

Testing proportional sentence analogies on SATS: from vector offsets to conditional generation

Yves Blain-Montesano*, Philippe Langlais

Université de Montréal, Canada

Abstract

Analogy, the correspondence between two things, has been hailed as an important reasoning capability. Proportional analogy, denoted $a : b :: c : d$ and read “ a is to b as c is to d ”, is a special case of this where a correspondence is made in the relation that holds between the two elements of two different pairs (a, b) and (c, d) . A common task in this domain is to solve for the conclusion d given the premise (a, b, c) . Few datasets of proportional sentence analogies exist which aren’t limited in the variety of relations we hope to capture, including those of a semantic or common sense nature. In this work, we curate a dataset of pairs of sentences for which such relations hold, constructing 78,400 analogies involving 32 relations. Our experiments demonstrate little basis for analogical reasoning of this kind using offsets of vector embeddings, in agreement with previous work, for retrieval and generation. We leverage the representations learned by pretrained language models, and the natural language input interface they provide, to solve analogies by generating from a prompt, as well as finetuned in a sequence-to-sequence setting. From this we gain insights into their failure modes and disparity in task ability.

Keywords

natural language processing, sentence analogy, text generation, sentence embedding

1. Introduction

We study the analogy-solving ability of language models and representations derived therefrom, specifically as regards proportional sentence analogies (those said “ A is to B as C is to D ” and denoted $a : b :: c : d$) formed on the basis of syntax, semantics, and encyclopedic knowledge. That is, we are interested in finding the conclusion d —given the premise (a, b, c) —of paired natural language sentences (a, b) and (c, d) which are analogous in ways that can be considered common sense to many humans. For example, in the analogy $I'm happy : I'm angry :: I sang : I yelled$, there is a relation of opposition of mood that holds which is relatively intuitive, though ambiguous as the solution $I yelled$ need not be unique. This is the kind of analogy we refer to in our work.

The framework of proportional analogies is drawn from classical mathematical analogies such as $5 - 3 = 12 - 10$ or $\frac{2}{1} = \frac{4}{2}$ in addition to conceptual ones, for example when Aristotle writes “as old age is to life, so is evening to day” [1]. These analogies are a quaternary relation

IARML@IJCAI'2024: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI'2024, August, 2024, Jeju, South Korea


*Corresponding author.

✉ yves.blain-montesano@umontreal.ca (Y. Blain-Montesano); felipe@iro.umontreal.ca (P. Langlais)

🌐 <https://www-labs.iro.umontreal.ca/~felipe/> (P. Langlais)

🆔 0000-0002-2602-2019 (P. Langlais)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

where for some $(a, b, c, d) \in X^4$ we denote it with $a : b :: c : d$ if and only if some relation $R \subseteq X^2$ holds both for (a, b) and (c, d) :¹

$$a : b :: c : d \iff \exists R \subseteq X^2 \text{ s.t. } aRb \wedge cRd$$

Previous work has postulated equivalence relations between some permutations of the terms in $a : b :: c : d$ which apply in formal languages and Boolean logic. Instead, we follow [2] in distinguishing this “parallelogram model” of analogy from NLP-style analogies. Rather than leveraging domain-specific properties to prove the equivalence of analogy permutations, the latter kind only admits analogies whose relations R are in a predetermined set S , such as the Google analogy set [3] or BATS (Bigger Analogy Test Set) [4]. There is no guarantee of symmetry, i.e. $R \in S \iff R^{-1} \in S$ (hence $b : a :: d : c$), nor of central permutation, i.e. $\forall R_i \in S \text{ s.t. } a : b :: c : d, \exists R_j \in S \text{ s.t. } a : c :: b : d$,² as these must be deliberately included in S .

We narrow our interest onto distributional models of natural language, where interest in proportional analogies has persisted since the popularizing work of Mikolov et al. [5] on word embeddings, interest which has extended to sentence embeddings. In this tradition, the typical method for solving proportional analogies, called the **vector offset** or **vector arithmetic** method, is to take the embedding $E(\cdot)$ for each term of the analogy under a model of arithmetic analogy: assuming the embeddings of (a, b) and (c, d) define a parallelogram and thus have equal offset vectors $E(b) - E(a) = E(d) - E(c)$, then $E(d) = E(c) + E(b) - E(a)$. As the predicted d vector will rarely correspond exactly to an embedding, a solution will be found by retrieval among a set of candidates C from a vector space model (VSM) by maximum cosine similarity, in which case the method is called **3CosAdd**. In general, we will call a **vector solver** any function f which aims to predict $E(d) = f(E(a), E(b), E(c))$. Above, C refers to the set of candidates for retrieval. In word analogy, this is the vocabulary for which we have embeddings. As this work regards sentence analogy, C is a set of sentences, which we’ll take as the unique sentences from a set of analogies. Vector arithmetic has been used to retrieve [6] as well as decode [7] the solution to sentence analogies. Yet, analogical reasoning with a vector offset has been shown to lack empirical and theoretical justification by its assumption of binary opposition of relations, the commutative properties of addition used by the offset, and its dependence on spurious properties of the embedding vectors used [8, 9].

While sentence embeddings from pooled token encodings produced by large language models (LLMs) of the Transformer architecture are often used [6, 10, 11, 12, 13], less attention has been given to conditionally generating the solution d by directly providing the premises (a, b, c) as input to the language model. Language models have been shown to display impressive performance including at unseen tasks in a few-shot setting, scaling with size [14]. Such an approach may offer the improvements of (1) retaining all information rather than pooling encodings, and (2) using the natural language interface which the model has conveniently already learned in order to specify the problem context. As such we need not enforce a model of proportional analogy such as vector arithmetic, nor require sufficient training examples to fit one on the pooled embedding space of a language model which is already pretrained.

¹We define it this way while noting that trivial analogies under e.g. $R = X^2$ are not typically of interest.

²In *pillow : bed :: headrest : car*, the relation is between head supports and their associated locations. It is less clear what relation holds between a bed and a car.

In light of this, our contributions are as follows. We evaluate various models on their ability to solve proportional analogies of natural language sentences held in varied (e.g. syntactic, semantic) relational proportions. For this purpose, we elaborate a dataset of relational sentence analogies described in Section 2. Extracting sentence embeddings from several language models, we evaluate vector solvers via retrieval (see Section 3) and compare this approach to generating solutions. For generation, we compare decoding from a vector solver prediction to finetuning a pretrained language model on our analogies as a sequence-to-sequence task, as well as prompting pretrained LLMs ranging into the billion-parameter size, as described in Section 4.

2. Dataset

While several sentence analogy datasets have been elaborated in previous works, we do not overall find them sufficiently adapted to evaluating analogical reasoning on natural language held in intuitive relations.

Zhu and de Melo [6] use NLI entailment and negation pairs such as *There is no skilled person riding a bicycle on one wheel. : A skilled person is riding a bicycle on one wheel.* and template sentences pairs filled with word analogies from the Google analogy set [3] such as *I've never been to Amman. : I've never been to Jordan.* However, it has been shown that the Google analogy set's imbalanced relations are dominated by easier geographical and morphological word pairs such as *possible:impossible* or *long:longer* which can skew the evaluation of analogical reasoning with embeddings, while worse performance was found against BATS [4], which is a broader, more balanced selection of relational word analogies. We expect the minimal variation between template sentences coupled with the lack of distractor sentences (see Section 3.2) in their candidate set to preserve the Google set's skew.

Some previous work [7, 11] has used a set of 5,607 semantico-formal analogies extracted from the Tatoeba corpus [15]. This set does not use predefined relations, though contains many surface changes (e.g. pluralization, present versus past tense), and other unclear semantic analogies such as the example *I do not need a wheelchair. : I do not need a girlfriend. :: I do not have a cat. : I do not have a boyfriend.*

Others collate analogies from existing resources [10, 13], for example the DSBATS set [12] of paired definition sentences corresponding to encyclopedic relations from BATS [4], but these remain relatively restricted in the breadth of relations considered.

For these reasons, we collect and indeed write our own relational sentence pairs to ensure an (1) adequate diversity of relations and sentences, and (2) that they constitute relatively "valid" analogies by their manual curation. Our sentence analogy test set (SATS)³ is a collection of 32 relation sets, listed in Table 1, each of 50 pairs of sentences (totalling 1,600 pairs from 3,024 unique sentences) from which 2450 non-identity quadruples (i.e. not $a : b :: a : b$) can be made by combining pairs of a same relation, for a total of 78,400 analogies. We coarsely categorize each relation set as Encyclopedic, Lexical, Syntactic, or Semantic for later aggregation of results.

We manually construct syntactic pairs such as *My parents **turned on** the TV. : My parents **turned the TV on*** (from the canonical-verb-particle-movement relation set), and also include relations of declarative sentences and questions taken from the QA2D dataset [16], which are

³Available at <https://github.com/rali-udem/sats-sentence-analogy>.

Table 1
SATS relations by category and split

Training	Encyclopedic	hypernym-animal	Test	Encyclopedic	capital-country
	Encyclopedic	misc-hypernym		Encyclopedic	country-language
	Encyclopedic	person-occupation		Encyclopedic	invention-creator
	Lexical	present-past		Encyclopedic	member-band
	Semantic	informal-formal		Lexical	idiom-literal
	Semantic	sentence-opposite		Lexical	numeral-spelled
	Semantic	sentiment-good-bad		Lexical	numeric-approximation
	Syntactic	because-so		Lexical	past-future
	Syntactic	canonical-extrapolation		Semantic	cause-effect
	Syntactic	qa2d-declarative-howmany		Semantic	description-state
Validation	Syntactic	qa2d-declarative-when	Semantic	home-outdoors	
	Syntactic	qa2d-declarative-who	Semantic	simple-implicative-entailment	
	Encyclopedic	meronym-substance	Syntactic	active-passive	
	Lexical	present-future	Syntactic	canonical-verb-particle-movement	
	Semantic	phrasal-implicative-entailment	Syntactic	qa2d-declarative-howmuch	
	Syntactic	qa2d-declarative-what	Syntactic	qa2d-declarative-where	

formed by replacing a constituent by a wh- question word (e.g. *She opened the car door : What did she open?*).⁴ Those in the Semantic category have a more ambiguous relation, such as the home-outdoors relation (e.g. *He gave himself a haircut in the bathroom. : He went for a haircut at the barbershop.*) or the informal-formal relation (e.g. *We gotta get going. : We need to start moving.*). Encyclopedic relations are inspired by the encyclopedic and lexicographic relation sets in BATS [4]. Here, where possible, sentences are chosen which match a corresponding word pair,⁵ specifically for the hypernym-animal, misc-hypernym, person-occupation, meronym-substance, capital-country, and country-language relations. The sentences themselves are collected from the first sentences of Wikipedia articles corresponding to those words, e.g. from the meronym-substance relation we have *A lens is a transmissive optical device which ... : Glass is a non-crystalline ... :: A mirror or looking glass is ... : Glass is ...*, from which it can also be noted that several such relations are non-unique.

We also note that many attempts to train models on proportional analogies, while they use data splits, do so within relation types [6, 7, 12, 17], rather than between them, thus confounding whether they generalize analogical reasoning as such to unseen relations. For this reason we split our data as outlined in Table 1.

3. Retrieving solutions in a VSM

The first approach we examine is to retrieve the solutions to our analogies from a set of candidate sentences by maximum cosine similarity. The two questions this experiment seeks to answer are: (1) How do different models' embeddings lend themselves to analogical reasoning? (2) Do vector-based models of analogy (such as vector arithmetic) vary in their ability to recover useful features for this purpose?

⁴This example pair is not contained in SATS.

⁵For example, for the BATS pair *Mozart : composer*, we will take the first sentence from the corresponding Wikipedia articles for "Mozart", and for "Composer"

3.1. Models

To the first point, the embeddings we use are those mean-pooled from the final encodings of transformers,⁶ namely Flan-T5⁷ [19], BERT [20], RoBERTa [21], DeBERTa V1 and V3⁸ [22, 23], SBERT⁹ [18], and summed embeddings from the English Common Crawl CBOV FastText embeddings [24]. We use the Base checkpoint for all models where applicable.

To the second point, we examine three vector solvers: 3CosAdd, a feedforward network—as has been attempted by others [6, 7]—and an Abelian neural network [17] which is a universal approximator of Abelian Lie operations¹⁰ (such as addition, used in vector arithmetic). Unlike 3CosAdd, the feedforward (hereafter shortened to FF) and Abelian neural networks have parameters which must be trained¹¹ for each different embedding model. Both trained solvers are feedforward neural networks composed of five blocks.

The FF blocks are composed of two affine transformations using the GELU activation function with residual connections and layer normalization. All hidden states retain the input dimensionality, though the first block has the dimensionality of the concatenation of the premise sentence embeddings $E(a) \circ E(b) \circ E(c)$, followed by an affine transformation which reduces it to the original embedding dimension.

The Abelian neural network is an invertible neural network ϕ which transforms embeddings into a space of the same dimensionality before applying the vector arithmetic method. The predicted solution is then found by using the inverse $x = \phi^{-1}(\phi(E(c)) + \phi(E(b)) - \phi(E(a)))$. For this we use the AllInOneBlock invertible block provided by the FrEIA Python module [25], which takes an inner function which we compose from two affine layers with a GELU activation function.

For each embedding model, the Abelian and FF solvers are trained on batches¹² of 8 SATS analogies, with additive Gaussian noise on the order of 10^{-2} , using the Adafactor optimizer [26] with a learning rate of 3×10^{-5} . As a loss, we use negative cosine similarity,¹³ i.e. $\text{loss}(x, y) = -\frac{x^T y}{\|x\| \cdot \|\hat{y}\|}$. We observe overfitting over two epochs of training, thus the best checkpoint is selected on the validation split using the harmonic mean of the top-1 and top-5 retrieval accuracy to encourage choosing a model that at least predicts the neighborhood of the solution.

The Flan-T5 model being an encoder-decoder, for a sequence of token length L , it performs cross-attention on all L token encoding vectors in order to autoregressively generate an output (which we explore further in Section 4). We can use this to gauge whether a model trained jointly to solve analogies in vector space and decode the solution results in better retrieval accuracy by adapting Flan-T5 to decode solely by cross-attention on a single bottleneck vector, which

⁶Mean-pooling has been found to outperform the [CLS] token’s encoding [18].

⁷Note that this is the only encoder-decoder model we use. All others have an encoder-only architecture.

⁸The latter differs by the use of a replaced token classification task rather than the usual masked language modeling objective.

⁹Specifically the all-mpnet-base-v2 checkpoint.

¹⁰This is a differentiable operation, over some set, which has associativity and commutativity, as well as identity and inverse elements.

¹¹Architecture and hyperparameters were manually tweaked. While the FF solver remains essentially a feedforward neural network, our tweaking prevents direct comparison with e.g. the feedforward solver of [7].

¹²Here and elsewhere, batches are obtained by a random shuffling of all analogies in a given split.

¹³We found regression with a mean square error loss to perform worse overall.

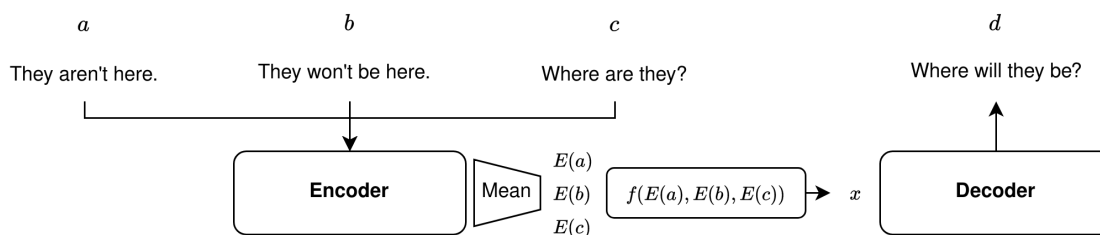


Figure 1: A visualization of the end-to-end vector solver and decoder architecture. First each sentence is encoded and its encodings mean-pooled. f is the vector solver used, e.g. the offset method. In a retrieval task, x is used as the predicted solution vector for retrieval. When training (and for generative evaluation in Section 4), the decoder performs cross-attention on x to autoregressively generate d .

we take to be a vector solver prediction $f(E(a), E(b), E(c))$. All vector solvers mentioned previously being differentiable, we can train such an end-to-end solver-decoder model (**E2E-Flan-T5**, visualized in Figure 1) for each different f , backpropagating gradients into the solvers’ as well as the pretrained model’s parameters. We use these finetuned models both for retrieval in this experiment—since at test time we can simply extract the output of the embedded vector solver—and for generation in Section 4 by decoding from the vector solver’s output.

In order to use Flan-T5 as a vector bottleneck sentence autoencoder, we must first “warm up” the pretrained model by finetuning it for this purpose using the mean-pooled encoding vector, rather than all L encodings. Using a negative log-likelihood (NLL) loss, we train it on a next-token prediction task on interleaved, padded (up to 500 tokens) batches of 72 sentences split from the Reddit comments dataset provided by the Sentence-Transformers team¹⁴ and from the Online Language Modelling [27] dataset of Wikipedia articles dated 20-12-2022.¹⁵ This finetuning was performed in a single epoch (with no intentionally repeated data) for 281,600 batches,¹⁶ at which point it reached a validation loss of 0.372 (compared to the pretrained Flan-T5’s 0.30) on some 1,024 held-out sentences. We use a learning rate of 3×10^{-5} and the same Adafactor optimizer [26] as the model was originally trained with.

This checkpoint is used as the initialization of the end-to-end model with each of the vector solvers mentioned previously, trained to generate the solutions to analogies from the SATS training split, using the same NLL objective and optimizer, a learning rate of 10^{-4} , and a batch size of 64. Observing overfitting over a single epoch, we select the best checkpoint on the validation split by METEOR score [28].

3.2. Evaluation

One aspect of evaluation that shouldn’t be overlooked is the choice of candidate set, which often excludes the premises (a, b, c), thereby artificially improving accuracy [29], as the offset vector often points away from d and indeed every other candidate but c [9]. While we use the 3,024

¹⁴<https://huggingface.co/datasets/sentence-transformers/reddit-title-body>

¹⁵<https://huggingface.co/datasets/olm/olm-wikipedia-20221220>

¹⁶At this point the model had not finished converging, due to the abundance of data.

Table 2

Retrieval accuracy of the analogy solution using the 3CosAdd and FF solvers. Columns describe accuracy under different candidate sets. Neither: premises (a, b, c) excluded, no distractors added; +distractors: distractors added; +(a,b,c): premises retained; Both: both distractors and premises included.

Model	3CosAdd				FF			
	Neither	+distractors	+(a,b,c)	Both	Neither	+distractors	+(a,b,c)	Both
E2E-Flan-T5	0.60	0.25	0.15	0.14	0.00	0.00	0.00	0.00
Flan-T5	0.61	0.21	0.10	0.08	0.36	0.10	0.15	0.07
SBERT	0.82	0.10	0.07	0.03	0.50	0.04	0.15	0.02
RoBERTa	0.57	0.27	0.15	0.12	0.23	0.12	0.12	0.08
BERT	0.61	0.16	0.07	0.05	0.32	0.12	0.13	0.08
DeBERTa	0.59	0.20	0.08	0.07	0.22	0.03	0.15	0.03
DeBERTa-V3	0.34	0.20	0.08	0.06	0.01	0.00	0.01	0.00
FastText	0.57	0.00	0.21	0.00	0.01	0.00	0.01	0.00

unique sentences in SATS as our candidates, they are very few in number. Thus, we evaluate accuracy under different candidate sets: with or without premises, and with or without added distractor sentences which we automatically create as follows.

For each unique sentence in SATS, we construct distractors by sampling whitespace-tokenized pairs of words to swap, or individual words to remove or replace with a randomly chosen nearest neighbour from the FastText vocabulary.¹⁷ For example, from *You're in trouble, friend*, we might swap to get *friend. You're trouble, in*, delete to get *You're \emptyset trouble, friend.*, or replace to get *You're **inside** trouble, friend*. We sample from each sentence at most 5 times (length permitting) per type of modification, producing 41,356 unique distractor sentences.

We test the retrieval accuracy of 3CosAdd and the trained FF and Abelian vector solvers for each embedding model. To this we add the E2E-Flan-T5 model (for each embedded vector solver). We find mostly similar retrieval accuracy for the Abelian and 3CosAdd methods.¹⁸ Hence, in Table 2, we present the retrieval accuracies for 3CosAdd and FF only.

FF, even with premises and distractors excluded, attains only roughly half the accuracy of the Abelian solver and 3CosAdd. In the hardest condition it performs at a similar level, depending on the model. We'll note that it did not converge in training for E2E-Flan-T5, attaining zero accuracy. It also obtains near-zero test accuracy for DeBERTa-V3 on all candidate sets. Insofar as it succeeds with Flan-T5, BERT, and RoBERTa embeddings, the majority of its accuracy comes from the declarative-question, canonical-verb-particle-movement, numeral-spelled, and numeric-approximation relations.

This concentration of accuracy in only some relations is not a particularity of the FF solver or of training. 3CosAdd and the Abelian solver obtain near or above 20% accuracy for Lexical and Syntactic relations, depending on the embedding model, using the most difficult candidate set, and near zero for Encyclopedic and Semantic ones. Relatedly, we find that for base sentence pairs (a, b) in SATS, the nearest neighbor to a is b over 80% of the time for most embedding models

¹⁷We select the top 20 nearest neighbours with fewer than 25 characters.

¹⁸We surmise that the Abelian solver's invertible neural network has preserved much of the original embedding space. Perhaps the algebraic properties of any Abelian operation will impede embedding-based solvers.

Table 3

Example analogy from the member-band relation and top two retrieved solutions for SBERT and 3CosAdd. When the premises are removed, the correct solution is found. When they or their distractors are included, it is one of the premises which is retrieved. In the hardest case, it is in fact a distractor for c which ranks first.

Gene Simmons ... : <u>Kiss</u> ... :: <u>Bradford Phillip Delson</u> ... : Linkin Park ...		
Neither	+(a,b,c)	+(a,b,c) +distractors
Linkin Park is an American rock band ...	<u>Bradford Phillip Delson</u> (born December 1, 1977) is an American musician, best known as ...	<u>Bradford Phillip Delson</u> 1, 1977) is an musician, best known as ...
Def Leppard are an English ...	<u>Kiss</u> (stylized as ...) is an ...	Bradford Phillip Delson ...

for Lexical and Syntactic relations, 30-50% for Semantic ones, and roughly 0% for Encyclopedic ones—except for SBERT, where it’s 46%. Nor is the decrease in accuracy simply an effect of our distractors. It is still the case that when a distractor is picked, it is of the premises (a, b, c) rather than of d , as exemplified in Table 3. Such caveats are in line with results from the word analogy literature, where keeping premises in the candidate set reduces accuracy immensely [29], and where some surface relations are preferred [4].

We will note that when premises are included, FastText embeddings attain 40% and 46% for the Lexical and Syntactic categories but near zero in other categories, and even this drops to zero when distractors are added, as summed word embeddings cannot account for word order. However, this summation does mean that vector offsets can correspond to differences in used vocabulary.

It has been shown by [9] for lexical analogies that the cosine similarity to d of the offset prediction $c + b - a$ is dependent on the similarity of the offsets of the two pairs (a, b) and (c, d) and the similarity of the terms within a single pair (c, d) , which can be high or low for spurious reasons. Instead, they introduce the pairing consistency score (PCS), which measures the linear distinguishability of a relation for a particular embedding space. To compute PCS for a given relation, we treat sets of offset vectors of true pairs and those of false shuffled pairs as observations for a binary classification task where the predictive score of each set of offset

Table 4

PCS for test split relations by model and category. Higher is better, 0.5 is chance level.

	Encyclopedic	Lexical	Semantic	Syntactic
E2E-Flan-T5 (arithmetic)	0.52	0.85	0.64	0.94
Flan-T5	0.55	0.83	0.62	0.86
SBERT	0.82	0.80	0.61	0.74
RoBERTa	0.54	0.83	0.66	0.87
BERT	0.62	0.84	0.63	0.81
DeBERTa	0.58	0.83	0.67	0.89
DeBERTa-V3	0.51	0.68	0.58	0.65
FastText	0.57	0.86	0.62	0.87

vectors is the average cosine similarity of all $\binom{n}{2}$ combinations of offsets. With this the area under the receiver operating characteristic (AUROC) is computed.¹⁹ As with [9], we take PCS as the average AUROC over $N = 50$ different false pair shufflings per relation.

Table 4 shows how embedding models have high PCS, i.e. linear separability of true offsets, for Lexical and Syntactic relations. PCS is much closer to chance for Semantic and, most notably, Encyclopedic relations. There are two exceptions to this. SBERT has exceptionally high PCS for Encyclopedic relations, likely due to its contrastive training on sources such as Wikipedia, whereas it has been found that contrastive loss leads to parallel offsets [30]. The second exception is DeBERTa-V3, which has exceptionally low PCS, and yet is a performant state of the art model. This is likely due to DeBERTa-V3’s replaced token detection task, whereas all other embedding models are trained on a language modeling objective. The latter objective has been shown to result in linear representations [31]. We will suggest from these results that even when an embedding space adequately represents a relation in terms of parallel offsets—and even this should not be taken for granted among otherwise performant models like DeBERTa-V3—vector arithmetic fails to successfully utilize this regularity, often leading to the retrieval of a premise.

4. Generating solutions

We explore several approaches to generating the solution to analogies. First, we directly decode it with the E2E-Flan-T5 model.²⁰ Second, we finetune the Flan-T5 Base and Large checkpoints on our analogies as a sequence-to-sequence (Seq2Seq) task. Third, we directly prompt the Flan-T5 model in its parameter sizes of Base (250M), Large (780M), XL (3B) and XXL (11B). When sampling, we use η -sampling [32] with $\eta = 6 \times 10^{-4}$. We use BLEU [33], METEOR [28], word error rate (WER), and exact match accuracy, which we report in Table 5.

The Seq2Seq Flan-T5 models (see Figure 2) are finetuned on SATS analogies to conditionally generate d from (a, b, c) . First, we input to the encoder portion of the transformer the concate-

Table 5

SATS test split generation metrics. Copy rates refer to the exact match of the generated prediction with a premise sentence.

		Exact Match \uparrow	WER \downarrow	METEOR \uparrow	BLEU \uparrow
(a, b) Baseline		—	0.86	0.54	0.23
E2E-Flan-T5	Base	0.03	0.92	0.38	0.14
Seq2Seq	Base	0.00	1.10	0.31	0.10
	Large	0.01	0.92	0.42	0.16
Prompted	Base	0.00	1.84	0.27	0.05
	Large	0.00	0.94	0.40	0.14
	XL	0.01	0.98	0.37	0.13
	XXL	0.02	0.94	0.41	0.16

¹⁹Thus, if a relation is linearly distinguishable in an embedding space, as we increase the threshold for positive classification from 0 to 1, we expect the similarity of false offsets (false positives) to fall below it quickly, and that of true offsets (true positives) to do so slowly.

²⁰We use the vector arithmetic variant since we find it performs identically to the Abelian one.

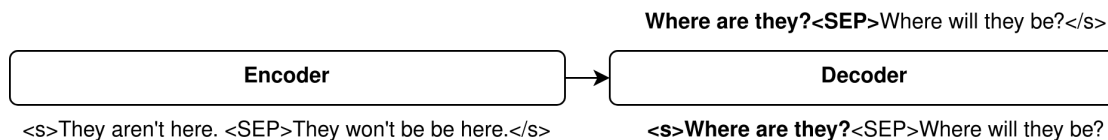


Figure 2: Depiction of the sequence-to-sequence analogy task. The bolded text refers to sentence c which is prepended to the generated solution.

nated input sentences $a \oplus b$. The decoder portion of the transformer performs cross-attention on the encodings of $a \oplus b$, and auto-regressively generates d , given the c premise sentence which is prepended to decoder output (and masked from the loss computation, as it is an input). The rationale is that the decoder may extract the change in the pair (a, b) and complete (c, d) in a similar fashion.²¹ We use a NLL loss for 3 epochs with batches of 64, using the Adafactor optimizer with a constant learning rate of 3×10^{-5} for the Base models and 10^{-5} for the Large models. NLL loss on the validation set is used to pick the best checkpoint.

Seq2Seq improves for the Large model, though its output mostly consists of attempted copies of premise sentences. While E2E-Flan-T5 obtains better evaluation metrics, we can note (see Table 6) that it has also overfit to training examples from the person-occupation relation. Its output is often similar to training data, if not nonsensical.

Given that the pretrained Flan-T5 model is trained on text-to-text instruction-prompted tasks, we informally settled on a few-shot prompt incorporating three new example analogies not found in SATS, which elicits the model to output an appropriate response separated by a (FINAL ANSWER) substring without any finetuning. We automatically separate the solution from the generated sequence using the final (if any) occurrence of this answer separator.

We find that as model size increases the exact match accuracy increases. Copy rates hover around or below 5%, only shooting up to a c copy rate of 14% and 24% for the Flan-T5-Large Seq2Seq and Prompted outputs, respectively, returning to low levels for the XL and XXL sizes. Importantly, Table 6 shows that the largest model checkpoint exhibits understanding of the task, though it too ultimately copies the c premise after repeating the answer separator substring.

As an informal experiment, we prompted ChatGPT-3.5, which is finetuned by reinforcement learning with human feedback [34], and davinci-002, which is trained solely on the language modeling objective, both of which we expect to have 175 billion parameters [35]. As shown in Table 6, while davinci-002 falls into the familiar premise copying trap, ChatGPT-3.5 answers ideally, indicating that a language modeling objective alone may not suffice for immense models.

5. Conclusion

This paper presented a SATS, a novel sentence-level proportional analogy dataset which we use to evaluate common methods of analogical reasoning using vector embeddings which originate in work on word analogies. We identify similar gaps in the ability of such methods to solve sentence analogies as have previously been found for lexical analogies. To what

²¹Providing (a, b, c) as one to the encoder could potentially improve upon this, though we opted not to.

Table 6

Comparison of example analogies and different models' predictions. E2E-Flan-T5 reproduces an unrelated sentence from the person-occupation relation in the training set. Flan-T5-XXL first correctly identifies the relation in question, though later repeats the answer separator and copies the *c* premise sentence, thus failing. The davinci-002 model also copies the *c* premise sentence. ChatGPT-3.5, however, correctly identifies the relation and solves the analogy, though the solution was manually extracted as it disregarded the answer separator.

E2E-Flan-T5 (capital-country)			
Taipei ...	:	Taiwan ...	:: Berlin ... : Germany ...
*An author is someone who writes music ...			
Flan-T5-XXL (capital-country)			
Lisbon ...	:	Portugal ...	:: <u>Manila ...</u> : The Philippines ...
(FINAL ANSWER) The change between sentences one and two is the country that Lisbon is the capital and largest city of. ... (FINAL ANSWER) The city of Manila			
OpenAI (person-occupation)			
John Christopher Depp ...	:	<u>An actor or actress is</u>	:: Christopher Columbus ... : An explorer is a ...
davinci-002			
(FINAL ANSWER) <u>An actor or actress is a person who ...</u>			
ChatGPT-3.5			
An explorer or navigator is a person who completes voyages across the Atlantic ...			

extent analogical reasoning can be achieved with sentence embeddings is uncertain. While we evaluate a variety of approaches, further exploration is necessary. The use of 3CosAdd could be compared to 3CosMul [36] or retrieval by pairwise similarity [37]. Trained vector solvers may see improvements from exhaustive hyperparameter search. Comparisons to existing neural solver architectures [7, 11] should be made.

We experiment with solving sentence analogies generatively using the Flan-T5 language model by finetuning on them as a sequence-to-sequence task, and by few-shot prompting, finding improvements as models grow into the many billion parameter range. However, better generative metrics should be found for predictions and targets which have high surface similarity. Prompt engineering should also improve the performance of the pretrained Flan-T5 checkpoints by inducing less premise copying, better chain-of-thought reasoning, and fewer incidences of answer separator repetition

Our dataset is an important limitation. The reoccurrence of sentences in training analogies may lead to overfitting, and we find it likely that their small quantity is the main contributor to the poor performance of trained solvers. Finally, SATS should be improved by a principled method for collecting and validating a variety of humanlike proportional analogies.

References

- [1] N. Barbot, L. Miclet, H. Prade, Analogy between concepts, *Artificial Intelligence* 275 (2019) 487–539. doi:10.1016/j.artint.2019.06.008.

- [2] S. Afantenos, S. Lim, H. Prade, G. Richard, Theoretical study and empirical investigation of sentence analogies, in: M. Couceiro, P.-A. Murena (Eds.), IJCAI-ECAI Workshop: Workshop on the Interactions between Analogical Reasoning and Machine Learning (IAMRL 2022) @ IJCAI-ECAI 2022, volume 3174 of *CEUR Proceedings*, CEUR-WS.org, Vienna, Austria, 2022, pp. 15–28.
- [3] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Y. Bengio, Y. LeCun (Eds.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013.
- [4] A. Gladkova, A. Drozd, S. Matsuoka, Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't., in: Proceedings of the NAACL Student Research Workshop, Association for Computational Linguistics, San Diego, California, 2016, pp. 8–15. doi:10.18653/v1/N16-2002.
- [5] T. Mikolov, W.-t. Yih, G. Zweig, 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, Association for Computational Linguistics, Atlanta, Georgia, 2013, pp. 746–751.
- [6] X. Zhu, G. de Melo, Sentence Analogies: Exploring Linguistic Relationships and Regularities in Sentence Embeddings, 2020. arXiv:2003.04036.
- [7] L. Wang, Y. Lepage, Vector-to-Sequence Models for Sentence Analogies, in: 2020 International Conference on Advanced Computer Science and Information Systems (ICAC-SIS), IEEE, Depok, Indonesia, 2020, pp. 441–446. doi:10.1109/ICAC-SIS51025.2020.9263191.
- [8] A. Rogers, A. Drozd, B. Li, The (too Many) Problems of Analogical Reasoning with Word Vectors, in: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 135–148. doi:10.18653/v1/S17-1017.
- [9] L. Fournier, E. Dupoux, E. Dunbar, Analogies minus analogy test: Measuring regularities in word embeddings, in: Proceedings of the 24th Conference on Computational Natural Language Learning, Association for Computational Linguistics, Online, 2020, pp. 365–375. doi:10.18653/v1/2020.conll-1.29.
- [10] T. Barbero, S. D. Afantenos, Some preliminary results on analogies between sentences using contextual and non-contextual embeddings, in: M. Couceiro, S. D. Afantenos, P.-A. Murena (Eds.), Proceedings of the Workshop on the Interactions between Analogical Reasoning and Machine Learning Co-Located with International Joint Conference on Artificial Intelligence (IJCAI 2023), Macau, China, August 21, 2023, volume 3492 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 34–45.
- [11] W. Mao, Y. Lepage, Embedding-to-embedding method based on autoencoder for solving sentence analogies, in: L. Malburg, D. Verma (Eds.), Proceedings of the Workshops at the 31st International Conference on Case-Based Reasoning (ICCB-WS 2023) Co-Located with the 31st International Conference on Case-Based Reasoning (ICCB-2023), Aberdeen, Scotland, UK, July 17, 2023, volume 3438 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 15–26.
- [12] Q. Zhang, Y. Lepage, Improving sentence embedding with sentence relationships from

- word analogies, in: L. Malburg, D. Verma (Eds.), Proceedings of the Workshops at the 31st International Conference on Case-Based Reasoning (ICCBR-WS 2023) Co-Located with the 31st International Conference on Case-Based Reasoning (ICCBR 2023), Aberdeen, Scotland, UK, July 17, 2023, volume 3438 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 43–53.
- [13] T. Wijesiriwardene, R. Wickramarachchi, B. Gajera, S. Gowaikar, C. Gupta, A. Chadha, A. N. Reganti, A. Sheth, A. Das, ANALOGICAL - A Novel Benchmark for Long Text Analogy Evaluation in Large Language Models, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 3534–3549. doi:10.18653/v1/2023.findings-acl.218.
- [14] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H.-h. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus, Emergent abilities of large language models, *Trans. Mach. Learn. Res.* 2022 (2022).
- [15] Y. Lepage, Analogies between short sentences: A semantico-formal approach, in: Human Language Technology. Challenges for Computer Science and Linguistics: 9th Language and Technology Conference, LTC 2019, Poznan, Poland, May 17–19, 2019, Revised Selected Papers, Springer-Verlag, Berlin, Heidelberg, 2019, pp. 163–179. doi:10.1007/978-3-031-05328-3_11.
- [16] D. Demszky, K. Guu, P. Liang, Transforming Question Answering Datasets Into Natural Language Inference Datasets (2018). doi:10.48550/ARXIV.1809.02922.
- [17] K. Abe, T. Maehara, I. Sato, Abelian Neural Networks, 2021. arXiv:2102.12232.
- [18] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-Networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.
- [19] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling Instruction-Finetuned Language Models (2022). doi:10.48550/ARXIV.2210.11416.
- [20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
- [21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. arXiv:1907.11692.
- [22] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with Disentangled Attention (2020). doi:10.48550/ARXIV.2006.03654.
- [23] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing (2021). doi:10.48550/ARXIV.2111.09543.
- [24] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157

- languages, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.
- [25] L. Ardizzone, T. Bungert, F. Draxler, U. Köthe, J. Kruse, R. Schmier, P. Sorrenson, Framework for easily invertible architectures (FrEIA), 2018/2022.
- [26] N. Shazeer, M. Stern, Adafactor: Adaptive Learning Rates with Sublinear Memory Cost, 2018. [arXiv:1804.04235](https://arxiv.org/abs/1804.04235).
- [27] T. Thrush, H. Ngo, N. Lambert, D. Kiela, Online language modelling data pipeline, 2022.
- [28] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: J. Goldstein, A. Lavie, C.-Y. Lin, C. Voss (Eds.), Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 65–72.
- [29] N. Schlueter, The Word Analogy Testing Caveat, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 242–246. doi:10.18653/v1/N18-2039.
- [30] N. Ri, F.-T. Lee, N. Verma, Contrastive Loss is All You Need to Recover Analogies as Parallel Lines, in: Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 164–173. doi:10.18653/v1/2023.rep4nlp-1.14.
- [31] Y. Jiang, G. Rajendran, P. Ravikumar, B. Aragam, V. Veitch, On the Origins of Linear Representations in Large Language Models, 2024. [arXiv:2403.03867](https://arxiv.org/abs/2403.03867).
- [32] J. Hewitt, C. Manning, P. Liang, Truncation sampling as language model desmoothing, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3414–3427. doi:10.18653/v1/2022.findings-emnlp.249.
- [33] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: A method for automatic evaluation of machine translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. doi:10.3115/1073083.1073135.
- [34] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. [arXiv:2203.02155](https://arxiv.org/abs/2203.02155).
- [35] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, X. Huang, A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models, 2023. [arXiv:2303.10420](https://arxiv.org/abs/2303.10420).
- [36] O. Levy, Y. Goldberg, Linguistic Regularities in Sparse and Explicit Word Representations, in: Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Ann Arbor, Michigan, 2014, pp. 171–180. doi:10.3115/v1/W14-1618.
- [37] P. D. Turney, Distributional Semantics Beyond Words: Supervised Learning of Analogy and Paraphrase, *TACL* 1 (2013) 353–366. doi:10.1162/tac1_a_00233.