Jerrold J. Katz and Paul M. Postal. 1963. Semantic interpretation of idioms and sentences containing them. *Quarterly Progress Report*, 70:275–282.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2025. Pmet: precise model editing in a transformer. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

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.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. Decompx: Explaining transformers decisions by propagating token decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2649–2664.

Richard Montague. 1970a. English as a formal language. In *Linguaggi nella società e nella tecnica*, pages 189–223. Edizioni di Comunità.

Richard Montague. 1970b. Universal grammar. *Theoria*, 36(3):373–398.

Richard Montague and Richmond H Thomason. 1975. Formal philosophy. selected papers of richard montague. *Erkenntnis*, 9(2).

Nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong.

Barbara H. Partee. 1984. Compositionality. In Fred Landman and Frank Veltman, editors, *Varieties of Formal Semantics*, pages 281–312. Foris Publications.

Jackson Petty, Sjoerd Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. 2024. The Impact of Depth on Compositional Generalization in Transformer Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7239–7252, Mexico City, Mexico. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Sean Tull, Robin Lorenz, Stephen Clark, Ilyas Khan, and Bob Coecke. 2024. Towards compositional interpretability for xai. *arXiv preprint arXiv:2406.17583*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Haoyu Wang, Guozheng Ma, Cong Yu, Ning Gui, Linrui Zhang, Zhiqi Huang, Suwei Ma, Yongzhe Chang, Sen Zhang, Li Shen, et al. 2023. Are large language models really robust to word-level perturbations? In *Socially Responsible Language Modelling Research Workshop*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yongjing Yin, Lian Fu, Yafu Li, and Yue Zhang. 2024. On compositional generalization of transformer-based neural machine translation. *Information Fusion*, 111:102491.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907.

## A Compositionality and Localisation

The concept of linguistic compositionality has evolved from its origins in Frege's work (Frege, 1892), which started conceptualising the notion that the meaning of a complex expression is determined by its constituent parts and their syntactic arrangement. This principle was formalised by Montague (Montague, 1970b,a), who applied mathematical rigour to natural language semantics, thereby reinforcing the compositional approach within formal semantics. Linguistic phenomena such as idioms, context-dependence, and metaphor, which seemed to violate compositionality, prompted debates on its universality (Katz and Postal, 1963; Jackendoff, 1997), with theoretical accounts evolving to integrate these phenomena, leading to a more nuanced understanding that balances strict compositional rules with allowances for non-compositional elements (Partee, 1984).

While the syntactic-logical connection entailed by formal models is not assumed to be induced by neural language models, there is a common assumption that those models should entail a syntactic compositionality function, which allows for a systematic model for meaning composition, i.e., that the syntactic structure of a complex expression $s$ is significantly determined by the syntactic properties of its constituent parts and the rules used to combine them. Formally, for any sentence $s$, its syntactic properties can be defined as a function $f$ of the syntactic properties of its immediate constituents $s_1, s_2, \ldots, s_n$ and the syntactic operations applied:

$$\text{Syntax}(s) = f\left(\text{Syntax}(s_1), \text{Syntax}(s_2), \ldots, \text{Syntax}(s_n), \text{Rules}\right) \tag{7}$$

Within the context of distributed representations, a meaning representation can be factored into its syntactic and content (term embedding) components. A compositional distributional semantic model merges syntactic compositionality with distributional semantics by representing token meanings as vectors (token embeddings) in a continuous semantic space and combining them according to syntactic structure. Formally, each token $t$ is associated with a vector $\mathbf{v}_t \in \mathbb{R}^n$ that captures its semantic content based on distributional information.

For a complex syntactic expression $s$ composed of constituents $s_1, s_2, \ldots, s_n$, the semantic representation $\mathbf{v}_s$ is computed using a compositional function $f$ that integrates both the vectors of the constituents and the syntactic operations applied:

$$\mathbf{v}_s = f\left(\mathbf{v}_{s_1}, \mathbf{v}_{s_2}, \ldots, \mathbf{v}_{s_n}, \text{Syntactic structure}\right) \tag{8}$$

This function $f$ is designed to reflect syntactic compositionality by structurally combining the embeddings of the constituents according to the syntactic rules governing their combination.

In the context of a specific transformer-based LM model implementing an interpretation function of an input s, the question which is central to this work is whether the contiguous composition of tokens is reflected within the structure of the transformer-based LMs and its constituent parts, layers $l_0 \ldots l_n$, multi-head attention, feedforward layers and residual connections, i.e. whether the representations $\mathbf{h}_i^{(k)}$ at each layer $l_k$ explicitly encode the composition of contiguous tokens $t_i, t_{i+1}$, and how the model's components contribute to this encoding.

## B Elaborations on Experimental Setup

### B.1 Downstream Task Definitions

The tasks selected for this study are designed to evaluate the effects of compositional aggregation, focusing on tasks that are strictly dependent on input tokens and their compositional semantics while minimising variability. Each task produces a single-token output, and predictions are considered correct if they exactly match the target token. The following are the formal definitions for each task.

**Inverse Definition Modelling (IDM):** The *IDM* task involves predicting a term $T$ based on a given natural language definition $D$. Let $D = \{d_1, d_2, \ldots, d_n\}$ represent the sequence of tokens constituting the definition. The goal is to generate the corresponding term $T$, where:

$$T = \arg\max_{t \in \mathcal{V}} P(t \mid D) \tag{9}$$

Here, $\mathcal{V}$ is the vocabulary of possible terms, and $t$ is a candidate term. A prediction is correct if the term $T$ exactly matches the target term. The task prompt used for IDM was structured as follows:

```
"<definition> is called a"
```

For example, given the definition "A domesticated carnivorous mammal that typically has a long snout, an acute sense of smell, non-retractile claws, and a barking or howling voice," the task would require the model to predict the term "dog."

**Synonym Prediction (SP):** The *SP* task requires the model to generate a synonym $S$ for a given word $W$. Let $W \in \mathcal{V}$ represent the input word. The task is to predict a synonym $S$, such that:

$$S = \arg\max_{s \in \mathcal{V}} P(s \mid W) \tag{10}$$

where $s$ is a candidate synonym from the vocabulary $\mathcal{V}$. The prediction is considered correct if $S$ exactly matches the target synonym. The task prompt used for SP was structured as follows:

```
"<word> is a synonym of"
```

For instance, given the input word "happy," the task would ask the model to predict the synonym "joyful."

**Hypernym Prediction (HP):** The *HP* task involves predicting a more general term, or hypernym, $H$ for a given word $W$. Let $W \in \mathcal{V}$ represent the input word. The objective is to predict a hypernym $H$, such that:

$$H = \arg\max_{h \in \mathcal{V}} P(h \mid W) \tag{11}$$

where $h$ is a candidate hypernym. The prediction is correct if $H$ exactly matches the intended hypernym. The task prompt used for HP was structured as follows:

```
"<word> is a type of"
```

For example, given the word "cat," the task would ask the model to predict the hypernym "animal."

These tasks focus on generating precise, single-token predictions, allowing for a rigorous evaluation of the model's ability to capture and process compositional semantics.

### B.2 Dataset Descriptions and Preprocessing

The training and test datasets are constructed by extracting definitions, hypernyms, and synonyms for each synset from WordNet (Fellbaum, 1998), whose usage is unencumbered by licensing restrictions. WordNet is a lexical database of the English language, containing over 117,000 synsets of nouns, verbs, adjectives, and adverbs. Each synset represents a unique concept and is annotated with part of speech, definition, hypernyms, synonyms, and other semantic relationships. It is focused on general-purpose vocabulary and does not target specific demographic groups or domains. Definitions were cleaned using typical preprocessing techniques, such as removing special characters, punctuation, and extra spaces, and removing parenthesised content when necessary. The dataset was initially split 80-20, with 20% used for training. The remaining 80% was then split 90-10, with 10% for validation and 90% for testing. The test dataset was filtered to retain only single-token predictions matching each model's tokenisation. Table 2 shows the test dataset sizes used for each task and model, including inverse dictionary modelling (IDM), synonym prediction (SP), and hypernym prediction (HP).

| Model | Task | Original Test Set | Fine-tuned Test Set |
|---|---|---|---|
| GPT2 (S,M,L) | IDM | 11,948 | 8,651 |
| | SP | 7,753 | 5,578 |
| | HP | 25,364 | 18,273 |
| Gemma-2B | IDM | 24,831 | 17,859 |
| | SP | 16,014 | 11,533 |
| | HP | 44,687 | 32,209 |
| Llama3 (3B, 8B | IDM | 14,991 | 10,828 |
| | SP | 9,360 | 6,723 |
| | HP | 31,962 | 23,070 |
| Qwen2.5 (0.5B, 1.5B, 3B) | IDM | 14,927 | 10,780 |
| | SP | 9,195 | 6,598 |
| | HP | 31,845 | 23,000 |

Table 2: Test set sizes for each model and task (IDM: Inverse Dictionary Modelling, SP: Synonym Prediction, HP: Hypernym Prediction) derived from WordNet.

| Model | Params | Layers | $D_{model}$ | Heads | Act. | MLP Dim |
|---|---|---|---|---|---|---|
| GPT2-small | 124M | 12 | 768 | 12 | GELU | 3072 |
| GPT2-medium | 302M | 24 | 1024 | 16 | GELU | 4096 |
| GPT2-large | 708M | 36 | 1280 | 20 | GELU | 5120 |
| Gemma-2B | 2B | 32 | 4096 | 16 | GELU | 8192 |
| LLama3-3B | 3.2B | 28 | 3072 | 24 | SiLU | 8192 |
| LLama3-8B | 7.8B | 32 | 4096 | 32 | SiLU | 14336 |
| Qwen2.5-0.5B | 391M | 24 | 896 | 14 | SiLU | 4864 |
| Qwen2.5-1.5B | 1.4B | 28 | 1536 | 12 | SiLU | 8960 |
| Qwen2.5-3B | 3.0B | 36 | 2048 | 16 | SiLU | 11008 |

Table 3: Model properties across architectures. Params: number of parameters, Layers: number of layers, $D_{model}$: size of word embeddings and hidden states, Heads: number of attention heads, Act.: Activation function, MLP Dim: dimensionality of the FF layers.

### B.3 Model Specifications and Fine-tuning Parameters

Table 3 provides a comparative overview of various Transformer models used in this study. We used GPT2 models (released under the Modified MIT License), Gemma-2B (released under the Gemma Terms of Use), Llama3 models (released under the Meta Llama 3 Community License), and Qwen models (released under Apache License 2.0). The used models were mainly pre-trained on English