

Investigating Semantic Similarity’s Role in Word Analogies

Anonymous ACL submission

Abstract

Inspired by the parallelogram model for word analogy, given proportional word analogies such as King: Queen:: Man: Woman, their offset similarity is viewed as relational similarity between the words $\text{king} - \text{queen} = \text{man} - \text{woman} = r$. To that end, despite numerous studies, the role of semantic similarity in offset or relational similarity remains unclear. As such, various works framing the problem as that of analogy completion $x : x^* :: y : ?$, show that often finding the answer y^* , by choosing the word most similar to y , gives better results than finding it with the offset vector $y^* = y + (x - x^*)$. In this paper, we view word analogy as that of relational mapping. Accordingly, given a query $x - x^* = ?$, the task is to find word pairs that are related to one another as words in the query are related. Accordingly, noting that word vectors are obtained through word co-occurrence, and distinguishing between similarity due to relatedness (such as similarity between coffee and cup) and semantic similarity (such as similarity between coffee and tea), we show that in common word embedding models, word vectors exhibit similarities with both word categories mentioned. Additionally, in multiple choice word analogy questions; we show how word similarity can be the reason why the offset vectors fail in recognizing relational similarity.

1 Introduction

Analogy and analogical reasoning are a crucial aspect of human cognition. As such, understanding them is of significant importance. One common example of analogy is the so-called proportional word analogies, traditionally used to evaluate word embedding models. To that end, given that word embedding models such as fasttext and word2vec obtain the representation of a given word using words that accompany it in textual contexts, words can be similar in two main possible ways: Firstly,

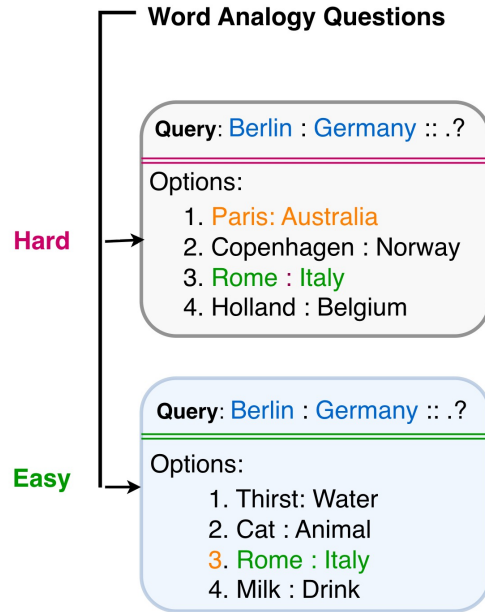


Figure 1: Illustrating difficulty factor of answering word analogy questions with semantic similarity.

two given words that often appear together in textual contexts such as coffee and cup, may have similar embeddings. Assuming words appearing together in textual contexts are related, we can call such similarity in word vectors as similarity due to relatedness. It is to note that, in literature, such similarity is also called topical similarity, or domain similarity (Hill et al., 2015). Secondly, words that do not appear in the same contexts can still have similarity in their vectors, given that there are words that often accompany both words. Arguing that the common words in different contexts represent relation similarity involving our given words, we can call such similarity, as similarity due to relation similarity. Considering the set theoretic view of similarity, where similarity is described as feature-mapping process (Tversky, 1977), our view of semantic similarity being due to relation similarity is coherent with literature. For example, consider proportional analogy word:language :: note

: music; one can imagine there are words that express the relations between words (word, language), similar to the relations between words (note, music) in various contexts. Noting the one-to-one correspondence between head (word, language) and tail (note, music) words, in this work we also call such similarity as pair wise similarity between head and tail words. As alternative intuition such similarity in similarly related words is that relations have directions, therefore we expect "word" to be similar to "note", while "language" should be similar to "music". Semantic similarity from the Google word analogy dataset is illustrated in Figure 2.

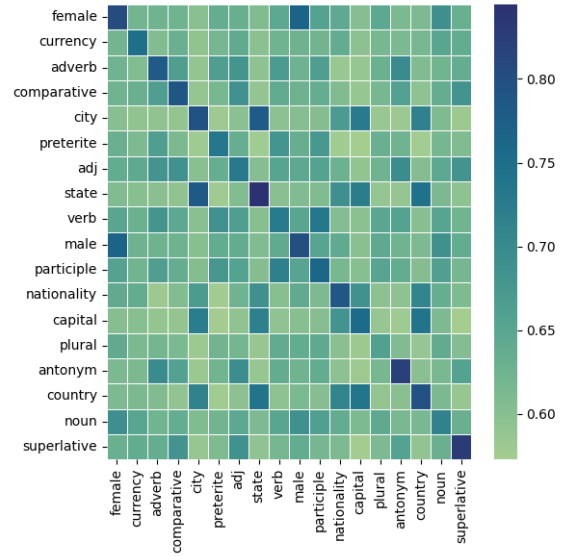


Figure 2: Similarity of relationally similar words using the word analogy dataset Google, wherein each head and tail word for a given relation r corresponds to a unique concept (for the relation "citizen of", the head word is a person, and the tail word is nationality).

Word embedding models are shown to capture word relations without being explicitly trained on them. As such, inspired by the parallelogram model for analogy, the similarity between offset of word pairs is viewed as the relation similarity between word pairs $\text{king} - \text{queen} = \text{man} - \text{woman} = \mathbf{r}$, with $\mathbf{r}$ being referred to as the offset vector. To that end, the similarity between offsets of such words, and the similarity between words themselves have been a topic of research. Accordingly, given proportional word analogies $x : x^* :: y : y^*$, studies framing the problem as multi-choice analogy completion problem $x : x^* :: y : ?$, show that for offset vector to be helpful in answering the question, y and y^* should be very close to each other (Similar). Additionally, studies also show that similarity alone is sometimes more successful at the analogy completion problem. In this paper, we frame the problem as that of relational mapping (Gentner and Forbus, 2011) $x : x^* :: ?$. This simple change of problem representation makes the word analogy problem consistent with the structural mapping definition of analogy (Gentner, 1983), where objects, as well as relations are mapped. Accordingly, we show that for offset of analogous words to be similar, important is semantic similarity between head (x, y) and tail (x^*, y^*) terms. Moreover, the linear connection between pairwise head and tail similarity and that of offset similarity is also consistent with the notion of rerepresentation from cognitive science. As such, according to rerepresentation, during analogical reasoning humans change the representation of entities or objects so that the relation matches (Gentner and Forbus, 2011).

2 Related Works

The related work of this paper concerns 2 categories of research papers: those that investigate the emergence of linear analogies in word embedding models, and those that explore the relation between word similarity and offset similarity.

Vector Arithmetic Based Approaches for Word Analogy: Pennington et al. show that for the given proportional analogies, offset vectors will be similar given that the proportion $p(w|x)/p(w|x^*) = p(w|y)/p(w|y^*)$ holds for all words w in the vocab. Furthermore, (Ethayarajh et al., 2019) proves the above conjecture theoretically. Moreover, using a word co-occurrence matrix, (Korchinski et al., 2025; Karkada et al., 2025) show how simple linear analogies in the word embedding space emerge.

Word Similarity and Offset Similarity in Literature: Various works study such a connection in the context of analogy completion in a multiple-choice question-answering format, such as $\text{King} : \text{Queen} :: \text{Man} : ?$, where it is commonly solved by finding the word closest to $\text{man} + (\text{queen} - \text{king})$. As such, the parallelogram revisited (Peterson et al., 2020) shows that word similarity alone often gives better results on the task than offset vectors. Additionally, (Linzen, 2016) argue that, given the multiple-choice analogy completion ques-

tion $x : x^* :: y : ?$, when x , x^* and y are added to the distractors, the performance drops significantly. He also shows that, for analogy completion, the change in performance when offsets are not used is not much different from that when offsets are used. (Rogers et al., 2017) further extend the work of (Linzen, 2016) and show that as the similarity between y and y^* decreases, so does the success of the offset at analogy completion. In other words, (Rogers et al., 2017; Linzen, 2016) point out that in order for the offset vector to work, y should be very similar to y^* , and the help due to the use of the offset vector is insignificant, whereas in this work we show that y and y^* may be similar as they are related and appear often in same contexts; however, y can still be similar to word x_i , such that there are some forms of relation similarity such as $x_i : x_i^* :: y : y_j^*$, for some i and j , and the offset vector may to an extent exhibit such relation similarities. Important to note is that, when it comes to comparison as is need in a multi-choice word analogy questions, offset vector might point to most similar words. To that end, both similarity types seem to influence the offset vectors.

3 Common Word Analogy Benchmarks

(Ushio et al., 2021) categorizes word analogy datasets into psychometric analogy tests and lexical semantics benchmarks. For lexical semantics benchmarks, there are Google (Mikolov et al., 2013) and BATS (Gladkova et al., 2016). Google contains only semantic and morphological relations, whereas BATS includes a more diverse set of analogies categorized into lexicographic, encyclopedic, derivational, and inflectional morphology (Ushio et al., 2021). From Psychometric Analogy Tests we have SAT (Turney et al., 2005), U4 (Ushio et al., 2021) and U2 (Boteanu and Chernova, 2015). SAT is a college admissions test designed to be challenging for humans. U2 was collected from the English learning website by (Boteanu and Chernova, 2015). U2 is intended for school students of grades 2-12. Similarly, U4 is collected by (Ushio et al., 2021) from the same website and is categorized by difficulty level. Furthermore, EKAR (Chen et al., 2022) is constructed from the Civil Service Exams in China and is quite challenging for language models to solve.

Various datasets used in this paper, and the links to the data, are given in Table 1.

Dataset	choices	(valid/test)
ekar 2022	4	(262/870)
Scan 2022	3-153	178/1616
Google 2021	4	(50/500)
BAT 2021	4	(199/1799)
U4 2021	5/4/3	(48/432)
U2 2021	5/4/3	(24/228)
SAT 2021	5	(48/432)

Table 1: Datasets used in this work.

4 Motivation: Word Similarity and Word Analogy

QUERY: word:language

d_1 : paint:portrait

d_2 : **poetry:rhythm**

d_3 : **note:music** $\leftarrow$ answer

d_4 : **tale:story**

d_5 : **week:year**

Figure 3: A typical word analogy question. Note the query, the answer(d_3) and distractors (d_i , with $i \neq 3$). The types of similarity between query and distractors: semantic similarity (color), and similarity due to relatedness (color).

Noting types of similarity in word vectors, namely: **Similarity Due to Relatedness**, which can be attributed to the frequent appearance of words in the same textual contexts, and **Similarity due to Relation Similarity**, which can be attributed to relation similarity between entities or words expressed in various textual contexts; and the notion of rerepresentation from cognitive science, we argue that studying semantic similarity and relation similarity should involve framing the problem as the of relation mapping rather than analogy completion, as firstly: the analogy completion is influenced by similarity due to relatedness, and secondly, given that word analogies are a form of analogy, relational mapping format of word analogy as was introduced, better match the common accepted understanding of analogies.

Accordingly, given our assumption regarding types of similarity, and our motivation regarding the connection between offset similarity and semantic similarity, we show evidence indicating that in word embedding models, related words might be similar, and that relationally similar words might be similar as a result of semantic similarity (one-to-one similarity between head and tail words from

similarly related words). Additionally, we show that semantic similarity without relational similarity (such as Berlin-Germany!=Paris-Denmark) may fool the vector offset model for word analogy.

With the being said, we propose to study any possible connection between the two, by framing the problem according to accepted definition of analogy; namely relational and structural mapping, where we map not only the entities, but also the relations.

Dataset	Accuracy	$\phi(x, y) < \phi(x, d)$	
		Count(True)	Count(False)
Sat	0.4613	114 (62.9%)	67 (37.0%)
Ekar	0.2912	116 (63.3%)	67 (36.6%)
U2	0.3833	91 (65.0%)	49 (35.0%)
Bats	0.7063	309 (58.5%)	219 (41.4%)
Google	0.9479	18 (69.2%)	8 (30.7%)
U4	0.3828	166 (62.4%)	100 (37.5%)
Total	-	814 (61.5%)	510 (38.5%)

Table 2: An analysis of incorrectly answered questions using the difference(offset) vector across benchmarks. Given the query $x : x^*$, the answer $y : y^*$ and the distractor chosen by offset vector as the answer $d_i : d_i^*$, and the cosine similarity function ψ , and taking $w : w^*$ to be either of concerned distractor or the true answer $\phi(x, w) = \max(\psi(x, w), \psi(x^*, w^*))$. The number in parentheses shows the percentage with respect to the total count of incorrectly answered questions. The last row reports the total counts (with corresponding percentages) across all datasets.

5 The Semantic Similarity and Offset Similarity

5.1 Evidence from Vector Offset Error Analysis

Semantic similarity can be both a limitation, and also an strength of offset vector. For example in the context word analogy question answering, when similarity between query and distractors exist, the offset vector may point to the corresponding distractor as the answer, whereas when such similarity between query and the true answer exist, it helps with identifying the correct answer. Considering the true answer, and the incorrectly predicted answer by offset vector, the similarity between the query and predicted answer compared with the similarity between query and true answer is often in favor the incorrectly chosen distractor by offset vector. Accordingly, as can be seen in Table 2,

similarity between query and the incorrectly selected distractor (as the predicted answer) are often greater than the same between query and the true answer across all the incorrectly answered questions. To that end, using one sided binomial test of H_0 , it can be shown that such similarity between query and incorrectly predicted answer is not a random observation, Appendix Table 6.

5.2 Evidence from Various Permutations of Word Analogy

Based on the parallelogram model for word analogy, given a proportional word analogy $x : x^* :: y : y^*$, we can have further valid analogous permutations. For example, given King : Queen :: Man : Woman, we also have the proportional analogy King : Man :: Queen : Woman. As such, all permutations that result in an equal offset in the parallelogram (Figure 5) are valid proportional analogies, while those that violate offset equality are considered non-analogous and therefore invalid permutations. To that end, (Prade and Richard, 2013) meaningfully links various valid permutations with logic. Here, we use the permutation names as in their work. In addition, we consider two invalid permutations as well, namely $x : y^* :: y : x^*$ and $x : y^* :: x^* : y$ called here perpendicular and perpendicular_ respectively, as the corresponding vectors tend to be perpendicular in the parallelogram Figure 5. Note that the left-hand side in both perpendicular and perpendicular_ corresponds to the same vector, and the right-hand side corresponds to opposite vectors. An overview of various permutations is shown in Figure 5.

Assuming that the identity permutation is about the relation from one context to the relation in another context, such as color: colors:: computer: computers, the central permutation color: computer:: colors:: computers is the permutation wherein similarity due to relatedness and similarity due to semantics align: head words (color, colors) and tail words (computer, computers) each correspond to a unique context, whereas in the identity permutation, head words (color, computer) and tail words (colors, computers) are not from unique contexts. As such, in Figure 4, we can see that often the distribution of offset similarity for the central permutation is placed closer to 1 than the same for the identity permutation. Furthermore, among the invalid permutations perpendicular and perpendicular_, often offset similarity is greater for the perpendicular_ permutation. This in fact shows

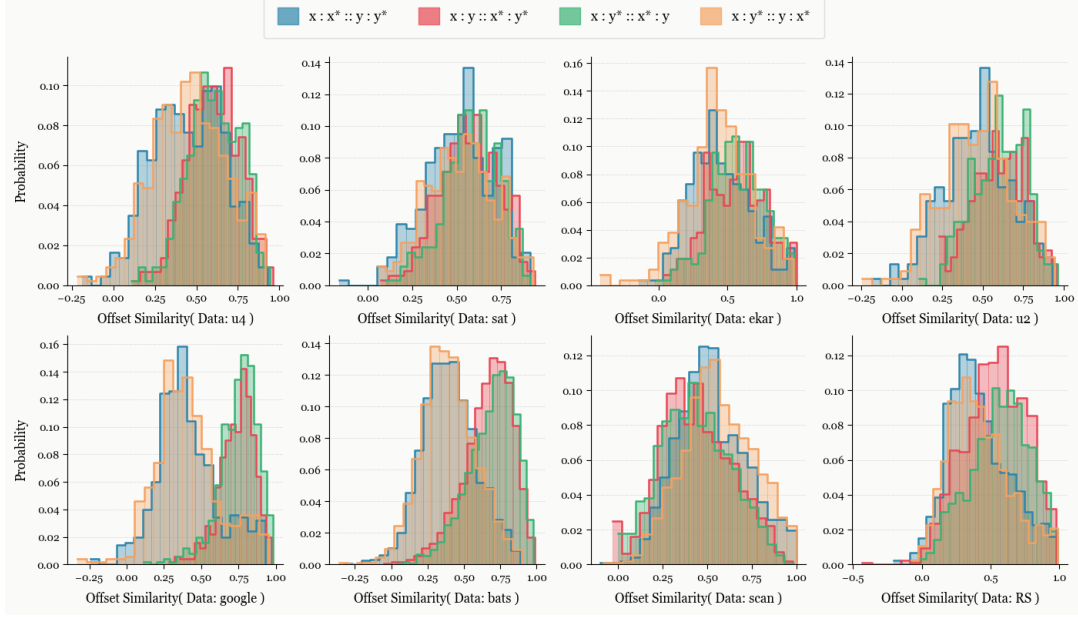


Figure 4: Distribution of offset similarity for various permutations of analogous terms with word embedding model Fasttext. As can be seen, the **central** and **perpendicular** permutations almost have similar distributions, whereas the **identity** and **perpendicular** permutations have similar distributions. Additionally, in some datasets, the central permutation is much closer to 1 than the identity permutation.

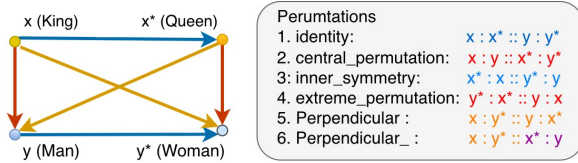


Figure 5: According to the parallelogram model for analogy, the offset vectors of analogous words for various valid permutations are equal. Here the 5th and 6th permutations are invalid, as the corresponding vectors are perpendicular.

that when head and tail are from the same contexts, offset similarity is higher: $(x - y^*, x^* - y)$ both have the head words from the source context, while tail words are from the target context (similar to the central permutation). In the perpendicular permutation, $(x - y^*, y - x^*)$ it is the other way around (see Figure 4). According to our hypothesis stating that while constructing word analogy questions, when choosing distractors, preference is given to words with pairwise similarity with the query, we expect that with the offset vector, the inner_symmetry permutation $(x^* : x :: y^* : y)$ should be easier to solve as the direction of the offset reverses. In other words, if distractors (d_i, d_i^*) have pairwise head and tail similarity with word pair (x, x^*) , then the query in the inner_symmetry permutation is less similar to distractors, as the head word in inner_symmetry is swapped with the tail word (x^*, x) . Accordingly,

to verify if the stated assumptions are indeed correct, according to the different permutations, in each word analogy question we replace the query and answer, and then, in addition to accuracy, we also calculate the average similarity in head and tail words between query, distractors, and the true answer. Additionally, in each case, the offset similarity is calculated as well. The result is shown in Table 3.

As such, Table 3 shows that the inner_sym permutation yields improved accuracy. The reason for this improvement is that distractors are chosen to be pairwise similar to the query under the identity permutation. As such, when head and tail are swapped in inner_symmetry, the direction of the relation changes. Additionally, the central permutation results in much improved performance, as the pairwise similarity between head and tail terms and the so-called relatedness similarity combine. As a takeaway from this experiment, we can state that, in word embedding models, related words are, to some extent similar. Moreover, in word analogy benchmarks, to an extent, distractors are chosen to be semantically similar to the query, however not relationally similar to the query or the true answer. Furthermore, word embedding model may capture the direction of relations through offset vector. To evaluate if the difference in accuracy between any two permutation is significant, we also conducted

Model	Permutation	w	sat			ekar			bats			google		
			avg(Q)	$\Phi(x, w)$	$\psi(r_x, r_w)$	avg(Q)	$\Phi(x, w)$	$\psi(r_x, r_w)$	avg(Q)	$\Phi(x, w)$	$\psi(r_x, r_w)$	avg(Q)	$\Phi(x, w)$	$\psi(r_x, r_w)$
Fasttext	identity	$w = y$	0.46	0.426	0.16938	0.29	0.387	0.05222	0.70	0.408	0.24336	0.94	0.483	0.42609
		$w = d_i$	0.54	0.459	0.08682	0.71	0.478	0.02616	0.30	0.536	0.01245	0.06	0.630	0.00358
	inner_sym	$w = y$	0.77	0.426	0.16938	0.39	0.387	0.05222	0.74	0.408	0.24336	0.96	0.483	0.42609
		$w = d_i$	0.23	0.459	-0.08682	0.61	0.478	-0.02616	0.26	0.536	-0.01245	0.04	0.630	-0.00358
	central	$w = y$	0.80	0.469	0.22976	0.63	0.482	0.18974	0.93	0.602	0.47262	0.99	0.711	0.67655
		$w = d_i$	0.20	0.434	-0.00162	0.37	0.430	0.01640	0.07	0.432	-0.00120	0.01	0.509	0.00083
	extreme	$w = y$	0.76	0.469	0.22976	0.70	0.481	0.18862	0.93	0.602	0.47262	0.99	0.711	0.67655
		$w = d_i$	0.24	0.438	-0.00096	0.30	0.424	-0.02150	0.07	0.443	-0.00007	0.01	0.522	-0.00207
Word2vec	identity	$w = y$	0.41	0.504	0.16668	0.11	0.506	0.06991	0.65	0.533	0.21506	0.70	0.635	0.35761
		$w = d_i$	0.59	0.556	0.08126	0.89	0.632	0.02164	0.35	0.638	0.00798	0.30	0.761	0.02274
	inner_sym	$w = y$	0.73	0.504	0.16668	0.18	0.506	0.06991	0.69	0.533	0.21506	0.73	0.635	0.35761
		$w = d_i$	0.27	0.556	-0.08126	0.82	0.632	-0.02164	0.31	0.638	-0.00798	0.27	0.761	-0.02274
	central	$w = y$	0.75	0.548	0.24704	0.23	0.616	0.27411	0.92	0.678	0.47914	0.80	0.840	0.70733
		$w = d_i$	0.25	0.517	0.01303	0.77	0.570	0.00876	0.08	0.575	-0.00101	0.20	0.675	0.00312
	extreme	$w = y$	0.76	0.548	0.24704	0.26	0.616	0.27411	0.93	0.678	0.47914	0.73	0.840	0.70733
		$w = d_i$	0.24	0.567	-0.00316	0.74	0.568	-0.00556	0.07	0.555	-0.00022	0.27	0.634	-0.00011

Table 3: Accuracy and Similarity across various permutations of word analogies in different benchmarks. ψ is cosine similarity. avg(Q) stands for average questions for which relation similarity of the query is considered similar to w by the corresponding model. Note, given $w=y$, avg(Q) is the total question correctly answer (accuracy). For $w \in [x, y, d_i]$, w stands for word pair (w, w^*) . r_w is the offset corresponding to (w, w^*) . $\Phi(x, w_i) = \frac{1}{n} \sum_{i=1}^n \frac{\psi(x, w_i) + \psi(x^*, w_i^*)}{2}$. For all datasets and models, if the accuracy for inner_sym and central permutations is greater than that of identity permutation, they are highlighted. A statistical significance test is done in the Table 7.

Model	Permutation	sat	ekar	u2	u4
gpt-4.1-mini	identity	0.579	0.538	0.801	0.695
	inner_sym	0.609	0.590	0.808	0.668
	central	0.584	0.779	0.830	0.737
	extreme	0.667	0.764	0.896	0.779
gpt-5.2	identity	0.705	0.766	0.947	0.807
	inner_sym	0.720	0.799	0.950	0.835
	central	0.770	0.951	0.988	0.864
	extreme	0.978	0.954	0.990	0.859

Table 4: Comparative performance of gpt-4.1-mini and gpt5.2 across permutations. An statistical significance test is shown in the Appendix, Table 9.

and a two-proportion Z-test, the result of which is shown in the Appendix, Table 7. According, except permutations: central, and extreme, any other two permutations result in significant difference in model performance.

We conducted a similar experiment with LLMs. To be exact, using ChatGPT API calls, we asked question constructed according to various permutations from models: gpt-5.2 and gpt-4.1-mini as can be seen in the Table 4. To that end, the mentioned models too exhibit improved performance on inner_sym and central permutations that the identity permutations. The statistical test for LLMs, however shown in the Appendix, Table 9, convey slightly different results. Accordingly, for LLMs, the identity and inner_sym do not yield significant

difference in accuracy, however, accuracy significantly differ when comparing identity permutation with central or extreme permutations across benchmarks.

5.3 Evidence From Word Analogy Questions: Google

Using the columnized version of the word analogy dataset Google ¹ (Mikolov et al., 2013), we show that what makes solving word analogy questions with offset vectors difficult is the semantic similarity between the query and distractors. As such, given that the dataset Google covers 3 relation types, namely, 1. capital-common-countries (e.g., Athens, Greece) 2. currency (e.g., USA dollar) 3. gram3-comparative (e.g., big bigger), we first sort word pairs into relation categories, and then from each given relation, sample two pair of words as the query and the answer respectively. To that end, we make two versions of word analogy questions, Easy and Hard. In the Hard version, distractors are chosen such that given the query (x, x^*) , and a distractor d_i, d_j^* , there are word pairs the relation corresponding to the query such that (d_i, d_i^*) , and (d_j, d_j^*) are analogous to the query. This makes the distractors of the Hard variant semantically similar to the query, as shown in Figure

¹<https://huggingface.co/datasets/almogtavor/google-analogy-dataset>

1; however, in the Easy version, distractors are sampled from a relation category other than the query's. As shown in Table 5, across various word embedding models, performance is significantly lower on the Hard version when the query and distractors share semantic similarity, Appendix Table 8. Moreover, in the Table 5, in the so called Easy questions, answering word analogy question using the similarity between head and tail words (x, y), and (x^*, y^*) respectively, can result in better or equivalent performance as offset similarity. However, when distractors (d_i, d_i^*) are chosen to have pairwise head and tail similarity with the query, as in Hard questions), offset similarity achieves relatively better result or performance is close to random.

Category	Model	Easy	Hard	Diff
MLM	BERT-Large*	26.0	17.2	9.3
	BERT-Large	26.4	18.9	7.5
	RoBERTa-Large*	31.9	16.8	15.1
	RoBERTa-Large	46.6	17.2	29.4
Causal	GPT-2*	48.3	38.2	10.1
	GPT-2	59.2	21.4	37.8
Enc-Dec	T5-Large*	32.7	21.0	12.7
	T5-Large	43.6	20.1	23.5
Other	FastText*	88.2	47.2	41.0
	FastText	87.3	21.4	65.8

Table 5: Multi-choice analogy questions (238 questions in each category) answering. For each model, * indicates offset similarity was used; otherwise pairwise similarity between head and tail terms was used. Diff denotes the accuracy gap between Easy and Hard settings; Bold indicates the better accuracy between the offset (*) and pairwise variants for each model, per column. An statistical significance test is shown in the Table 8.

6 Conclusion

In this paper, we show that related words are to an extend similar. Additionally, word embedding model may capture the direction of relation between words as inner_sym permutation result in improved accuracy across benchmarks. Furthermore, the pair wise similarity influences offset similarity between words, and hence can be viewed as limitation for offset vector as it do not always mean relation similarity (as Paris-Belgium should not equal: Berlin- Germany).

Limitations

This paper relies on empirical evidence from state-of-the-art models and word analogy benchmarks. Accordingly, the theoretical background for the observations is not explored. Additionally, the harm due to word similarity for state of the art models other than the two models tested in this work is not clear.

Ethical Considerations

This work is in coherence with ACL code of conduct and ethical considerations.

References

- Adrian Boteanu and Sonia Chernova. 2015. Solving and explaining analogy questions using semantic networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29.
- Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua Xiao, and Hao Zhou. 2022. E-KAR: A benchmark for rationalizing natural language analogical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3941–3955, Dublin, Ireland. Association for Computational Linguistics.
- Tamara Czinczoll, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2022. Scientific and creative analogies in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2094–2100, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.
- 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, Florence, Italy. Association for Computational Linguistics.
- Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.
- Dedre Gentner and Kenneth D. Forbus. 2011. Computational models of analogy. *WIREs Cognitive Science*, 2(3):266–276.
- Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuo. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the*

462	NAACL Student Research Workshop, pages 8–15, San	of transfer learning with a unified text-to-text trans-	517
463	Diego, California. Association for Computational	former. <i>J. Mach. Learn. Res.</i> , 21(1).	518
464	Linguistics.		
465	Felix Hill, Roi Reichart, and Anna Korhonen. 2015.	Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017.	519
466	SimLex-999: Evaluating semantic models with (genu-	The (too many) problems of analogical reasoning	520
467	ine) similarity estimation. <i>Computational Linguis-</i>	with word vectors. In <i>Proceedings of the 6th Joint</i>	521
468	<i>tics</i> , 41(4):665–695.	<i>Conference on Lexical and Computational Semantics</i>	522
469	Dhruva Karkada, James B. Simon, Yasaman Bahri, and	(*SEM 2017), pages 135–148, Vancouver, Canada.	523
470	Michael R. DeWeese. 2025. Closed-form training dy-	Association for Computational Linguistics.	524
471	namics reveal learned features and linear structure in		
472	word2vec-like models. <i>Preprint</i> , arXiv:2502.09863.	Peter D. Turney, Michael L. Littman, Jeffrey Bigham,	525
473		and Victor Shnayder. 2005. Combining indepen-	526
474	Daniel J. Korchinski, Dhruva Karkada, Yasaman Bahri,	dent modules in lexical multiple-choice problems.	527
475	and Matthieu Wyart. 2025. On the emergence of	<i>Preprint</i> , arXiv:cs/0501018.	528
476	linear analogies in word embeddings. <i>Preprint</i> ,		
477	arXiv:2505.18651.	Amos Tversky. 1977. Features of similarity. <i>Psycholog-</i>	529
478	Tal Linzen. 2016. Issues in evaluating semantic spaces	<i>ical Review</i> , 84(4):327–352.	530
479	using word analogies. In <i>Proceedings of the 1st Work-</i>		
480	<i>shop on Evaluating Vector-Space Representations for</i>	Asahi Ushio, Luis Espinosa Anke, Steven Schockaert,	531
481	<i>NLP</i> , pages 13–18, Berlin, Germany. Association for	and Jose Camacho-Collados. 2021. BERT is to NLP	532
482	Computational Linguistics.	what AlexNet is to CV: Can pre-trained language	533
483	Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Man-	models identify analogies? In <i>Proceedings of the</i>	534
484	dar Joshi, Danqi Chen, Omer Levy, Mike Lewis,	<i>59th Annual Meeting of the Association for Compu-</i>	535
485	Luke Zettlemoyer, and Veselin Stoyanov. 2019.	<i>tational Linguistics and the 11th International Joint</i>	536
486	Roberta: A robustly optimized BERT pretraining	<i>Conference on Natural Language Processing (Vol-</i>	537
487	approach. <i>CoRR</i> , abs/1907.11692.	<i>ume 1: Long Papers</i>), pages 3609–3624, Online. As-	538
488		sociation for Computational Linguistics.	539
489	Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig.		
490	2013. Linguistic regularities in continuous space		
491	word representations. In <i>Proceedings of the 2013</i>		
492	<i>Conference of the North American Chapter of the</i>		
493	<i>Association for Computational Linguistics: Human</i>		
494	<i>Language Technologies</i> , pages 746–751, Atlanta,		
495	Georgia. Association for Computational Linguistics.		
496	Jeffrey Pennington, Richard Socher, and Christopher		
497	Manning. 2014. GloVe: Global vectors for word		
498	representation. In <i>Proceedings of the 2014 Confer-</i>		
499	<i>ence on Empirical Methods in Natural Language Pro-</i>		
500	<i>cessing (EMNLP)</i> , pages 1532–1543, Doha, Qatar.		
501	Association for Computational Linguistics.		
502	Joshua C. Peterson, Dawn Chen, and Thomas L. Grif-		
503	fiths. 2020. Parallelograms revisited: Exploring the		
504	limitations of vector space models for simple analo-		
505	gies. <i>Cognition</i> , 205:104440.		
506	Henri Prade and Gilles Richard. 2013. From analogical		
507	proportion to logical proportions. <i>Logica Universalis</i> ,		
508	7(4):441–505.		
509	Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu,		
510	Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming		
511	Zhou. 2020. ProphetNet: Predicting future n-gram		
512	for sequence-to-SequencePre-training. In <i>Findings</i>		
513	<i>of the Association for Computational Linguistics:</i>		
514	<i>EMNLP 2020</i> , pages 2401–2410, Online. Association		
515	for Computational Linguistics.		
516	Colin Raffel, Noam Shazeer, Adam Roberts, Katherine		
	Lee, Sharan Narang, Michael Matena, Yanqi Zhou,		
	Wei Li, and Peter J. Liu. 2020. Exploring the limits		

.1 Reproducibility

https://github.com/bahramiramazan/offset_and_semantic

A Word Embedding Models Considered

Various word embedding models considered in this study are 1. bert-large-uncased (Devlin et al., 2018) 2. roberta-large (Liu et al., 2019) 3. gpt2 hugging-face 4. prophetnet (Qi et al., 2020) 5. opt hugging-face 6. T5-large (Raffel et al., 2020) 7. word2vec (Pennington et al., 2014)

B Experimental Setup

For GPT models considered in this experiments, *gpt* – 4.1 – *mini* and *gpt gpt* – 5.2, using the api, they were prompted with the question x is x^* as : 1. y is to y^* 2. d_i is to d_j^* 3. ..etc. The GPT models were accessed during 01.02.2026 to 15.07.2026.

B.1 Significance Test

For various observations in this paper, we also conduct suitable statistical tests to determine the significance of our observations.

Dataset	n	Count(True)	$\hat{p}$	z	p -value
Sat	181	114	0.630	3.50	2×10^{-4}
Ekar	183	116	0.634	3.62	1×10^{-4}
U2	140	91	0.650	3.55	2×10^{-4}
Bats	528	309	0.585	3.91	5×10^{-5}
Google	26	18	0.692	1.96	$0.038^\dagger$
U4	266	166	0.624	3.99	3×10^{-5}
Total	1324	814	0.615	8.36	$< 10^{-15}$

Table 6: One-sided binomial test of $H_0 : p = 0.5$ against $H_1 : p > 0.5$, where p is the probability that $\phi(x, y) < \phi(x, d)$ holds among incorrectly answered questions. $\hat{p}$ is the observed proportion of True cases, and z is the normal-approximation statistic. † Exact binomial p -value, used because n is small. All datasets are individually significant at $\alpha = 0.05$; all except Google remain significant after Bonferroni correction ($\alpha/6 \approx 0.0083$).

Model	Dataset	Pair Comparison	Accuracies (p_1 vs p_2)	Z-score	p-value
Fasttext	SAT ($N = 432$)	identity vs. inner_sym	0.46 vs 0.77	-9.18	< 0.0001*
		identity vs. central	0.46 vs 0.80	-10.22	< 0.0001*
		identity vs. extreme	0.46 vs 0.76	-8.84	< 0.0001*
		central vs. extreme	0.80 vs 0.76	1.41	0.1585
	EKAR ($N = 870$)	identity vs. inner_sym	0.29 vs 0.39	-4.43	< 0.0001*
		identity vs. central	0.29 vs 0.63	-14.24	< 0.0001*
		identity vs. extreme	0.29 vs 0.70	-17.04	< 0.0001*
		central vs. extreme	0.63 vs 0.70	-3.12	0.0018*
	BATS ($N = 1799$)	identity vs. inner_sym	0.70 vs 0.74	-2.71	0.0067*
		identity vs. central	0.70 vs 0.93	-17.06	< 0.0001*
		identity vs. extreme	0.70 vs 0.93	-17.06	< 0.0001*
		central vs. extreme	0.93 vs 0.93	0.00	1.0000
	Google ($N = 500$)	identity vs. inner_sym	0.94 vs 0.96	-1.48	0.1389
		identity vs. central	0.94 vs 0.99	-3.88	0.0001*
		identity vs. extreme	0.94 vs 0.99	-3.88	0.0001*
		central vs. extreme	0.99 vs 0.99	0.00	1.0000

* Statistically significant at $\alpha = 0.05$.

Table 7: Two-Proportion Z-Test comparing permutation performance (significant results marked with *).

Category	Model	Accuracies (Easy vs. Hard)	Z-score	p-value	Significant ($\alpha = 0.05$)
MLM	BERT-Large*	26.0% vs. 17.2%	2.33	0.0199	Yes*
	BERT-Large	26.4% vs. 18.9%	1.95	0.0511	No
	RoBERTa-Large*	31.9% vs. 16.8%	3.83	0.0001	Yes*
	RoBERTa-Large	46.6% vs. 17.2%	6.87	< 0.0001	Yes*
Causal	GPT-2*	48.3% vs. 38.2%	2.22	0.0265	Yes*
	GPT-2	59.2% vs. 21.4%	8.39	< 0.0001	Yes*
Enc-Dec	T5-Large*	32.7% vs. 21.0%	2.87	0.0041	Yes*
	T5-Large	43.6% vs. 20.1%	5.49	< 0.0001	Yes*
Other	FastText*	88.2% vs. 47.2%	9.54	< 0.0001	Yes*
	FastText	87.3% vs. 21.4%	14.40	< 0.0001	Yes*

* Statistically significant at $\alpha = 0.05$.

Table 8: Two-Proportion Z-Test comparing Easy vs. Hard performance for each model ($N = 237$).

Model	Dataset	Pair Comparison	Accuracies (p_1 vs p_2)	Z-score	p-value
gpt-4.1-mini	SAT ($N = 432$)	identity vs. inner_sym	0.579 vs 0.609	-0.90	0.3693
		identity vs. central	0.579 vs 0.584	-0.15	0.8816
		identity vs. extreme	0.579 vs 0.667	-2.67	0.0076*
		central vs. extreme	0.584 vs 0.667	-2.52	0.0117*
	EKAR ($N = 870$)	identity vs. inner_sym	0.538 vs 0.590	-2.19	0.0287*
		identity vs. central	0.538 vs 0.779	-10.60	< 0.0001*
		identity vs. extreme	0.538 vs 0.764	-9.89	< 0.0001*
		central vs. extreme	0.779 vs 0.764	0.75	0.4562
	U2 ($N = 228$)	identity vs. inner_sym	0.801 vs 0.808	-0.19	0.8505
		identity vs. central	0.801 vs 0.830	-0.80	0.4247
		identity vs. extreme	0.801 vs 0.896	-2.83	0.0047*
		central vs. extreme	0.830 vs 0.896	-2.05	0.0404*
	U4 ($N = 432$)	identity vs. inner_sym	0.695 vs 0.668	0.85	0.3944
		identity vs. central	0.695 vs 0.737	-1.37	0.1710
		identity vs. extreme	0.695 vs 0.779	-2.80	0.0050*
		central vs. extreme	0.737 vs 0.779	-1.44	0.1495
gpt-5.2	SAT ($N = 432$)	identity vs. inner_sym	0.705 vs 0.720	-0.49	0.6262
		identity vs. central	0.705 vs 0.770	-2.17	0.0299*
		identity vs. extreme	0.705 vs 0.978	-10.99	< 0.0001*
		central vs. extreme	0.770 vs 0.978	-9.21	< 0.0001*
	EKAR ($N = 870$)	identity vs. inner_sym	0.766 vs 0.799	-1.67	0.0952
		identity vs. central	0.766 vs 0.951	-11.07	< 0.0001*
		identity vs. extreme	0.766 vs 0.954	-11.30	< 0.0001*
		central vs. extreme	0.951 vs 0.954	-0.29	0.7686
	U2 ($N = 228$)	identity vs. inner_sym	0.947 vs 0.950	-0.14	0.8848
		identity vs. central	0.947 vs 0.988	-2.47	0.0136*
		identity vs. extreme	0.947 vs 0.990	-2.63	0.0086*
		central vs. extreme	0.988 vs 0.990	-0.20	0.8378
	U4 ($N = 432$)	identity vs. inner_sym	0.807 vs 0.835	-1.07	0.2831
		identity vs. central	0.807 vs 0.864	-2.26	0.0238*
		identity vs. extreme	0.807 vs 0.859	-2.05	0.0405*
		central vs. extreme	0.864 vs 0.859	0.21	0.8315

* Statistically significant at $\alpha = 0.05$.

Table 9: Two-Proportion Z-Test comparing permutation performance for gpt-4.1-mini and gpt-5.2.