demonstrated that final embeddings can be reconstructed as an additive combination of concept embeddings.

Our findings address the broader question of whether relations between structured entities can be captured through simple vector operations. Evaluating a range of word, sentence, and graph embeddings showed a common thread of additive compositionality and therefore shows the generalizability. For sentence embeddings, conceptual components can often be combined additively, indicating that even large, non-linear models (e.g., GPT or Llama) retain interpretable vector structures. Word embeddings effectively encode both morphological and semantic relationships, as illustrated by transformations like *weight* + *y* + *ly* = *weightily*.

Quantifying Compositionality The degree of compositionality increases across models regardless of training strategy. In MultiBERTs, compositionality steadily increase with training (see Cosine Real and L2 Norm in Table 3). In knowledge graph embeddings, the trends are similar. The high "Cosine Real score" observed with the initial, untrained embeddings (0k model) is noteworthy; however, it can be understood as a consequence of the model's initialization. This is because there is no significant statistical difference when comparing the "real pair" scores to those from "permuted pairs," which serve as a baseline. On the other hand, in SBERT, compositionality increases in earlier layers but declines slightly in later layers, presumably because these upper layers specialize in task-specific representations (Devlin et al., 2018; Tenney et al., 2019b).

Understanding Compositional Failures Our framework not only detects successful instances of additive compositionality but also quantifies failure cases where reconstructions significantly diverge from the ground truth. These residuals highlight the limits of linear compositionality and are essential for identifying semantic phenomena, such as non-linear interactions or context-dependent meanings that cannot be captured by known attributes. For instance, we observed fluctuations in the Hits@5 score across training stages and transformer layers. One contributing factor may be inaccuracies in the conceptual representation of sentences (e.g., a theater name inadvertently revealing location information) an inherent challenge when mapping natural language to structured concepts.

We also compared multiple sentence embedding models, all of which exhibit some degree of compositionality failure when evaluated against an idealized "perfect compositionality" scenario. In such a case, the model would achieve perfect accuracy, and additive compositions would result in no residuals. These findings, presented in Figure 4 ,underscore the limitations of current models and motivate future research into more expressive, non-linear approaches to compositional representation.

Overall, the results show that embeddings across modalities retain meaningful additive structure, supporting interpretability in representation learning. At the same time, our framework shows consistent failures of linear decomposition in both static and transformer models, leaving residuals that point to unresolved semantic complexity and motivate future research.

Conclusion We formalized a unified evaluation pipeline that aligns attribute matrices with embedding spaces via CCA and measures additive generalization through leave-one-out reconstruction. Applied to word, sentence, and knowledge graph embeddings, our experiments show that embeddings exhibit additive compositionality and can generalize to novel attribute combinations, though they also reveal consistent failure cases across models where linear composition breaks down. Our analysis quantifies such failure, underscoring the limits of linear composition. In transformers, the additive compositionality peaks in mid-to-late layers but declines in the last layer due to task specialization. We will release our code to support future work on more expressive, non-additive representations.

Limitations

1. Conceptual Structure: Conceptual structures (such as concepts in sentence embeddings or user demographic information in knowledge graph embeddings) are rooted in human understanding. As a result, failures in additive compositionality may stem from various factors, including inherent non-linearity, missing components, or incomplete representations.

2. Linearity assumption: As mentioned above, our method concentrates on detecting linearity-based compositionality, more methods are needed to handle non-linear cases.

3. The curse of dimensionality: Due to the large size of LLama embeddings, we did not compute correlation metrics. In future work, we plan to apply dimensionality reduction techniques, such as those based on the Johnson–Lindenstrauss lemma, to preserve statistical properties while improving computational efficiency and stability.

Acknowledgments

We would like to express our gratitude to Professor Nello Cristianini for his invaluable guidance and insightful discussions throughout the development of this work. His thoughtful feedback and generous time in repeatedly discussing the research ideas have greatly shaped and improved this study.

References

Jacob Andreas. 2019. Measuring compositionality in representation learning. *arXiv preprint arXiv:1902.07181*.

Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Avishek Bose and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pages 715–724. PMLR.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Grzegorz Chrupała and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. *arXiv preprint arXiv:1905.06401*.

Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the workshop on a broader perspective on multiword expressions*, pages 41–48.

Fabiano Dalpiaz, Ivor Van der Schalk, and Garm Lucassen. 2018. Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and nlp. In *Requirements Engineering: Foundation for Software Quality: 24th International Working Conference, REFSQ 2018, Utrecht, The Netherlands, March 19-22, 2018, Proceedings 24*, pages 119–135. Springer.

Verna Dankers and Christopher Lucas. 2023. Non-compositionality in sentiment: New data and analyses. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5150–5162.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv preprint arXiv:1810.04805.

Eric Elmoznino, Thomas Jiralerspong, Yoshua Bengio, and Guillaume Lajoie. 2024. A complexity-based theory of compositionality. *arXiv preprint arXiv:2410.14817*.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st workshop on evaluating vector-space representations for nlp*, pages 134–139.

Louis Fournier, Emmanuel Dupoux, and Ewan Dunbar. 2020. Analogies minus analogy test: measuring regularities in word embeddings. In *CoNLL 2020-24th Conference on Computational Natural Language Learning*.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. 2017. Skip-gram- zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Chenfeng Guo and Dongrui Wu. 2019. Canonical correlation analysis (cca) based multi-view learning: An overview. *arXiv preprint arXiv:1907.01693*.

Zhijin Guo, Zhaozhen Xu, Martha Lewis, and Nello Cristianini. 2023. Extract: Explainable transparent control of bias in embeddings. In *Proceedings of the 1st Workshop on Fairness and Bias in AI co-located with 26th European Conference on Artificial Intelligence (ECAI 2023)*.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 139–144, Brussels, Belgium. Association for Computational Linguistics.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Ray Jackendoff, Rochelle Lieber, and Pavol Štekauer. 2011. Compounding in the parallel architecture and conceptual semantics. In *The Oxford Handbook of Compounding*, Oxford Handbooks in Linguistics. Oxford University Press.

Vikas Kumar, Arun K Pujari, Vineet Padmanabhan, and Venkateswara Rao Kagita. 2019. Group preserving label embedding for multi-label classification. *Pattern Recognition*, 90:23–34.

Michael Lepori and R. Thomas McCoy. 2020. Picking BERT's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael Lepori, Thomas Serre, and Ellie Pavlick. 2023. Break it down: Evidence for structural compositionality in neural networks. *Advances in Neural Information Processing Systems*, 36:42623–42660.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 317–324.

Diana McCarthy, Sriram Venkatapathy, and Aravind Joshi. 2007. Detecting compositionality of verb-object combinations using selectional preferences. In *Proceedings of the 2007 Joint Conference on*

Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 369–379.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*.

Momose Oyama, Hiroaki Yamagiwa, and Hidetoshi Shimodaira. 2024. Understanding higher-order correlations among semantic components in embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2883–2899.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pages 39643–39666. PMLR.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Claudia H Sánchez-Gutiérrez, Hugo Mailhot, S Hélène Deacon, and Maximiliano A Wilson. 2018. Morpholex: A derivational morphological database for 70,000 english words. *Behavior research methods*, 50:1568–1580.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, and 1 others. 2022. The multiberts: Bert reproductions for robustness analysis. In *10th International Conference on Learning Representations, ICLR 2022*.

Jose A Seoane, Colin Campbell, Ian NM Day, Juan P Casas, and Tom R Gaunt. 2014. Canonical correlation analysis for gene-based pleiotropy discovery. *PLoS computational biology*, 10(10):e1003876.

Yeon Seonwoo, Sungjoon Park, Dongkwan Kim, and Alice Oh. 2019. Additive compositionality of word vectors. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 387–396.

John Shawe-Taylor, Nello Cristianini, and 1 others. 2004. *Kernel methods for pattern analysis*. Cambridge university press.

Pasi Tapanainen, Jussi Piitulainen, and Timo Jarvinen. 1998. Idiomatic object usage and support verbs. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, and 1 others. 2019a. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, and 1 others. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019*.

Vimal Thilak, Chen Huang, Omid Saremi, Laurent Dinh, Hanlin Goh, Preetum Nakkiran, Joshua M Susskind, and Etai Littwin. 2024. Lidar: Sensing linear probing performance in joint embedding ssl architectures. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and Stefano Soatto. 2023. Linear spaces of meanings: compositional structures in vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15395–15404.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

villmow. 2019. Datasets for knowledge graph completion with textual information about entities. Github.

Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. 2002. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Tianqi Wang and Xu Xu. 2024. Composition as nonlinear combination in semantic space: Exploring the effect of compositionality on chinese compound recognition. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Shikui Wei, Yao Zhao, and Zhenfeng Zhu. 2008. Ontology-based inter-concept relation fusion for concept detection. In *Advances in Multimedia Information Processing-PCM 2008: 9th Pacific Rim Conference on Multimedia, Tainan, Taiwan, December 9-13, 2008. Proceedings 9*, pages 721–730. Springer.

Zhaozhen Xu, Zhijin Guo, and Nello Cristianini. 2023. On compositionality in data embedding. In *Advances in Intelligent Data Analysis XXI: 21st International Symposium, IDA 2023*. Springer.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Majid Yazdani, Meghdad Farahmand, and James Henderson. 2015. Learning semantic composition to detect non-compositionality of multiword expressions. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1733–1742.

A Details of Correlation-Based Compositionality Detection Method

Canonical Correlation Analysis (CCA) is used to measure the correlation information between two multivariate random variables (Shawe-Taylor et al., 2004). Just like the univariate correlation coefficient, it is estimated on the basis of two aligned samples of observations.

A matrix of binary-valued attribute embeddings, denoted as $\mathbf{A}$, is essentially a matrix representation where each row corresponds to a specific attribute and each column corresponds to an individual data point (such as a word, image, or user). The entries of the matrix can take only two values, typically 0 or 1, signifying the absence or presence of a particular attribute. For example, in the context of textual data, an attribute might represent whether a word is a noun or not, and the matrix would be populated with 1s (presence) and 0s (absence) accordingly.

On the other hand, a matrix of user embeddings, denoted as $\mathbf{U}$, is a matrix where each row represents an individual user, and each column represents a certain feature or dimension of the embedding space. These embeddings are continuous-valued vectors that capture the movie preference of the users. The values in this matrix are not constrained to binary values and can span a continuous range.

These paired random variables are often different descriptions of the same object, for example genetic and clinical information about a set of patients (Seoane et al., 2014), French and English translations of the same document (Vinokourov et al., 2002), and even two images of the same object from different angles (Guo and Wu, 2019).

In the example of viewers and movies, we use this method to compare two descriptions of users. One matrix is based on demographic information, which are indicated by Boolean vectors. The other matrix is based on their behaviour, which is computed by their movie ratings only.

Assuming we have a vector for an individual user's attribute embedding, denoted as:

$$\mathbf{a} = (a_1, a_2, \ldots, a_n)^T$$

and a corresponding individual user's computed embedding:

$$\mathbf{u} = (u_1, u_2, \ldots, u_m)^T$$

we aim to explore the correlation between these two representations across multiple users. Given a set of q users, we define $\mathbf{A}$ as a $q \times n$ matrix where each row corresponds to the attribute embeddings for a specific user, and $\mathbf{U}$ as a $q \times m$ matrix where each row represents the computed embedding of the same user. Here, n is the number of attributes, and m is the dimensionality of the user embeddings.

Canonical Correlation Analysis (CCA) is then employed to find the projection matrices $\mathbf{W}_A \in \mathbb{R}^{n\times k}$ and $\mathbf{W}_U \in \mathbb{R}^{m\times k}$ that maximise the correlation between the transformed representations of $\mathbf{A}$ and $\mathbf{U}$. Each projection matrix contains k-projection vectors, where k is the number of canonical components that depend on the eigenvalues of the covariance matrix. For each k-th canonical component, the projections of the attribute and user embeddings are given by:

$$\mathbf{A}_k = \mathbf{A}\mathbf{w}_{a_k} \quad \text{and} \quad \mathbf{U}_k = \mathbf{U}\mathbf{w}_{u_k}$$

where $\mathbf{w}_{a_k} \in \mathbb{R}^n$ and $\mathbf{w}_{u_k} \in \mathbb{R}^m$ are the k-th projection vectors from the matrices $\mathbf{W}_A$ and $\mathbf{W}_U$, respectively. $\mathbf{A}_k \in \mathbb{R}^q$ represents the projection of the original attribute matrix $\mathbf{A}$ (size $q \times n$) onto the k-th canonical direction, using the projection vector $\mathbf{w}_{a_k} \in \mathbb{R}^n$. It results in a vector of size $q \times 1$ that contains the transformed values for each user for the k-th component. Similarly, $\mathbf{U}_k \in \mathbb{R}^q$ represents the projection of the original user embedding matrix $\mathbf{U}$ (size $q \times m$) onto the k-th canonical direction, using the projection vector $\mathbf{w}_{u_k} \in \mathbb{R}^m$. It also results in a vector of size $q \times 1$ that contains the transformed values for each user for the k-th component.

The goal of CCA is to maximise the Pearson Correlation Coefficient (PCC) between these transformed representations, i.e., between $\mathbf{A}_k$ and $\mathbf{U}_k$, for each k-th canonical component. The correlation for the k-th canonical component, denoted as ρ_k, is given by the formula:

$$\rho_k = \frac{\sum_{i=1}^{q} \left((\mathbf{A}_k)_i - \mu_{\mathbf{A}_k}\right)\left((\mathbf{U}_k)_i - \mu_{\mathbf{U}_k}\right)}{\sqrt{\sum_{i=1}^{q}\left((\mathbf{A}_k)_i - \mu_{\mathbf{A}_k}\right)^2}\sqrt{\sum_{i=1}^{q}\left((\mathbf{U}_k)_i - \mu_{\mathbf{U}_k}\right)^2}} \tag{1}$$

where $\mu_{\mathbf{A}_k}$ and $\mu_{\mathbf{U}_k}$ are the means of the transformed attribute and user embeddings, respectively, and q is the number of users.

In matrix form, we can express this objective as maximizing the correlation between the transformed matrices $\mathbf{A}\mathbf{W}_A$ and $\mathbf{U}\mathbf{W}_U$. This is formalized as:

$$\rho = \max_{\mathbf{W}_A, \mathbf{W}_U} \operatorname{corr}\left(\mathbf{A}\mathbf{W}_A, \mathbf{U}\mathbf{W}_U\right) \tag{2}$$

where the correlation is maximised across the projection matrices $\mathbf{W}_A$ and $\mathbf{W}_U$, and the result is a set of K-canonical correlations $\rho_1, \rho_2, \ldots, \rho_k$ that describe the relationship between the attribute embeddings and the user embeddings for the entire dataset.

Thus, by computing the canonical correlations ρ_K for each component, we obtain insights into how well the attribute embeddings and computed user embeddings are aligned in terms of their underlying structure.

Consider the case where we have 4 users. The attribute matrix $\mathbf{A}$ has dimensions 4×4, representing 4 users and 4 attributes, and the user embedding matrix $\mathbf{U}$ has dimensions 4×3, representing 4 users and 3 embedding dimensions. Thus: - $\mathbf{A} \in \mathbb{R}^{4\times 4}$: the attribute matrix where each row represents the 4-dimensional attribute vector for a user. - $\mathbf{U} \in \mathbb{R}^{4\times 3}$: the user embedding matrix where each row represents the 3-dimensional embedding vector for a user.

Canonical Correlation Analysis (CCA) is employed to find the projection matrices $\mathbf{W}_A \in \mathbb{R}^{4\times k}$ and $\mathbf{W}_U \in \mathbb{R}^{3\times k}$ that maximize the correlation between the transformed representations of $\mathbf{A}$ and $\mathbf{U}$. Each projection matrix contains k-projection vectors, where k is the number of canonical components that depend on the eigenvalues of the covariance matrix. For each k-th canonical component, the projections of the attribute and user embeddings are given by:

$$\mathbf{A}_k = \mathbf{A}\mathbf{w}_{a_k} \quad \text{and} \quad \mathbf{U}_k = \mathbf{U}\mathbf{w}_{u_k}$$

where $\mathbf{w}_{a_k} \in \mathbb{R}^4$ and $\mathbf{w}_{u_k} \in \mathbb{R}^3$ are the k-th projection vectors from the matrices $\mathbf{W}_A$ and $\mathbf{W}_U$, respectively.

B Leave-one-out experiment

1. **Leave Out**: Exclude the i-th row from $\mathbf{A}$ and $\mathbf{U}$ to obtain $\mathbf{A}_{-i}$ and $\mathbf{U}_{-i}$.

2. **Train**: Solve $\mathbf{A}_{-i}\mathbf{X} = \mathbf{U}_{-i}$ using the pseudo-inverse to obtain $\mathbf{X}$.

3. **Predict**: Estimate the left-out embedding using $\hat{\mathbf{u}}_i = \mathbf{a}_i\mathbf{X}$, where $\mathbf{a}_i$ is the attribute vector of entity i.

4. **Evaluate**: Compare $\hat{\mathbf{u}}_i$ with the actual embedding $\mathbf{u}_i$ using the following metrics:

   (a) *L2 Loss*: $L_2 = \|\hat{\mathbf{u}}_i - \mathbf{u}_i\|^2$, measuring the reconstruction error.

   (b) *Embedding Prediction (Cosine Similarity)*: $\cos(\theta) = \dfrac{\hat{\mathbf{u}}_i \cdot \mathbf{u}_i}{\|\hat{\mathbf{u}}_i\| \, \|\mathbf{u}_i\|}$, assessing the alignment between the predicted and actual embeddings.

   (c) *Identity Prediction (Retrieval Accuracy)*: Determines if $\hat{\mathbf{u}}_i$ correctly identifies entity i by checking if $\mathbf{u}_i$ is the nearest neighbor to $\hat{\mathbf{u}}_i$ among all embeddings.

C Knowledge Graph Embedding

Knowledge Graph Embedding A *graph* $G = (V, E)$ consists of a set of vertices V with edges E between pairs of vertices. In a *knowledge graph*, the vertices V represent entities in the real world, and the edges E encode that some relation holds between a pair of vertices. As a running example, we consider the case where the vertices V are a set of viewers and films, and the edges E encode the fact that a viewer has rated a film.

Knowledge Graphs represent information in terms of entities (or nodes) and the relationships (or edges) between them. The specific relation r that exists between two entities is depicted as a directed edge, and this connection is represented by a triple (h, r, t). In this structure, we distinguish between the two nodes involved: the *head* (h) and the *tail* (t), represented by vectors $\mathbf{h}$ and $\mathbf{t}$ respectively. Such a triple is termed a *fact*, denoted by f:

$$f = (h, r, t)$$

Embedding Function. A knowledge graph $G = (V, E)$ can be embedded by assigning each node $v \in V$ a vector $\mathbf{x}_v \in \mathbb{R}^d$. This embedding function, $\Phi_{KG} : V \to \mathbb{R}^d$, maps nodes into a continuous space where their relationships are captured by a scoring or distance function. For instance, one could define a threshold θ such that an edge $(v_i, v_j) \in E$ exists if and only if $D(\mathbf{x}_{v_i}, \mathbf{x}_{v_j}) < \theta$. Conversely, given a set of embedded points, links between nodes can be recovered by applying a learned scoring function $S\big(\mathbf{x}_{v_i}, \mathbf{x}_{v_j}\big)$.

C.1 Multiplicative Scoring

Nickel et al. (2011) introduced a tensor-factorization approach for relational learning, treating each frontal slice of a three-dimensional tensor as a co-occurrence matrix for a given relation. In this model, a triple (h, r, t) is scored using embeddings $\mathbf{h}, \mathbf{R}, \mathbf{t}$, where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ represent head and tail entities, and $\mathbf{R} \in \mathbb{R}^{d \times d}$ represents the relation:

$$S(h, r, t) \;=\; \mathbf{h}^T \mathbf{R}\, \mathbf{t}. \qquad (3)$$

Various specializations exist, such as DistMult (Yang et al., 2014), which restricts $\mathbf{R}$ to a diagonal matrix (reducing overfitting), and ComplEx (Trouillon et al., 2016), which employs complex-valued embeddings for asymmetric relations. In this work, we adopt DistMult due to its simplicity and scalability, particularly its suitability for large knowledge graphs.

C.2 Additive Scoring

TransE (Bordes et al., 2013) introduces a translation-based perspective, where each relation is a vector that shifts the embedding of the head entity to the tail entity. A triple (h, r, t) is scored by:

$$S(h, r, t) \;=\; \|\mathbf{h} + \mathbf{r} \;-\; \mathbf{t}\|, \quad \mathbf{h}, \mathbf{r}, \mathbf{t} \;\in\; \mathbb{R}^d. \qquad (4)$$

For example, *King* + *FemaleOf* $\approx$ *Queen*. This translation idea captures relational semantics by minimizing the distance between $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$.

Rating Prediction In alignment with (Berg et al., 2017), we establish a function P that, given a triple of embeddings $(\mathbf{h}, \mathbf{R}, \mathbf{t})$, calculates the probability of the relation against all potential alternatives.

$$P\left(\mathbf{h}, \mathbf{R}, \mathbf{t}\right) = \text{SoftArgmax}(S(f)) = \frac{e^{S(f)}}{e^{S(f)} + \sum_{r' \neq r \in \mathscr{R}} e^{S(f')}} \qquad (5)$$

In the above formula, $f = (h, r, t)$ denotes a true triple, and $f' = (h, r', t)$ denotes a corrupted triple, that is a randomly generated one, that we use as a proxy for a negative example (a pair of nodes that are not connected).

Assigning numerical values to relations r, the predicted relation is then just the expected value prediction $= \sum_{r \in \mathscr{R}} r P\left(\mathbf{h}, \mathbf{R}, \mathbf{t}\right)$ In our application of viewers and movies, the set of relations $\mathscr{R}$ could be the possible ratings that a user can give a movie. The predicted rating is then the expected value of the ratings, given the probability distribution produced by the scoring function. $S(f)$ refers to the scoring function in Yang et al. (2014).

To learn a graph embedding, we follow the setting of Bose and Hamilton (2019) as follows,

$$L = -\sum_{f \in \mathscr{F}} \log \frac{e^{S(f)}}{e^{S(f)} + \sum_{f' \in \mathscr{F}'} e^{S(f')}} \qquad (6)$$

This loss function maximizes the probabilities of true triples (f) and minimizes the probability of triples with corrupted triples: (f').

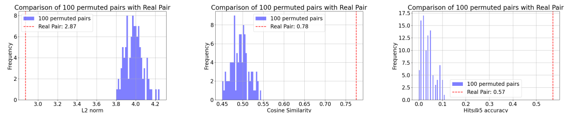
Evaluation Metrics We use 4 metrics to evaluate our performance on the link prediction task. These are root mean square error (RMSE, $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}$, where $\hat{y}_i$ is our predicted relation and y_i is the true relation), Hits@K - the probability that our target value is in the top K predictions, mean rank (MR) - the average ranking of each prediction, and mean reciprocal rank (MRR) to evaluate our performance on the link prediction task. These are standard metrics in the knowledge graph embedding community.

D Additive Compositionality by Model

E Compositionality across Layers and Training Stages

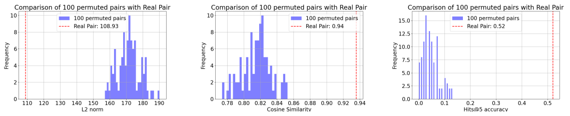
E.1 Comparison of Different Layers

Comparison Metrics To fairly compare different layers, we cannot rely solely on raw cosine



(a) Linear System Loss (b) Cosine Similarity (c) Retrieval Accuracy@5

Figure 5: Test statistics for GPT embedding decomposition. Dashed line is the average performance of $\hat{\mathbf{U}}$ learned from the user embedding. Bars are the distribution of the results from 100 random permutations.



(a) Linear System Loss (b) Cosine Similarity (c) Retrieval Accuracy@5

Figure 6: Test statistics for Llama embedding decomposition. Dashed line is the average performance of $\hat{\mathbf{U}}$ learned from the user embedding. Bars are the distribution of the results from 100 random permutations.

similarities or retrieval accuracies due to variations in scales and distributions. Instead, we use normalized metrics for comparability. The **Normalized Cosine Similarity** computes the difference between the mean real similarity and the mean permuted similarity, normalized by the maximum possible difference, $(1 - \text{Mean Permuted Similarity})$. The **Absolute Difference** is a simple measure of the difference between mean real and permuted similarities. Lastly, the **Relative Difference (Percentage Improvement)** expresses this difference as a percentage of the permuted similarity, indicating proportional improvement. These metrics enable robust and fair comparisons across models.

F Word Embeddings

F.1 WordNet Embedding

WordNet (Miller, 1995) is a large lexical database of English that combines dictionary and thesaurus features with a graph structure. It organizes nouns, verbs, adjectives, and adverbs into synsets (sets of cognitive synonyms) interlinked by semantic and lexical relations. We use WN18RR (Dettmers et al., 2018), a subset of WordNet with 40,943 entities and 11 types of relation. For linearity experiments, we select 25,781 words from the intersection of the WN18RR and word2vec (GoogleNews-vectors-negative300) vocabularies. We train embeddings for these words over WordNet on an entity prediction task, which involves predicting the tail entity given a head entity and relation. Training details are provided in Appendix F.1. We build a continuous embedding matrix $\mathbf{U}_{w2v-wn} \in \mathbb{R}^{25,781\times 300}$ using vectors from GoogleNews-vectors-negative300, and a second continuous embedding matrix $\mathbf{U}_{wn-w2v} \in \mathbb{R}^{25,781\times 20}$. We want to ensure our WordNet embedding can contain the semantic relation in it. Therefore, we train the embedding with the task of predicting the tail entity given a head entity and relation. For example, we might want to predict the hypernym of cat:

$$< \textbf{cat}, hypernym, \textbf{?} >$$

Mapping Freebase ID to text WordNet is constructed with Freebase ID only, an example triple could be <00260881, hypernym, 00260622>. We follow villmow (2019) to preprocess the data and map each entity with the text with a real meaning.

The above triple can then be processed with the real semantic meaning: <land reform, hypernym, reform>. The word2vec word embedding is pretrained from a google news corpus. We train the WordNet Embedding in the following way:

1. We split our dataset to use 90% for training, 10% for testing.

2. Triples of $(head, relation, tail)$ are encoded as relational triples (h, r, t).

3. We randomly initialize embeddings for each h_i, r_j, t_k, use the scoring function in Equation 4 and minimize the loss by Margin Loss.

4. We sampled 20 corrupted entities. Learning rate is set at 0.05 and training epoch at 300.

Results can be found in the Table 7, which shows that our WordNet embeddings do contain semantic information.

G List of words of experiments: Decomposing word2vec Embedding by Additive Compositionality Detection

allegorically, whimsicality, whimsically, voyeuristically, weightier, weightiest, weightily, weightiness, weightlessly, veritably, visualizations, tyrannically, traitorously, transcendentally, transitionally, tangentially, temperamentally, surgically, structurally, studiously, studiousness, stylistically, spiritualistic, slipperiest, slipperiness, serviceability, serviceably, sectionalism, sentimentalism, sentimental-

Table 5: Additive Compositionality Metrics Across SBERT Layers

Layer	Mean Sim (Real)	Mean Sim (Permuted)	Norm. Cosine Sim	Hits@5 Acc (Real)	Hits@5 Acc (Permuted)	Norm. Retrieval Acc
0	0.8889	0.7808	0.4930	0.59	0.0397	0.5731
1	0.9366	0.8671	0.5228	0.57	0.0406	0.5518
2	0.9397	0.8576	0.5767	0.61	0.0390	0.5942
3	0.9403	0.8330	0.6424	0.64	0.0412	0.6245
4	0.9408	0.8298	0.6523	0.66	0.0400	0.6458
5	0.9409	0.8273	0.6577	0.62	0.0417	0.6035
6	0.7761	0.4865	0.5640	0.59	0.0405	0.5727

Table 6: Additive Compositionality Metrics at Different Training Steps of BERT

Model	Training Steps	Mean Sim (Real)	Mean Sim (Permuted)	Norm. Cosine Sim	Hits@5 Acc (Real)	Permuted Acc	Norm. Retrieval Acc
cls_0k	0	0.9884	0.9882	0.0163	0.44	0.0418	0.4156
cls_20k	20,000	0.8787	0.7722	0.4676	0.55	0.0407	0.5309
cls_40k	40,000	0.8773	0.7724	0.4607	0.48	0.0405	0.4581
cls_100k	100,000	0.9201	0.8323	0.5236	0.55	0.0417	0.5304
cls_1000k	1,000,000	0.9545	0.9149	0.4655	0.44	0.0408	0.4162
cls_2000k	2,000,000	0.9538	0.9094	0.4896	0.48	0.0415	0.4575

Table 7: Link prediction performance for WordNet

	Hits@1	Hits@3	Hits@10	MRR
WordNet	0.39	0.41	0.43	0.40

ist, sentimentalized, sentimentalizes, sentimentalizing, sentimentally, serialization, serializations, satanically, reverentially, ritualistically, regularization, quizzically, rapturously, puritanically, probationers, probationer, psychiatrically, preferentially, practicably, practicalities, pleasurably, polarizations, phenomenally, personalizations, pessimistically, pathetically, occupationally, optionally, oratorically, nationalizations, nautically, neutralizer, neutralizers, mysteriously, mysteriousness, narcissistically, moralistically, melodiously, melodiousness, memorializes, memorializing, metrication, materialistically, mechanistically, mechanizations, longitudinally, lexically, liberalization, liquidator, liquidators, journalistic, inferentially, injuriously, hysterically, idealistically, heretically, futuristically, fractionally, fluoridation, fictionalized, fictionalizes, fictionalizing, figuratively, farcically, fatalistic, environmentalists, environmentally, episodically, equitably, emotionlessly, ecclesiastically, editorialized, editorializes, editorializing, editorially, educationalist, educationalists, educationally, egotistical, egotistically, dictatorships, differentiations, derivatively, developmentally, deviationist, deviationists, decoratively, deferentially, definitively, demagogically, demonically, decimalization, cumulatively, conversationalists, conversationally, confidentialities, conspiratorially, collectivization, colonialists, classically, chauvinistically, censoriousness, certifications, capitalizations, breathalyser, breathalysers, brutalization, antagonistically, apocalyptically, weightlessness, westernization, victoriously, visualization, vocalization, urbanization, Unitarian, Unitarians, transcendentalism, transcendentalist, transcendentalists, theatrically, theoretically, technicalities, technicality, technically, speculatively, socialistic, socialization, sophisticate, sophisticated, sophisticates, significantly, sensational, sentimentalists, sentimentality, sentimentalize, scientifically, satirically, rotationally, residentially, relativistic, realistically, prudentially, pressurization, probabilistic, probationary, professionalism, professionally, popularization, potentialities, potentiality, potentially, practicability, practicality, practically, polarization, phosphorescence, phosphorescent, physically, physicalness, periodically, personalization, particularistic, paternalistic, operational, oratorical, organizational, normalization, numerically, negatively, negativism, neutralization, nationalistic, nationalization, naturalistic, naturalization, mechanically, memorialize, memorialized, metrically, maturational, localization, linguistically, liquidation, liquidations, juridical, justifiably, industrialization, imperialistic, incidentally, identically, imaginatively, historically, harmoniously, graphically, generalization, generalizations, fictionalize, existentialism, existentialist, existentialists, evangelicalism, environmentalism, environmentalist, equalization, equalizers, equatorial, electrically, electronically, emotionalism, emotionality, emotionally, economically, editorialist, editorialize, directionality, directionally, dictatorial, dictatorship, differentiation, conversationalist, confidentiality, confidentially, colonialism, colonialist, commercialization, communicational, civilizational, centralization, certification, chemically, capitalistic, capitalization, catastrophically, categorically, behaviorally, authentication, authentications, authenticator, artistically, architecturally, anatomically, Anglicanism, alternatively, altruistically, adventurously, acoustically, activation, additionally

H Movie-Lens Training Details

This experiment was conducted on the MovieLens 1M dataset (Harper and Konstan, 2015) which consists of a large set of movies and users, and a set of movie ratings for each individual user. It is widely used to create and test recommender systems. Typically, the goal of a recommender system is to predict the rating of an unrated movie for a given user, based on the rest of the data. The dataset contains 6040 users and approximately 3900 movies. Each user-movie rating can take values in 1 to 5. There are 1 million triples (out of a possible $6040 \times 3900 = 23.6m$), so that the vast majority of user-movie pairs are not rated.

Users and movies each have additional attributes attached. For example, users have demographic information such as gender, age, or occupation. Whilst this information is typically used to improve the accuracy of recommendations, we use it to test whether the embedding of a user correlates to private attributes, such as gender or age. We compute our graph embedding based only on ratings, leaving user attributes out. Experiments for training knowledge graph embeddings are implemented with the OpenKE (Han et al., 2018) toolkit. We train our model on GeForce GTX TITAN X.

We embed the knowledge graph in the following way:

1. We split our dataset to use 90% for training, 10% for testing.

2. Triples of $(user, rating, movie)$ are encoded as relational triples (h, r, t).

3. We randomly initialize embeddings for each h_i, r_j, t_k and train embeddings to minimize the loss in equation 6 [7].

4. We sampled 10 corrupted entities and 4 corrupted relations per true triple. Learning rate is set at 0.01 and training epoch at 300.

We verify the quality of the embeddings by carrying out a link prediction task on the remaining 10% test set. We achieved a RMSE score of 0.88, Hits@1 score of 0.46 and Hits@3 as 0.92, MRR as 0.68 and MR as 1.89.

I Analysis of the Canonical Component of CCA for Knowledge Graph Embedding

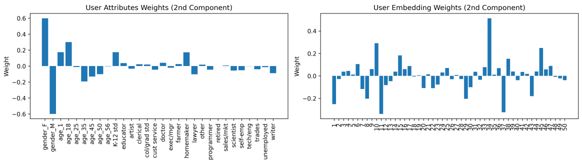


Figure 7: Positive weights in the second canonical component of CCA show the association between user attributes (e.g., *Female* and *Age: 18*) and user embedding dimensions (e.g., *34* and *10*).

[7] We add a negative sign for the additive scoring function, since we want to maximize the probability of the true triple, which aligns the setting of this loss function