Université de Montréal

Evaluating approaches to solving proportional sentence analogies

par

Yves Blain-Montesano

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

Orientation Intelligence artificielle

February 8, 2024

© Yves Blain-Montesano, 2024

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Evaluating approaches to solving proportional sentence analogies

présenté par

Yves Blain-Montesano

a été évalué par un jury composé des personnes suivantes :

Bang Liu

(président-rapporteur)

Philippe Langlais

(directeur de recherche)

Stefan Monnier

(membre du jury)

Résumé

L'analogie, c'est-à-dire une correspondance entre deux entités, est considérée une capacité de raisonnement importante. L'analogie proportionnelle, écrite $a : b :: c : d$ et qui se lit "a est à b ce que c est à d", en est un cas particulier où la correspondance tient de par la relation entre les éléments de deux paires d'objets. Le mémoire évalue certaines méthodes issues de l'usage de représentations distributionnelles vectorielles dans la résolution d'analogies proportionnelles verbales et les mène à leur prolongement naturel, la phrase. Nous ciblons la compétence de modèles de langue et des représentations qui peuvent en être extraites à la résolution d'analogies proportionnelles formées sur la base de relations syntaxiques, sémantiques, ou de connaissance encyclopédique. Peu d'ensembles de données existent pour les analogies de phrase et sinon comprennent pour la plupart des analogies au niveau de la forme, composées de phrases construites à partir de gabarits, ou bien variant peu dans les relations sémantiques qui tiennent entre les phrases. Nous construisons donc un ensemble de données contenant des phrases en paires relationnelles qui nous permet de construire des analogies en appariant deux paires. Nous essayons différentes variations de méthodes qui comportent un objectif de recouvrement par un modèle vectoriel. D'autres méthodes de résolution d'analogies proportionnelles sont explorées par voie de génération de texte. Nous expérimentons par le peaufinement du modèle de langue Flan-T5, pré-entraîné sur des paires instruction-réponse, sur nos analogies par une tâche séquence à séquence, ainsi que par l'incitation avec peu d'exemples en utilisant des versions de ce modèle en variant la capacité jusque dans la gamme des milliards de paramètres. En somme, la performance observée est faible pour toutes les tâches. Nous concluons, de l'utilisation de plongements de phrase, quelques mises en garde similaires à celles que l'on trouve avec la résolution d'analogies verbales par plongements lexicaux. Nos expérimentations génératives démontrent l'importance de données à la fois de bonne qualité et de bonne quantité, ainsi que le potentiel de l'apprentissage en contexte. Nous ajoutons à cela un aperçu qualitatif de la disparité entre l'habileté de modèles probabilistes entraînés pour prédire, à partir d'une instruction, la séquence correcte, et celle d'un modèle peaufiné par la méthode d'apprentissage par renforcement avec commentaires humains, à savoir ChatGPT.

Mots-clés : résolution d'analogie, analogie de phrase, traitement automatique des langues naturelles, plongement de phrase, génération de texte

Abstract

Analogy, the correspondence between two things, has been hailed as an important reasoning capability. Proportional analogy, denoted $a : b :: c : d$, 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 elements of two pairs. This thesis evaluates methods originating in the recent use of distributional vector representations for solving four-part word analogies, bringing them to their natural extension, sentences. Few datasets of proportional sentence analogies exist, typically comprising purely formal analogies or sentences constructed by templates, and where semantic relations are typically limited in the variety we would hope to capture. Thus, for the purposes of our experiments, we curate a dataset of pairs of sentences for which a given relation holds and from which analogies can be constructed by matching pairs within a relation together. We target the analogy-solving ability of language models and representations derived therefrom, specifically as regards proportional sentence analogies formed on the basis of syntax, semantics, or encyclopedic knowledge. Different variations on previous methods are explored, all based on retrieval of the solution in a vector space model. Other methods of solving proportional sentence analogies by generation are attempted. We experiment with finetuning the instruction-trained Flan-T5 language model on sentence analogies as a sequence-to-sequence task, as well as prompting model checkpoints up into the billion-parameter range with few-shot examples. Overall performance at the task is poor in both settings. We find that similar caveats which apply to analogical reasoning with word vectors apply to sentence embeddings as well. Our generative experiments show the importance of data of suitable quality and quantity, as well the potential of in-context learning. Some qualitative insights are shown as to the disparity in task ability of instruction-trained probabilistic language models and one finetuned by reinforcement learning with human feedback, namely ChatGPT.

Keywords : analogy solving, sentence analogy, natural language processing, sentence embedding, text generation

Contents

Résumé v

Abstract vii

List of Tables xiii

List of Figures xv

List of acronyms and abbreviations xvii

Acknowledgements xxi

Introduction 1

Chapter 1. Related work 3

1.1. Background 3

1.2. Probabilistic language models 9

1.3. Analogy 10

1.4. Word analogy in vector space models 12
1.4.1. Word embedding 13
1.4.2. Vector offset method 13
1.4.3. Word analogy datasets 14
1.4.4. Offset method caveats 15
1.4.5. Linear regularities 16
1.4.6. Other solving methods 18

1.5. Sequence analogy 20
1.5.1. Sentence embedding 20
1.5.2. Retrieval using vector offset 21
1.5.3. Decoding from an embedding 23
1.5.4. Classification 24

1.5.5. Figurative and predictive analogy . . . 26

Chapter 2. Sentence analogy test set . . . 29

2.1. Dataset construction . . . 30

2.2. Limitations and bias . . . 33

Chapter 3. Experiments . . . 39

3.1. Pretrained models . . . 40
3.1.1. FastText . . . 40
3.1.2. BERT . . . 40
3.1.3. RoBERTa . . . 41
3.1.4. DeBERTa . . . 41
3.1.5. Sentence-BERT . . . 41
3.1.6. Instructor . . . 42
3.1.7. Flan-T5 . . . 42

3.2. Finetuned Flan-T5 autoencoder . . . 43

3.3. Vector solvers . . . 44
3.3.1. Feedforward solver . . . 44
3.3.2. Abelian solver . . . 45
3.3.3. Training . . . 45
3.3.4. End-to-end decoder solver . . . 46

3.4. Retrieval task . . . 46

3.5. Generative task . . . 48
3.5.1. Metrics . . . 48
3.5.2. Finetuning Flan-T5 for sequence-to-sequence analogies . . . 49
3.5.3. Few-shot Flan-T5 solver . . . 50

Chapter 4. Analysis . . . 53

4.1. Retrieval . . . 53
4.1.1. Pairing consistency score . . . 54
4.1.2. Candidate sets . . . 55
4.1.3. Pair and offset similarity . . . 56

4.1.4. Takeaways 58
4.2. Generation 62
4.2.1. Results 62
4.2.2. Limitations 65
Chapter 5. Conclusion 71
References 75
Appendix A. Tables and figures 89

List of Tables

1.1 Sentence pair count and examples per relation for the analogy dataset of Zhu and de Melo (2020) . . . 22

2.1 SATS relations by category and split . . . 30

2.2 Number of unique sentences for many-to-one relations . . . 32

2.3 SATS sentence length statistics . . . 33

2.4 Jaccard similarity of SATS pairs by category . . . 34

2.5 Examples of encyclopedic SATS sentence pairs per relation, with phonetic transcriptions replaced with ellipses due to typesetting errors . . . 35

2.6 Examples of lexical SATS sentence pairs per relation . . . 36

2.7 Examples of syntactic SATS sentence pairs per relation . . . 36

2.8 Examples of semantic SATS sentence pairs per relation . . . 37

4.1 Pairing consistency scores by model and category . . . 54

4.2 SATS test accuracy by category using the Abelian solver . . . 55

4.3 Nearest neighbour retrieval baseline on SATS test split (a,b) pairs . . . 56

4.4 Example analogy and top retrieved solutions under different candidate sets for Sentence-BERT using the arithmetic solver . . . 58

4.5 Retrieval accuracy per model on all SATS splits . . . 59

4.6 Top 5 retrieved solutions under different candidate sets for Deberta-V3-Base using the arithmetic solver . . . 60

4.7 Top 5 retrieved solutions under different candidate sets for the end-to-end arithmetic solver Flan-T5 model . . . 60

4.8 SATS test retrieval accuracies under different candidate sets, by solver type and model . . . 61

4.9 SATS test split generation metrics . . . 62

4.10 SATS test split copy rates 64
4.11 Few-shot exact match and summed copy rates for different Flan-T5 model sizes, per relation category 64
4.12 Flan-T5-XXL analogy examples before splitting by the answer separator 67
4.13 davinci-002 and ChatGPT-3.5 analogy examples 68
4.14 Arithmetic E2E-Flan-T5-Base analogy examples 69

A.1 Average Jaccard similarity between SATS pairs per relation 90

List of Figures

1.1 A simplistic graphical example of structure mapping between two concept networks 6
1.2 Bigger analogy test set retrieval accuracy per relation 16
1.3 Example analogical "attack-dispersion" procedural texts, originally from Gick and Holyoak (1980), and extracted analogical alignments from (Sultan and Shahaf, 2022) 27
1.4 Example of text-based abstraction of a progressive matrix problem 28

3.1 A visualization of the mean-pooled Flan-T5 autoencoder architecture 43
3.2 A visualization of the end-to-end solver and decoder architecture 46
3.3 Depiction of the sequence-to-sequence analogy task 50
3.4 Prompt used for few-shot solving of proportional analogies 51

4.1 Within-pair similarity versus Jaccard similarity, averaged per SATS test relation 57
4.2 Arithmetic solver retrieval accuracy with premises included versus the product of average within-pair similarity and OCS per SATS test relation 57
4.3 Exact match accuracy comparison of best models 63
4.4 Test exact match accuracy per model by parameter size, copy rates overlaid 65
4.5 Test METEOR (higher is better) per model by parameter size with METEOR of (a,b) pair baseline overlaid as a dash 66

A.1 Retrieval accuracy on SATS test split using the Abelian solver 91
A.2 Retrieval accuracy on SATS test split using the arithmetic solver 92
A.3 Retrieval accuracy on SATS test split using the feedforward solver 93
A.4 Retrieval accuracy on SATS test split using the mean premise solver 94

List of acronyms and abbreviations

VSM	Vector space model
FFNN	Feedforward neural network
INN	Invertible neural network
RNN	Recurrent neural network
LSTM	Long short-term memory network
GRU	Gated recurrent unit
NLL	Negative log-likelihood
KL divergence	Kullback-Leibler divergence
CBOW	Continuous bag-of-words
LLM	Large language model

MLM	Masked language modeling
3CosAdd	Three-term vector offset equation using cosine similarity to retrieve the solution to a proportional analogy
AGN	Abelian group network
BATS	Bigger analogy test set
NLI	Natural language inference
Vec2Seq	Vector-to-sequence decoding
DSBATS	Definition sentences from BATS
SATS	Our sentence analogy test set
QA	Question-answering task
QA2D	Question to declarative sentence dataset
GELU	Gaussian error linear unit activation function
PCS	Pairing consistency score

OCS	Offset concentration score
AUROC	Area under the receiver operating characteristic curve
TSDAE	Transformer sequential denoising autoencoder
WER	Word error rate

Acknowledgements

It's hard to overstate my gratitude for the support I've had over the course of my studies. First and foremost, I must thank my supervisor Philippe Langlais for his guidance and patience over the course of an initially fraught period working on this thesis. I also thank my friends and family, who offered me their unconditional love and encouragement during this time, for which I will forever be thankful.

Of my beloved peers with whom I've had the greatest pleasure of spending all too much time, for the great fun and banter we've had working together, hanging out, failing to grow tomatoes, and confined for the longest time, I thank inexhaustively and in no particular order Charlotte, Paloma, Rie, Magali, Ajanta, Alisha, Davide, Nicolas, Jeremie, Mariam, João, Lina, Jonny, Graham, Sarah, Emi, Ellie, Emily, as well as Bean and Olive, who as cats won't read this, and Oli, who I know will.

I thank my brothers Pasha and Sami for inspiring my interest in computers before I could even tie my own shoe laces, my mother Carline for her affection and faith in my success, and my father Antonio, whose unconditional love, wisdom, and unwavering energy dragged me kicking and screaming through my pursuit of a higher education. Without them, I could not have completed this thesis.

Finally, Yuan, my loving partner, who has had to observe this whole process, who has seen me through every step of its completion with love, guidance, and the many memories we've made together, and who is as excited as I am for its conclusion: thank you.

Introduction

It is today a dictum found in the abstracts and conclusions of many publications in the field of natural language processing, especially as concerns probabilistic models of language and vector space models of word meaning, that, as Firth (1957) wrote, you shall know a word by the company it keeps. One further adage is that analogy—vaguely, the comparative inference of an unknown implied by the similarity of two things—forms the core mechanism of human cognition (Hofstadter, 1995). In the intersection of these two maxims we find the main body of literature that inspires this work.

In short, this thesis targets 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 notated $a : b :: c : d$) formed on the basis of syntax, semantics, pragmatics, and encyclopedic knowledge. That is, we are interested in completing quadruples—given the first three terms—of paired natural language sentences 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, although ambiguous and lacking in rigour.

The general structure of this thesis is an overview of the literature relating to analogical reasoning, especially computational models thereof, followed by a description of works contemporary and directly related to our experiments, which we then present. We close with a discussion of our findings and general analysis in the context of the literature.

We will gloss over the ancient thoughts on reasoning and the medieval theological discourse which prove a storied use of the word "analogy" (Ashworth and D'Ettore, 2021) as well as some notable early contemporary works in the study of analogical reasoning in the fields of psychology and cognitive science (Spearman, 1927; Hadamard, 1945; Oppenheimer, 1956; Sternberg, 1977), where there was nascent interest in how humans apply their reasoning fluidly to novel or ambiguous circumstances to creatively find meaning.

Instead, our jumping-off point will be the more recent research, continuing into the present day, that has been done into computer models of analogical reasoning in various forms (Gick and Holyoak, 1980; Gentner, 1983; Falkenhainer et al., 1989; Holyoak and Thagard, 1989; Halford et al., 1993; Hofstadter, 1995; Turney and Littman, 2003; Mikolov et al., 2013c), though this thesis will remain focused on that which relates to natural language. We will refine our interest to models of distributional semantics, where a steadfast interest in the application of proportional analogy (also known as four-part analogies, those of the form “a is to b as c is to d”) using offsets from vector space models has persisted since the popularizing work of Mikolov et al. (2013c) on word embeddings, and has extended into related work on sentence embeddings, especially those enabled by large and pretrained language models of the recent Transformer architecture (Vaswani et al., 2017).

It’s in light of the latter more recent models that this work finds its space for contribution. As their capability to represent and apply meaningful human knowledge and cognitive biases becomes ever more possible (Wei et al., 2022), there is reason to believe that they may provide a suitable avenue to model human analogical reasoning (Wijesiriwardene et al., 2023; Webb et al., 2023), especially by leveraging their ability for creative generation (as opposed to e.g. retrieval) of natural language descriptions of various concepts.

Operating solely in a setting of proportional analogies between sentences, this work aims to gauge on the one hand the aptness of these more powerful language models for analogical reasoning via sequence generation, and on the other hand the aptness of common methods of analogical reasoning using vector spaces to recover useful features from these models. To this end our contributions are as follows:

(1) A dataset of pairs of sentences in varied relations from which we can construct four-part analogies.
(2) Evaluation of methods of analogy-solving with vector space representations, both via retrieval and via generation conditioned on a vector, using embeddings extracted from pretrained language models and other sentence embeddings.
(3) An exploration of finetuning pretrained language models for solving analogies in a sequence-to-sequence framework.
(4) The few-shot prompting of pretrained language models up into the billion-parameter regime for solving proportional analogies.

Chapter 1

Related work

1.1. Background

The word "αναλογια", analogy, originates in ancient Greece. With Euclid it came to denote an equality of two or more ratios, and later, with Plato, it was used in the comparison of things both concrete and abstract. The concept was later extended in the writings of Aristotle to many branches of study, for example in the comparison of the anatomy of different animals. In several of Aristotle's works, analogies—varyingly referred to in corresponding Greek as *paradigm*, *proportion*, *example*, *metaphor*, *induction*, or *analogy*, depending on the exact usage—describe several kinds of comparison-like operations, including the familiar form "A is to B as is C to D", denoted "proportional analogy". Brown (1989) distinguishes from this a so-described "predictive analogy", characterized by the inference of some properties in a second object—the "analandum"—based on similarities of some other properties with a first object—the "analans". In the absence of an inference, i.e. when two objects are simply found to correspond, he calls this "figurative analogy".

John Stuart Mill (1843) commented that "no word is used more loosely, or in a greater variety of senses, than 'analogy'." In addition to this quip, however, he concluded that successful analogical reasoning should result from situations where "(i) the resemblance is very great, (ii) the known difference very small, and (iii) our knowledge of the subject-matter fairly extensive".

Toward the end of the late modern period and with the advent of psychometrics following the First World War, the use of four-part proportional analogies as a tool to measure cognitive ability is noted by Spearman (1927) and other contemporaries. It's equally salient in its application in aptitude testing for college admissions.[1] Forgetting misgivings regarding the purposes and assumptions of such tests, much thought appears to have been given

[1] See for example the Miller Aptitude Test, which is coincidentally to be discontinued in November 2023, at time of writing.

at this time to the application of verbal analogies for measuring intelligence. For example, it is suggested as a matter of protocol in constructing verbal analogy tests that whatever relationship holds between terms A and B should equally hold between C and D,[2] though this advice was apparently not heeded by some, at the cost of corrupting what is described as a well-defined "relational thinking" exercise (Paterson, 1925; Levine, 1950). At this point analogy more generally is also noted for its role in advances in science (Hadamard, 1945; Oppenheimer, 1956) by providing structures of conceptual relations with which to think about novel circumstances.

With the advent of general-purpose computing and computational theory, coupled with greater understanding of neurophysiology and new hypotheses on learning and cognition (e.g. Hebb, 1949), came further interest in modeling human thought. We can distinguish this from the prior approach of general population statistics given in psychometrics, which attempted to distinguish individuals by ability and, to that end, find properties that could be measured in individuals in order to distinguish them. Enabled by computing machinery, a new goal revealed itself which was instead to mimic human ability artificially (Turing, 1950), and through doing so perhaps elucidate mechanisms of thought that provide the basis for human cognition. For example, the Argus model of thought (Reitman et al., 1964) is explicitly inspired by the Hebbian model whereby thinking emerges from association between individual units of processing, namely neurons. In Argus, semantic units similar to neurons mutually activate or inhibit each other in a weighted graph structure that models both firing and potentiating of connections. Notably, this model is applied to none other than the familiar proportional verbal analogy task that has become a mainstay.

Either way, it is with inspiration from computing systems and the conceptualization of human problem solving as an information processing system complete with control flow and symbolic logic (Newell et al., 1972) that consideration was now being given purposefully to defining analogical reasoning and frameworks of thought that allow it (Sternberg, 1977; Gick and Holyoak, 1980). Yet even with this novel conceptual basis, the narrowing of interest onto creating algorithms that fulfill individual human-like reasoning procedures led to the relative abandonment of developmental theories of learning and the link between mental representation and reasoning (Airenti, 2019).

In the characterization of analogy as a distinct procedure to enact on a given set of inputs as its own strategy for problem solving, Sternberg (1977), in the context of proportional analogies of the form $a : b :: c : d$, describes several control flow models which apply the following steps in varying sequences: (i) retrieving attributes of the analogy terms, (ii)

[2]This is still the case in common word analogy test sets such as those of Mikolov et al. (2013c) or Gladkova et al. (2016).

inferring the relation between the pair (a,b) by aligning common and different attributes, (iii) mapping, i.e. the alignment of common attributes and hence common “type” between terms a and c, and (iv) the application of the inferred relation to obtain d from c. Here all information is stored as attribute-value pairs.

Gick and Holyoak (1980) examine analogical reasoning in solutions to pairs of problems stated in prose. Here it is assumed that a representation, called *schema*, that abstracts relational structures of predicates that apply to terminal arguments (essentially a sort of graph) can be obtained for each problem, though they note that they “thus inherit all the problems associated with text comprehension”, which will hint at the present work’s particular lens, namely of applying rich representations extracted from text which allow for flexible analogical reasoning. They then study whether and how subjects apply the same steps described by Sternberg in order to solve the analogical problem.

In the structure-mapping model set out by Gentner (1983), analogy is described as independent of featural, i.e. surface similarity, giving the example of the comparison of an electric battery to a reservoir: the essential similarity is the storage of potential energy; even though both batteries and reservoirs may often be cylindrical in form, it is immaterial to the analogy. The core mechanism in this model of analogy is to assert “that a relational structure that normally applies in one domain can be applied in another domain.” The structure here, similar to Sternberg (1977) and Gick and Holyoak (1980), is a set of conceptual terminal nodes, as well as predicates and relations, such as in “FULL(CONTAINER, WATER)” (taken from Gentner (1983), meaning that a container is full of water), that together form a graph or equivalently a sort of related set of logical statements, which can be visualized as something akin to Figure 1.1.

According to this theory it might not be considered an analogy to state that, much as Rabbit A has a spotted pattern, long ears, and four legs, because Rabbit B has long ears and four legs it must also have a spotted pattern—an inference which, while faulty, is nevertheless based on similarity. This is due to the lack of a generic latent structure of relations, that is, one independent of the incidental concrete aspects in the description of Rabbit A that we can easily map to Rabbit B. In the structure-mapping model, mapping between analogical pairs can only occur between n-ary predicates with more than one argument. Hence, something like “SPOTTED(RABBIT A)” is ineligible for analogical mapping.

While the technical distinction is clear, we’ll venture that there remains ambiguity and arbitrariness in such handcrafted predicates as the one above, as well as in the differentiation of featural and structural similarity, which can be explained by the technical barriers to automatically obtaining a rich representation bottom-up. Let us also suggest that there lies

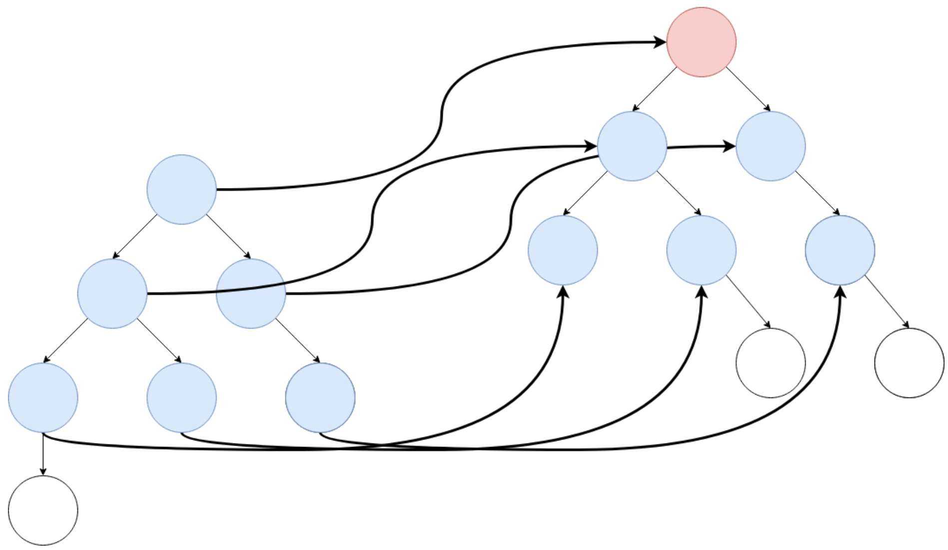

Figure 1.1. A simplistic graphical example of structure mapping between two concept networks

Nodes correspond to concept terms of any order (constants, predicates, relations), such as "atom", "revolves around", "temperature", "greater than", etc., for which thin arrows indicate relational arguments. Bold arrows between blue nodes indicate alignment. White nodes are non-alignable. The red node represents a term which can be inferred from aligned structures.

beneath this prescription the goal of modeling analogies of the kind that have been reported to enlighten scientific discoveries, those of which Oppenheimer (1956) said:

> I do not mean metaphor; I do not mean allegory; I do not even mean similarity; but I mean a special kind of similarity which is the similarity of structure, the similarity of form, a similarity of constellation between two sets of structures, two sets of particulars, that are manifestly very different but have structural parallels. It has to do with relation and interconnection.

It's very similarly that Gentner (1983) writes:

> To state a general law requires another step beyond creating a temporary correspondence between unlike domains: The person must create a new relational structure whose objects are so lacking in specific attributes that the structure can be applied across widely different domains.

At present, we believe the above prerequisite to modeling analogical reasoning is not inevitable and might be sidestepped with the recent impressive technical breakthroughs in representation learning and associated work. We should hope for a contextual representation which, with goal in mind, does away with irrelevant attributes when needed, and likewise keeps others when helpful, in order to elicit those more obscure and abstract properties which form a relational structure. Rather than asserting that analogical reasoning gives rise to relational abstraction, these new capabilities might allow us to verify the converse hypothesis: that a representation which discovers higher-order concepts (i.e. abstract relations) should allow analogical reasoning, in other words apply those same higher-order concepts as features in gauging how much and in what respects two things are similar.

Either way, symbolic models continued to be explored under various regimes (Holyoak and Thagard, 1989; Halford et al., 1993; Hofstadter et al., 1995) in the midst of discussion regarding the aptness of hand-coded high-level representations for constructing theories of human cognition, given the lack of robustness to varying contexts and of low-level information in building representations (Chalmers et al., 1992). While connectionist models were entering their second debut (Rumelhart and Mcclelland, 1986; Hinton, 1989), including in models of analogy running counter to symbolic techniques for the reasons stated above (Blank, 1997), it was advanced that what they couldn't offer in terms of reasoning with complex, abstract, highly structured representations, symbolic approaches could (Gentner and Markman, 1992). After all, it had yet to be demonstrated that there was a mechanism for connectionist models to represent and reason with high-level concept structures.

However, French (2002) suggested, in a review of computational models of analogy, that there were several developments still awaited to build upon the success of symbolic models. Among these, that there was a need for such models to apply context to the representations they act on, and that representations should be learned rather than built by experimenters. Keane (2013) questioned the empirical justifications for symbolic strategies and their computational time complexity. Separately, Thibodeau et al. (2013) defends that the question of whether connectionist models are capable of representing high-level or abstract structures was an unanswered empirical question. Counter to claims that this lack would prevent analogical reasoning, they argue that this capability should emerge "from the overlapping, distributed representations that are learned in the hidden layers" by learning conceptual representations through exposure to low-level features, without an explicit mapping mechanism.

There have been attempts to model purely symbolic proportional word analogies using information theoretic and formal language approaches. Lepage (2004) formalizes four-part analogies, obtaining postulates of equivalence between eight permutations of the terms $a : b :: c : d$. He characterizes a class of formal languages that can be constructed by recursive application of analogical derivation rules. In this vein, Miclet et al. (2008) present a sequence alignment-based analogical dissimilarity metric and algorithms to apply it to classification. Langlais et al. (2009) sample shufflings (i.e. the splicing of spans of strings) of analogical pairs (a,b) and select promising candidates using a Monte Carlo method to translate c to d. Murena et al. (2020) define a domain-specific language to encode operations on strings, and use this to simplify the computation of a minimal complexity program that maps a to b, which they then use to map from c to d. Naturally, in the interest of modeling cognition via human language, we note that these approaches do not readily extend to analogies that lack orthographic regularity such as *cat* : *kitten* :: *adult* : *baby*.

Concurrently, proportional analogy was approached in an effort to progress modeling natural language. Turney and Littman (2003) model word analogies using a vector space model (VSM) (Salton, 1989) obtained from the frequency statistics of collocations such as "X of Y" to embed the pair of words (X,Y), using a cosine similarity metric to retrieve the closest analogous pair among candidates. This kind of work is in keeping with the distributional hypothesis of word meaning (Firth, 1957). Mikolov et al. (2013c) fit a language model from which they extract an embedding, i.e. a collection of vectors which represent the words of a vocabulary. Using it as a VSM, they note that the vector space has significant linear regularities which permit them to retrieve the answer to a proportional word analogy using simple arithmetic. Stemming from this, a body of literature has since flourished regarding the use of proportional word analogies to evaluate vector representations, the pursuit of solving word analogies using VSMs in its own right, and the use of analogical proportions in representation learning. The questions of when linear regularities emerge, what is represented, how best to represent proportions in the vector space, which models offer the best representations, whether vectors provide a good avenue for analogical reasoning—all these questions have since been studied in various respects. They have equally since shifted from VSMs of word semantics to those of sentences and documents more generally, especially since the development of large language models (LLMs) stemming from the Transformer architecture (Vaswani et al., 2017).

In terms of analogy, there are three important opportunities offered by recent language models. First, we can model natural language descriptions of complex concepts for analogical reasoning, for example by extracting the vector representations they produce and operating on them. Second, we can leverage the generative ability of autoregressive language models to straightforwardly overcome a limitation of word embedding-based retrieval attempts at modeling analogy, namely the lack of creative inference we would hope from analogical problem solving. The third opportunity comes opposed to many approaches to date, which operate on an extracted representation in a piecewise fashion, potentially losing valuable information. Given that large language models have been shown to adapt to new tasks with no finetuning when provided natural language prompts given alongside their input (Brown et al., 2020), we can envision operating purely in the model's input domain, which it has been trained end-to-end to operate on and represent internally, and the representation of which may implicitly admit much more complex and numerous properties, relations, and latent structures than could be identified manually. That is, we might input a natural language description of an analogical problem which induces the language model to implicitly retrieve the relevant properties and perform predictive analogy, or otherwise discover a figurative analogy by latently discovered similarities.

We find this third line of research for analogy most promising, combining the desiderata of providing an empirically learned, contextually elicited representation (French, 2002) of complex relational structure (Gentner, 1983) emerging from distributed representations built from low-level features (Chalmers et al., 1992; Thibodeau et al., 2013). Indeed, work has already begun in this direction recently (Wijesiriwardene et al., 2023; Hu et al., 2023; Webb et al., 2023). However, our work only tangentially pursues this consideration of in-text reasoning, instead attempting to bridge its way there from the proportional analogy approaches which have been inherited from the word embedding evaluation methods of Mikolov et al. (2013c) and remain common at present even for sentences and text sequences more generally.

1.2. Probabilistic language models

Because later concepts may follow from this, we will begin by a rapid introduction of probabilistic language models.

Formally, a language model is a probability distribution over sequences in a language $L \subseteq V^*$ for some vocabulary or alphabet V. Among other things, we can call members of the vocabulary $w \in V$ tokens or words, depending on the context. These are basic units that can be composed into valid language, whether individual letters, whole words, or segments of strings that have been observed in a corpus. Measuring the probability of a sequence $s \in L$ of length l means measuring the joint probability of the entirety of its ordered tokens w_i, each of whose position i in the sequence is indexed from 1 to l:

$$P(s) = P(w_1, \cdots, w_l) \tag{1.2.1}$$

By the chain rule $P(x,y) = P(x \mid y)P(y)$, the joint probability is the product of the conditional probability and the marginal probability of the conditioner. In principle, this can be recursively applied in any order, but we will describe it in the autoregressive fashion:

$$P(w_1, \cdots, w_l) = \prod_{i=1}^{l} P(w_i \mid w_{1:l-1}) \tag{1.2.2}$$

A language model parameterized by θ is typically fit by maximizing the log-likelihood of some N observed sequences:

$$\mathcal{L}_\theta = \sum_{i=1}^{N} \sum_{j=1}^{l} log\left(\mathcal{L}_\theta(w_{i,j} \mid w_{i,1:j-1})\right) \tag{1.2.3}$$

though this is typically done by minimizing the negative log-likelihood (NLL). We can note that this objective is equivalent to maximizing the probability of the observations due to the monotonicity of the logarithmic function and by the fact that likelihood is an unnormalized probability score, which scales its values uniformly. Hence, the extrema are preserved. Other optimization schemes exist, such as masked language modeling (MLM), where only the conditional likelihoods of target tokens are used for the objective function.

Equivalently, if, when predicting a token, each option is given an output score l (called logit), token probabilities are often computed as the Softmax over the vocabulary:

$$\sigma(l_i) = \frac{e^{l_i}}{\sum_{j=1}^{N} e^{l_j}} \tag{1.2.4}$$

Then, removing the normalizing constant and taking the logarithm leaves us with the logits l_i as a token's log likelihood, and we can minimize a cross-entropy loss.

Many classes of model exist which have been applied to language modeling despite limiting architectural priors with respect to modeling sequences, such as n-gram models or hidden Markov models, which use simplifying assumptions of conditional independence outside of a context window to determine the probability of the next word in a sequence, or the continuous bag-of-words (CBOW) or Skip-gram architectures used by Mikolov et al. (2013a), which model words and their context independently of order. Other neural network architectures which integrate information over an arbitrary length sequence have been successfully applied to language modeling and machine translation, famously the various flavours of recurrent neural network (Rumelhart et al., 1985) such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). More recently, the easily parallelizable Transformer architecture (Vaswani et al., 2017) has spawned so-called large language models using billions of parameters which may have learned wide-ranging and robust representations of the world by maximizing the likelihood of observed natural language (Manning, 2022). These have been described as "foundation models" (Bommasani et al., 2022)—though the term applies to more than language models—since they purport to be general-purpose either in usage or in downstream application of their representations.

1.3. Analogy

As discussed in Section 1.1, there are two principal traditions of analogy. We will try to describe analogies using terminology drawn from Brown (1989).

In one tradition, a figurative analogy is the drawing of parallels from one object, the analans which we might call A, to another object B, the analandum. We will call it a predictive analogy if, given the two objects A and B which share some number of properties in common, we infer an additional shared property:

$$(P_1(A) \wedge P_1(B)) \wedge \cdots \wedge (P_n(A) \wedge P_n(B)) \wedge P_{n+1}(A) \implies P_{n+1}(B) \ .$$

The choice of which properties should form the premise of this analogical inference and how they should be represented varies by model, as we have seen. In symbolic approaches they may be found by performing some kind of alignment on sequences (Miclet et al., 2008) or graphs (Gentner, 1983). In connectionist models, a local (Reitman et al., 1964) or distributed representation may activate similarly for two inputs, whether relevant features are represented explicitly (Halford et al., 1993) or implicitly (Mikolov et al., 2013c; Thibodeau et al., 2013). Either way, it is apparent that analogy in the general case requires inferring common features on the basis of the alignment of something more than just surface features (Gentner and Markman, 1992).

There is then a separate tradition of what we will call proportional analogies, with which we are chiefly concerned in the present work. This framework is drawn from classical numeric additive or geometric analogies, that is, the arithmetic ones like $5 - 3 = 12 - 10$, or those of ratios as in $\frac{2}{1} = \frac{4}{2}$, in addition to conceptual ones, for example when Aristotle (from Barbot et al., 2019) writes "as old age is to life, so is evening to day". These analogies are a quaternary, typically homogeneous relation where for some $(a,b,c,d) \in X^4$ we notate it $a : b :: c : d$ (read "a is to b as c is to d") if and only if some relation $R \subseteq X^2$ holds both for (a,b) and (c,d):

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

We can appreciate with some reframing that this four-part analogy paradigm can be seen as a special case of figurative analogy operating on pairs (a,b) and (c,d). Here, predictive analogy consists of concluding d from the premise (a,b,c) by relying on properties that are shared between the analans and analandum. That is, we expect our representation of elements in X coupled with an inference system to reflect relevant properties and discover a relation that holds between two pairs when finding d.

In many proportional analogy settings, if such a quadruple satisfies these conditions, we may expect the following statements to hold (Afantenos et al., 2022):

(1) $a : b :: a : b$ (reflexivity)

(2) $a : b :: c : d \implies c : d :: a : b$ (symmetry)

(3) $a : b :: c : d \implies a : c :: b : d$ (central permutation)

We can further conclude from symmetry and central permutation the equivalence of eight forms:

$$a : b :: c : d \equiv c : d :: a : b \equiv c : a :: d : b \equiv d : b :: c : a$$
$$\equiv d : c :: b : a \equiv b : a : d : c \equiv b : d :: a : c \equiv a : c :: b : d \ .$$

Reflexivity and symmetry are obtained straightforwardly from the definition we gave above, by the identity of a pair and by the symmetry of the logical conjunction. Lepage (2004) describes Aristotle taking central permutation as an axiom resulting from common sense, and likewise constructs a theory of analogy in a formal language setting that fulfills this postulate. Prade and Richard (2017) as well define analogy built on formal logic using Boolean vectors so as to satisfy these postulates.

As argued by Afantenos et al. (2022), however, in proportional analogies seen in natural language processing or computational linguistics, one often expects a relation $R \in S$ from some set family $S \in \mathcal{P}(X^2)$ of relations considered admissible in some way. In such a case, given $a : b :: c : d$, we may not have an admissible relation satisfying our definition of analogy for $a : c :: b : d$. Instead of the central permutation postulate, they and Lim et al. (2021) advance the use of an internal reversal postulate $a : b :: c : d \implies b : a :: d : c$ for word and sentence analogies. However, we note that if we do not admit R^{-1} for all relations R in our set, this postulate does not hold for our definition.

Let us observe arithmetic analogy, where we consider analogies, formed by a class of relations $\{R_x \mid aR_xb \iff b - a = x\}$, for which the central permutation and internal reversal postulates both hold. In the instance $3 : 5 :: 6 : 8$, it is the relation for "$x = 2$" which is analogical between the pairs (3,5) and (6,8). Only incidentally does a new analogy of "$x = 3$" hold when reordered as $3 : 6 :: 5 : 8$, due to the algebraic properties we have on the integers under addition such as closure, associativity, and the existence of an inverse. Yet, it is not clear what relations we would consider valid so as to change $old\ age : life :: evening : day$ into its permutation $old\ age : evening :: life : day$.

For the reasons alluded to above, beyond the particularities of some domains, we might accept that the permutation of terms in proportional analogies is a matter of convention (Lim et al., 2021). We will keep our definition of proportional analogy as shown above, assuming only reflexivity and symmetry, and without imposing any assumptions of invertibility on our relations.

1.4. Word analogy in vector space models

A sizeable literature exists on methods for solving proportional word analogies. These typically use distributed representations for words which are usually obtained from the parameters of language models or from some other distributional statistic such as co-occurrence. We introduce these concepts for the reason that similar methods and limitations apply to their extension to sentence analogies.

1.4.1. Word embedding

As seen in Section 1.2, strings in a language consist of an ordered sequence of discrete elements, tokens. The number of possible strings of a given length l grows on the order of $\|V\|^l$. This exponential growth means that, while we may observe a given token often, we may almost never see sufficiently many combinations of tokens in order to generalize from our observations to the true distribution of language. This issue, referred to as the "curse of dimensionality", can be circumvented if we instead take each token as a dense vector in a much lower dimensional space than the size of the vocabulary. By leveraging this distributed representation, tokens which do not co-occur in observations can inform each other indirectly via shared context words, or by the similarity of their contexts, and we can model the probability of a sentence with a combination of words never before observed together (Bengio et al., 2003).

1.4.2. Vector offset method

This is also typically referred to as the 3CosAdd method (Levy and Goldberg, 2014). We might additionally refer to it as the vector arithmetic method, while qualifying any particularities.

It was advanced in the work of Mikolov et al. (2013a,c) that the linear regularities present in the vector representations obtained by their methods encoded linguistic information allowing to solve word analogies using simple vector arithmetic. With the relation between a pair of words represented by the offset—that is, the subtraction of one word embedding from another—they apply that relation to another word to solve proportional analogies of the form $a : b :: c : d$ simply by adding the offset $b - a$ to a third word embedding c, taking all vectors to be normalized. Additionally, taking the vocabulary embedding as a VSM, they retrieve the closest vocabulary item in terms of cosine similarity as the solution to the analogy. We present this in the following equation:

$$\hat{d} = \operatorname*{argmax}_{w \in V} \frac{w^\intercal (c + b - a)}{\|w\| \cdot \|c + b - a\|} \tag{1.4.1}$$

More generally, we may use:

$$\hat{d} = \operatorname*{argmax}_{w \in V} \; sim(w, x) \tag{1.4.2}$$

with sim typically being the cosine similarity and $x = c + b - a$, in which case we might call $sim(d,x)$ the analogy score:

$$sim(d,\, c + b - a) = \frac{d^\intercal (c + b - a)}{\|d\| \cdot \|c + b - a\|} \tag{1.4.3}$$

Equation 1.4.2 is simply a retrieval operation in a VSM, replacing V for any set of document representations. Note that in many works, a, b, and c are removed from the candidate set V (Mikolov et al., 2013a,c; Levy and Goldberg, 2014; Gladkova et al., 2016). This is discussed further in Sections 1.4.4 to 1.4.6, and will recur in later chapters. The metric typically used for this task is the accuracy, which is simply the percentage of analogies where the predicted vector—above, $c + b - a$—retrieves as a nearest neighbour the correct answer word in a VSM defined by the vocabulary embedding being used, i.e. the set of vectors each corresponding to a word.

1.4.3. Word analogy datasets

The proportional analogies originally tackled by the methods described in Section 1.4.2, referred to as the Google analogy test set[3] (Mikolov et al., 2013a) and the Microsoft Research syntactic analogies dataset (shortened to MSR, Mikolov et al., 2013c), are drawn from syntactic and semantic relations. MSR contains 8,000 analogies based on word morphology, depending on the part of speech. These include comparative and superlative adjectives (as in $large : larger :: tall : taller$ or $slim : slimmest :: rough : roughest$), pluralization and possessive forms of nouns (e.g. $mouse : mice :: dot : dots$ or $bird : bird's :: they : theirs$), and verb conjugations (such as $tap : tapped :: throw : threw$). The Google dataset has 19,544 analogies from 9 morphological and 5 semantic relation types. Of the former category, there are relations of inflection, namely comparative and superlative adjectives, noun pluralization, and past tense, present participle, and present tense third person conjugation of verbs, as well as morphological derivations[4] of adjectives to adverbs ($amazing : amazingly :: obvious : obviously$) and pairs of opposites ($possible : impossible :: known : unknown$), and finally pairs of countries and demonyms ($India : Indian :: Poland : Polish$). Of the semantic category, there are three relations akin to pairs of cities and geographic entities (cities and their US state such as $Chicago : Illinois :: Stockton : California$, some common capital cities such as $Cairo : Egypt :: Athens : Greece$, and capital cities from around the word, such as $Georgetown : Guyana :: Madrid : Spain$), supplemented by one relation of country

[3] Available at http://download.tensorflow.org/data/questions-words.txt.

[4] Morphological derivation is opposed to inflection by the former's change of the meaning of a word (e.g. **un**appreciated or defend**er**), versus the latter's change of grammatical category, such as number or person (e.g. I eat, the bird**s** eat, it eat**s**), though this dichotomy can be blurry (see for example Tuggy, 1985, for such a discussion). Morphological derivation can result from a regular transformation by addition of a morpheme, with the resulting meaning being clear from the composition of its parts (i.e. are *productive*), but it is not always so. Some derivations are unproductive, not readily parsable, or dependent on phonological constraints or a root word's original language family. See e.g. warm**th** but not small**th**, lev**ity** but not light**ity**, typ**ist** but not writ**ist**.

and currency (e.g. $Algeria : dinar :: Mexico : peso$), and one of gender opposition of nouns (e.g. *uncle* : *aunt* :: *prince* : *princess*).

Gladkova et al. (2016) note the imbalance of this dataset across different types of relations, and the high variation in accuracy across different relations found by Levy and Goldberg (2014), going from 10.53% to 99.41%. They instead propose a dataset, which they call the Bigger analogy test set (BATS), of 40 varied relations each holding 50 word pairs, constructing analogies by matching pairs of words within a given relation. We should clarify, as it will recur later in this work, that analogy datasets can be constructed from sets of more than one *pair*, as long as each pair in such a set (a_i,b_i) obeys a shared relation R (see Section 1.3). In such a case, we can shuffle pairs of a same relation set against each other to obtain an analogical quadruple $a_i : b_i :: a_j :: b_j$, i.e. such that $a_i R b_i \wedge a_j R b_j$.

BATS is divided into four broad categories of 10 relations each (which are individually displayed in Figure 1.2). These are inflectional (similar to those of MSR or Google), derivational (where a word's meaning shifts by the addition of a morpheme, e.g. *effort* : *effortless* :: *law* : *lawless*), lexicographic (relations of words, concepts, taxonomy, e.g. *cake* : *dessert* :: *dress* : *clothes*), and encyclopedic (instances of individuals and facts about the world, e.g. capital cities, people, e.g. *Beethoven* : *composer* :: *Euler* : *mathematician*, sensory features, e.g. *broccoli* : *green* :: *carrot* : *orange* or *bear* : *growl* :: *crow* : *caw*, and associated concepts, e.g. *bee* : *hive* :: *duck* : *pond*). In this reevaluation, they note that outside of inflectional relations and some encyclopedic relations such as those of cities to territories, the accuracy of the test using 3CosAdd falls sharply. They conclude that analogical reasoning with the offset of word embeddings only handles certain relations due to others either being unrepresented by the embedding or irretrievable via the offset. This is illustrated in Figure 1.2.

1.4.4. Offset method caveats

It is noted that when the inputs a, b, and c are kept in the candidate set, accuracy on the analogy test often drops to nil (Drozd et al., 2016). Linzen (2016) finds that when not excluding the inputs, the nearest neighbour to $c + b - a$ in 93% of cases is c, and that the nearest neighbour of c was a relatively effective baseline when compared with 3CosAdd when excluding inputs. Schluter (2018) finds that when 3CosAdd wrongly predicts an input vector, it is c 99% of the time. All the above are once again reasons to discard results which exclude inputs from the candidate set.

While it is implied that the magnitude of the offset is too small to move away from c (Rogers et al., 2017; Schluter, 2018), Fournier et al. (2020) instead find that the offsets of both pairs are of similar magnitudes: if it is large enough to move a to b, it should be large enough to move c to d. By studying a decomposition of the analogy score (see Equation 1.4.3)

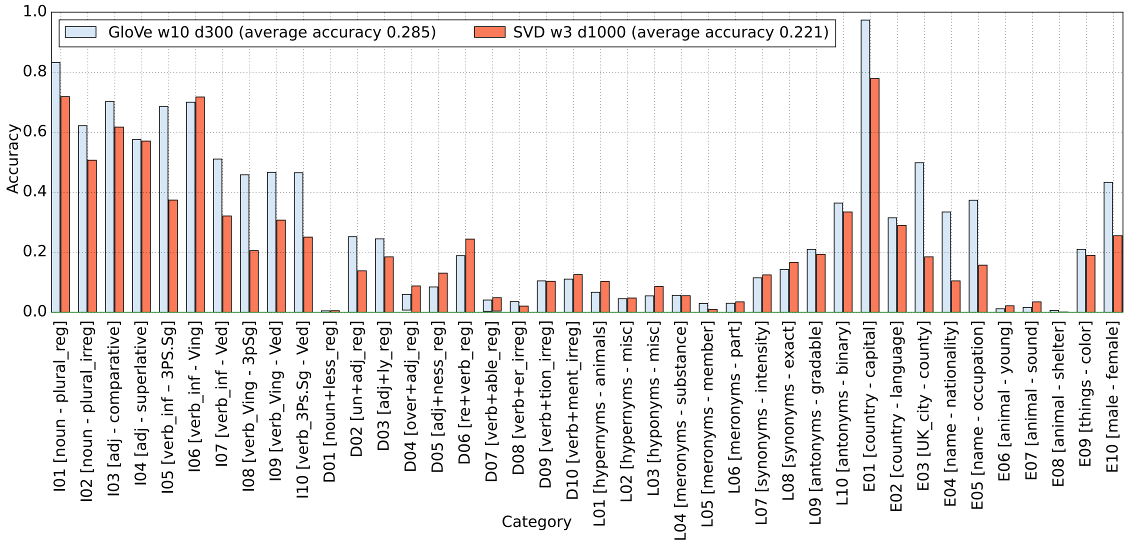

Figure 1.2. Bigger analogy test set retrieval accuracy per relation
Figure taken from Gladkova et al. (2016). GloVe refers to the word embeddings of Pennington et al. (2014). SVD refers to embeddings given by the singular value decomposition of a window cooccurrence count matrix.

subtracted by the similarity of the prediction to the input c, they find that the similarity of both pairs' offsets must be greater than the negative cosine distance of the terms of the second pair (c,d) in order to recover d. Oftentimes the offset in fact moves c away from every other candidate. They find that the offset method is inconclusive as a test for the presence of linear regularities in vector space models.

The assumptions of reasoning linearly in a vector space this way have also been criticized for lack of plausibility as a model of human reasoning by Rogers et al. (2017). It is considered insufficient to model semantics due to the binary oppositions of semantic attributes that it assumes along dimensions of regularity, and for the properties of addition. The offset model equivalently represents permutations of an analogy and permits offsets which imply the existence of relations of a dubious quality. We'll paraphrase their example $(remarry - marry) + write = rewrite$, which is intuitive, contrasted with the equivalent left-hand side $(write - marry) + remarry$, which is evidently nonsensical. They further note that, whereas distance in a VSM has symmetry, human similarity judgements do not. They conclude that such equations should "only be interpreted distributionally", rather than offering some regularity regarding semantics..

1.4.5. Linear regularities

With a pairing consistency score (PCS), Fournier et al. (2020) measure the linear separability of the distributions of true offsets (from BATS) and shuffled offsets (false pairs from within a relation set). They find with multiple word embeddings that even when for some relations the analogy test has a low score under both "honest" (where inputs remain candidates) and "dishonest" conditions, PCS remains above chance level, i.e. true offsets are more parallel than false ones between pairs of vectors from a given BATS relation. Hence, the vector spaces studied do in fact have globally linear regularities which are strong in the inflectional and derivational morphology categories, but very weak for the encyclopedic and lexicographic ones. They argue that the analogy score decomposes into terms which are affected by spurious properties of the embedding geometry, terms respectively proportional to the similarity of the offsets, the similarity of (c,d), and the similarity of c and $b - a$.

Some mathematical justifications have been advanced for when and how such linear regularities can emerge (Gittens et al., 2017; Allen and Hospedales, 2019; Ethayarajh et al., 2019; Ri et al., 2023). Gittens et al. (2017) reframe proportional word analogy in terms of compositionality and paraphrase. For additive composition, paraphrase is achieved ideally if a set of context words' embeddings $C = c_1 \cdots c_n$ defines a conditional probability distribution $P(x|C) = P(x|c)$ where $c = \sum_i c_i$ the sum of those context words' vectors. That is, they define paraphrase as a set of context words together having the same conditional distribution over the vocabulary as some other word (or, in practice, that the best candidate for such a composition minimizes the Kullback-Leibler (KL) divergence of its conditional distribution with that of the composed context words).

We will quickly clarify that the KL divergence is akin to a distance between probability distributions (though it is not a metric, like Euclidean distance, as it is asymmetric and does not satisfy the triangle inequality). Assume there was a target distribution p over a domain X which we were modeling with a distribution q. While we can compute the entropy (Shannon, 1948), i.e. the expected self-information $\mathrm{E}_p[-\log p(x)]$, the number of bits[5] needed to encode a value sampled from p itself (thus the optimal such number of bits), we may be more interested in knowing how bad of a job our model q is doing (in other words a *loss* value). In this case we can subtract the self-information from the cross-entropy $\mathrm{E}_p[-\log q(x)]$, the expected number of bits we would use to encode a sample drawn from p if we instead assume it is drawn from q. Doing so computes the KL divergence, whose equation we will write as follows:

$$D_{\mathrm{KL}}(p\|q) = \mathrm{E}_p[\log p(x) - \log q(x)] \tag{1.4.4}$$

Importantly, Gittens et al. (2017) prove that when word frequency is uniform, such composition is additive (and otherwise in general must be the solution to non-linear systems

[5]Bits if we take the logarithm in base 2, "nats" if we take the natural logarithm, though any base can be used.

of equations). However, language empirically follows a Zipfian distribution (see Piantadosi, 2014, for a discussion on Zipfian distributions), which devalues this assumption. They demonstrate that Skip-gram can approximate an information-theoretically optimal representation of co-occurrence features, which may explain that window co-occurrence statistic-based models such as those of Skip-gram embeddings (Mikolov et al., 2013b) best represent analogies of certain relations more than others, presumably when the terms of an analogy are capable of occurring in similar context windows, which are relatively small. More recently, Ri et al. (2023) similarly establish a relation between co-occurrence statistics and linear regularities in terms of parallel offsets which emerge from the push-pull dynamics of a contrastive loss function, which we'll note are used not only in word embedding models like Skip-gram, but also in sentence embedding models (Giorgi et al., 2020; Kim et al., 2021; Gao et al., 2022) where document paraphrases or augmented data play the role of a word's context and positive or negative examples. We might hypothesize that, as concerns solving sentence analogies using sentence embeddings—discussed in Section 1.5—the arithmetic analogy test may be more successful when the sentences "co-occur" as captured by the positive and negative training examples of a given embedding model.

1.4.6. Other solving methods

Other vector solving methods have previously been explored, some of which we'll quickly present.

Pitis (2016) explores using rotations (more generally orthogonal matrices) rather than offsets. That is, one can find such a matrix such that $b = Ma$ to recover d by setting $x = Mc$ in Equation 1.4.2. By evaluating an orthogonal matrix rotating a to b, as well as one between the midpoint of (a,b) and approximation to that of (c,d),[6] this method is found to perform comparably to 3CosAdd. Ethayarajh (2019) does not aim to provide a method for solving analogies, but finds that closed form solutions for matrices, whether linear (by least-squares) or orthogonal (by a closed form solution given by Schönemann, 1966) can capture the relations given in the Google analogy test set of (Mikolov et al., 2013a), achieving accuracies over 75% averaged across relations. However, it appears that the pairs used to solve for such matrices are sampled across the whole dataset without use of a split; we cannot conclude that this method generalizes.

To obtain a higher accuracy, Drozd et al. (2016) propose using a held-out set of n pairs for evaluation, using the rest to obtain average offsets from a same relation (the offset is then used in typical vector arithmetic), as well as a method they call LRCos (as in *logistic regression* and *cosine similarity*). In this latter method VSM retrieval candidates for $\hat{d}$ are

[6]The second midpoint is approximated since d is unknown.

scored by the probability they belong to the target class (that of b) multiplied by the cosine similarity with a. They augment this by learning to scale relevant features and down-weight irrelevant ones for similarity within a given relation. With this they argue that even though parallel offsets don't recover relations, they are still encoded by the embeddings. For every analogy, they reportedly hold out $n = 2$ pairs per relation—presumably (a,b) and (c,d)—for testing, using the rest to find the "rule".

It has been proposed that invertible neural networks (INN) allow an Abelian group network (AGN) to model an operation which can be substituted for the usual additive group. Abe et al. (2021) provide a proof that such a model is a universal approximator of Abelian Lie group operations, i.e. those with a commutative group operation and inverse function which are both differentiable. We find this a hopeful avenue to explore insofar as proportional analogy can be captured by an Abelian Lie operation, and such an operation learned from data exemplifying analogies we aim to solve. However, we do not see it alleviating criticisms in Section 1.4.4 regarding the additive operation.

Instead of setting $x = c + b - a$ in 1.4.2, they use

$$x = \phi^{-1}\left(\phi(c) + \phi(b) - \phi(a)\right) \tag{1.4.5}$$

where the function ϕ is parameterized as an INN. While the Abelian group network is more generally a binary operation, repeated applications simplify to Equation (1.4.5).

They fit the parameters of this model on a training split balanced across BATS relation categories, where they sample according to a split ratio for each category. That is, there are held-out pairs rather than held-out relations. They find that the AGN has a marked, usually double-digit improvement in most categories over 3CosAdd in the "honest" analogy test where the inputs are not excluded. When excluding inputs, it is still competitive, but sometimes outperformed by 3CosAdd. We will comment that we expect this behaviour in that the offset method typically leads further from any other word than the input c (Fournier et al., 2020), as discussed in Section 1.4.4. Given the sparsity of candidates in the retrieval regime and the lack of distractors among them, it is to be expected that the answer, which is typically a close neighbour of c, should be chosen when the premises are excluded. Further, the Abelian network needs to learn to model the space, which is difficult given the sparsity of training data they use from BATS.[7]

[7]Being built from 40 sets of 50 word pairs, BATS may only have on the order of 4,000 unique words.

1.5. Sequence analogy

As we have said, the present work seeks to take popular methods developed for proportional word analogies and apply them to their natural extension, sentences. Of course, there have already been attempts in this direction, so we shall summarize them here.

1.5.1. Sentence embedding

We introduced word embeddings in Section 1.4.1, where words are represented in a vector space that is of much lower dimension than the discrete space of observations which grows exponentially in sequence length. We can extend this line of reasoning from individual tokens to the whole observed sequence. This can be achieved for example by modeling the sequence as entirely conditional on a latent vector, by optimizing some other objective which results in the aggregation of sentence-wide information into a vector, or even simply by averaging the individual embeddings of all tokens in a sequence.

For example, Logeswaran and Lee (2018) use a contrastive objective to distinguish true neighbouring sentences from false ones, where the representations are vectors drawn from the hidden state of an RNN. They evaluate their sentence embeddings on a proportional analogy task drawn from the Google and Microsoft word analogy sets Mikolov et al. (2013a,c) formed of pairs of sentences from the Yelp review dataset which differ from each other by (approximately) the target word pair (Guu et al., 2018).

The bidirectional encoder representation transformer (BERT) model (Devlin et al., 2019) is pretrained using a masked language modeling objective, where only a portion of tokens are hidden as a special mask token ([MASK]) and are predicted from the rest of the sequence. This is coupled with a next sentence prediction objective, in which two subsequences are concatenated by a special separation token ([SEP]), and the model must classify whether the second subsequence follows the first in the data. This classification uses the final representation of a special classification token ([CLS]) which is prepended to all input. Sentence-BERT (Reimers and Gurevych, 2019) is a finetuning method where the parameters are fit using a classification or contrastive objective relying directly on sentence embeddings pooled from the final encodings of the tokens (e.g. by taking the maximum, the mean, or the [CLS] token). Instead of explicitly fitting a model for semantic similarity tasks to execute a forward pass which computes a similarity in [0,1] on a concatenated pair of sequences, one can use a VSM of these embeddings and take the cosine similarity of pairs of embeddings. This reduces the computational hurdle, as any given sequence need only be encoded once, rather than a forward pass done for every desired pair. Further work has since gone on to examine learning sentence embeddings using contrastive objectives (Giorgi et al., 2020; Kim et al., 2021; Gao et al., 2022).

Much as there was for word embeddings, there is for sentence embeddings a search to uncover what is encoded in their vectors, what linguistic regularities can be found in the learned embedding geometry, and how to learn richer features in these embeddings so that they can be fruitfully applied to downstream tasks (Conneau et al., 2018; Li et al., 2020; Su et al., 2021; Huang et al., 2021; Muennighoff et al., 2023). Similarly again, it has led to the use of proportional analogies for learning as well as evaluation.

1.5.2. Retrieval using vector offset

Zhu and de Melo (2020) use proportional analogies of sentences to evaluate what linguistic patterns are represented in various sentence embeddings. They create a dataset of sentence analogies by constructing many pairs of sentences in any of 12 fixed relations, leveraging templates, structural parses of sentences, and sentences from natural language inference (NLI) datasets. We show the number of sentence pairs for each relation and a representative example pair in Table 1.1. They create 5 "syntactic" relation sets, each corresponding to one of a subset of the lexical analogies of the Google analogy dataset (Mikolov et al., 2013a), by parsing sentences for the presence of substitutable words and substrings in order to replace adjectives with their opposite, pluralize noun phrases with numerals, modify verb conjugation, shorten sentences with comparative adjectives, and replace demonymic adjectives with preposition phrases of country names. They additionally create 5 "semantic" relation sets corresponding to the 5 semantic relations in the Google dataset using template sentences. These templates allow to replace target words in the template with corresponding words from the Google dataset pair. From NLI datasets, they extract pairs of sentences where entailment or negation holds. Ultimately, this data is not publicly accessible, though we present the examples shown by the authors.

For evaluation, they use the familiar 3CosAdd method to solve these analogies, as well as one called 3CosMul (Levy and Goldberg, 2014) where terms' similarities are multiplied rather than an offset used. Crucially, they evaluate with and without the easing constraint of excluding the premise vectors (a,b,c) from the candidate set, a practice we discussed in Section 1.4.4. They note that sentence embeddings composed from fixed word embeddings via averaging or discrete cosine transform (Almarwani and Diab, 2021) outperform those taken from BERT, RoBERTa (Liu et al., 2019), and Sentence-BERT, though the gap narrows in the constrained case. Their results also show that 3CosMul performs identically when premises are included in the candidates, and substantially worse when they are not, depending on the embedding.

It is to note that many of their pairs of sentences differ in limited ways and likely have considerable word overlap, being based on lexical analogies from the Google and Microsoft sets

Category	# Pairs	Example
Common capital cities	138	I'm not sure if they can travel to **Havana**. : I'm not sure if they can travel to **Cuba**.
All capital cities	928	I've never been to **Amman**. : I've never been to **Jordan**.
City in state	402	They are going down to **Chandler** when they get cold. : They are going down to **Arizona** when they get cold.
Currency	150	The economy in **Japan** was and always will be great. : The **Japanese yen** appreciated due to the strong economic performance of the country.
Male-female	126	The **man** makes wooden crafts and arts. : The **woman** makes wooden crafts and arts.
Comparative adj.	466	The second article was **long**. : The second article was **longer** than the first article.
Nationality adj.	513	The man from **Egypt** tapped his cheek. : The **Egyptian** man tapped his cheek.
Opposite	205	It's **possible** to measure it. : It's **impossible** to measure it.
Plural	512	The Harvard data examined one **city** on the East coast. : The Harvard data examined 6 **cities** on the East coast.
Verb conjugation	451	Duke will **play** better this year. : Duke **plays** better this year.
Entailment	673	The turtle is tracking the fish. : The turtle is following the fish.
Negation	511	There is no skilled person riding a bicycle on one wheel. : A skilled person is riding a bicycle on one wheel.

Table 1.1. Sentence pair count and examples per relation for the analogy dataset of Zhu and de Melo (2020)

Edited from the original authors' table. Words corresponding to lexical analogies from the Google dataset are bolded.

whose imbalance we know favorably represents word embeddings (Gladkova et al., 2016). Further, their data contains overall about 10,000 unique sentences which might be easily differentiated between, and they use no distractors except in the NLI analogies, where distractors are generated for each true answer and form the only alternative candidates, and for which they do not report the unconstrained accuracy. These concerns parallel those we have discussed in Section 1.4 about word analogy tests: retrieval from a vector offset may mostly measure clustering ability.

1.5.3. Decoding from an embedding

Instead of retrieving the answer to an analogy, Wang and Lepage (2020) fit an LSTM language model which decodes a sentence either from the sum of its word vectors or from Sentence-BERT. We may refer to this as vector-to-sequence, or "Vec2Seq". They find that their model successfully decodes summed fixed word vectors, though fails at decoding Sentence-BERT representations. They experiment with solving for vectors with $d = f(a,b,c)$, substituting for f the vector offset operation $c + b - a$ and a regression model parameterized as a feedforward neural network (FFNN) which takes as input a combination of the premise vectors among concatenation, summation, and the vector offset prediction. The solver network is trained piecewise on an 80% split of a sentence analogy dataset.

They use a set of 5,607 semantico-formal analogies[8] obtained by Lepage (2019) from the Tatoeba corpus by use of an alignment algorithm and pretrained FastText embeddings (Grave et al., 2018) to provide a distance between sequences' tokens. These are analogies of unspecified relation, though they usually involve word substitutions, inflection, negation, or an unusual change in meaning as in this example drawn from the set: *I do not need a wheelchair.* : *I do not need a girlfriend.* :: *I do not have a cat.* : *I do not have a boyfriend.*

Ultimately, they find that decoding from the vector offset solution performs worst, with half the exact-match accuracy of the solver trained on the offset prediction, which performs best in all metrics they test for, including edit distance, Jaccard similarity, BLEU (Papineni et al., 2002), and METEOR (Banerjee and Lavie, 2005). The concatenation representation is a close second, though noticeably worse. We venture the idea that there may be substantial overlap in examples seen in training and at test time, which an 80-10-10 split may not overcome, so it is unclear whether the solver is learning the operation of analogy or has simply learned to represent valid embeddings well. That being said, the reasons for better performance remain unclear.

[8] Available at http://lepage-lab.ips.waseda.ac.jp/projects/kakenhi-kiban-c-18k11447/ under Experimental Results.

Mao and Lepage (2023) take on a similar decoding framework to solve the analogies of Lepage (2019), though they fit an "offset network" composed of two neural networks: one extracts a vector representation $r = f(a,b)$ of the ratio $a : b$, and a second $\hat{d} = g(c,r)$ maps from c and the ratio r to the solution d. The decoder and solver are still trained piecewise. Comparing representations from summed word embeddings and one from an LSTM autoencoder they pretrain, they evaluate the vector solver of Wang and Lepage (2020) and their offset network, which they find notably underperforms the former model for both representations on the semantico-formal set.

1.5.4. Classification

A binary classification can be done on a quadruple (a,b,c,d) where the positive class represents that $a : b :: c : d$ is true, and the negative class means it is not. Taking advantage of surface regularity, Alsaidi et al. (2021) fit a classifier on word embeddings obtained from a convolution over a sequence of vector representations of individual characters. They tackle a total of millions of analogies constructed from pairs of words and their inflections, such as *do* : *doing*, across 10 different languages. For analogies obtained from morphological changes of words, the formal nature of the analogy can be leveraged to apply various axioms, postulates, and deduced properties of such analogies (Lepage, 2004), allowing to augment a dataset with valid and invalid permutations. However, for analogies of natural language sentences, if we permit them to stand in more than formal relation to each other, most valid permutations of a proportional analogy fall away (Afantenos et al., 2022), as discussed in Section 1.3.

Barbero and Afantenos (2023) have attempted to classify, using embeddings as features, sentence analogies constructed from sentence pairs between which NLI, paraphrase, and Penn Discourse Treebank (PDTB) relations (Prasad, Rashmi et al., 2019) hold. We have seen examples of NLI sentences (entailment and negation) used for analogies in Section 1.5.2. Paraphrase can be seen as a many-to-many relation between sentences, which could in principle be refined to more granular kinds of paraphrase based on syntactic or stylistic changes. As it is, however, one should not expect to unambiguously solve an analogy such as *Several cats are climbing up that tree* : *A bunch of animals are clambering up this oak* :: *Are you the one who broke my cup?* : *Could you be the one guilty of shattering my glass?* without knowledge that the relation in question is loosely that of paraphrase.

The sentence pairs they obtain from discourse relations are described as binned into the top level of the PDTB annotation hierarchy. Without delving too far into the details of this corpus, these relations would correspond to the Level-1 senses (Prasad et al., 2018) labeled as follows:[9]

(1) Temporal (e.g. *Small businesses say a recent trend is like a dream come true: more-affordable rates for employee-health insurance, initially at least.* ***But then*** *they wake up to a nightmare.*)

(2) Contingency (e.g. *I cannot recall any disorder in currency markets* ***since*** *the 1974 guidelines were adopted.*)

(3) Comparison (e.g. *The Soviets insisted that aircraft be brought into the talks,* ***(but)*** *then argued for exempting some 4,000 Russian planes because they are solely defensive.*)

(4) Expansion (e.g. *Not only do the actors stand outside their characters and make it clear they are at odds with them,* ***but*** *they often literally stand on their heads.*)

One can easily imagine dividing each example by the bolded text to obtain a pair, though it is not clear that they would make felicitous analogies.

They do not use any assumptions of parallel offsets or linear regularity, instead fitting both a FFNN on pooled token embeddings and a CNN which convolves over the dimensions of the four pooled vectors (a,b,c,d). Using both fixed word embeddings and BERT and RoBERTa representations, their results show similar, modest success in the 55-65% accuracy range at the classification task for most methods, more so for mean-pooled transformer encodings than for the [CLS] token, with the best performance coming from the FFNN using RoBERTa embeddings, which obtains 68% accuracy.

At time of writing, it has only recently been proposed by Zhang and Lepage (2023) to improve the sentence embeddings obtained from BERT, RoBERTa, and Sentence-BERT, by finetuning them on a contrastive objective similar to InfoNCE (Hoffmann et al., 2022) where positive examples are pairs from the same relation set. Ultimately, they find two-digit improvements on the Semantic Textual Similarity task, even for Sentence-BERT, and single-digit improvements on the task of classifying true vs false quadruples using analogies from an analogy test set. They automatically construct a dataset of definition sentences paired with words from BATS which they call DSBATS, consisting of semantic sentence analogies using sentences pulled from the semantic network BabelNet (Navigli and Ponzetto, 2010), corresponding to the encyclopedic and lexicographic category of relations taken from BATS (Gladkova et al., 2016). Doing so, they sidestep the surface similarity inherent in the template construction methods of Zhu and de Melo (2020) seen in Section 1.5.2. To take an example from the authors, using the BATS animal:sound relation, for the pair *pigeon* : *coo* they

[9] Examples drawn from `https://catalog.ldc.upenn.edu/desc/addenda/LDC2019T05_examples.html`.

recover from BabelNet multiple sentences, defining both "pigeon" ("Wild and domesticated birds having a heavy body and short legs") and different senses of the word "coo" ("To make a soft murmuring sound, as a pigeon", "Bird vocalization includes both bird calls and bird songs", "Murmuring sound made by a dove or pigeon"). We could expect a resulting quadruple to be *Wild and domesticated birds ...* : *Murmuring sound made by a dove ...* :: *Alert carnivorous mammal with pointed muzzle and ears and a bushy tail* : *The long plaintive cry of a hound or a wolf* for the lexical equivalent *pigeon* : *coo* :: *fox* : *howl*. In this manner they collect hundreds and thousands of sentence pairs per relation, respective to the encyclopedic and lexicographic BATS categories, without resorting to the use of only very few templates. Instead, each DSBATS sentence is unlikely to have unwarranted surface similarity with any other.

1.5.5. Figurative and predictive analogy

Outside of proportional analogies, the great strides that have been made in natural language processing and language modeling have inspired some to return to analogical reasoning in general. These works target more abstract or structural relations, or deliver analogy problems directly to language models without constraining or modifying the representations they derive from their input, letting them extract relevant features themselves. These projects ressemble the figurative and predictive analogies we described in Section 1.3 in following with Brown (1989). While this thesis does not target this type of analogy, instead focusing on those called proportional, we will take this opportunity to present briefly some of the most recent and exciting strides in the computational modeling of analogical reasoning.

Sultan and Shahaf (2022) take on the structural analogy framework of Gentner (1983) on procedural texts, without relying on hand-constructed representations. They use a neural model to label semantic roles and align the extracted roles between pairs of texts, providing interpretable structural mappings. We show an example drawn from the authors' publication in Figure 1.3. They also attempt to discover analogies from question-answering corpora, where the answers provide the analogous stories. They find that using Sentence-BERT to rank analogical candidates in the corpus by the similarity of their questions results in zero false positives, but that virtually all analogies found are self-analogies, i.e. describe the same circumstances, rather than two different situations both sharing an underlying relational structure. In comparison, a score based on their mapping method finds 3-4 times more true analogies than Sentence-BERT, although over half of the analogies it finds are either self- or non-analogies.

Wijesiriwardene et al. (2023) leave the proportional analogy framework to develop a dataset of analogies between two sequences of text which they use to evaluate embeddings

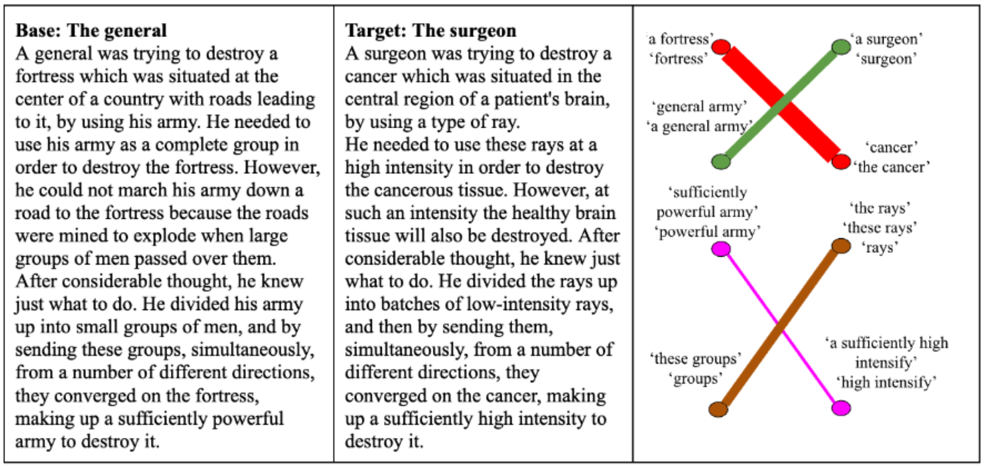

Figure 1.3. Example analogical "attack-dispersion" procedural texts, originally from Gick and Holyoak (1980), and extracted analogical alignments from (Sultan and Shahaf, 2022) Nodes represent clusters of text spans linked to a single entity. Line thickness represents the similarity of analogical entities in terms of the role they fill in the story.

from transformer language models by computing distance metrics on the vectors corresponding to two texts. That is, a pair of sequences which are analogical in some way should be highly related. Their dataset is structured taxonomically, rising in levels of abstraction from word pairs, pairs of word pairs, pairs of a word and a sentence, of sentences and their corruptions, NLI pairs, and, at the highest level, pairs of proverbs and stories that express them as well as quotes and their elaborated explanations. For words, they use the familiar word analogy datasets shown in Section 1.4.3. Proportional analogies of words and sentences are made from a crossword dataset (e.g. *amen* : *famous last words*) and from word-definition pairs. They create some pairs of sentences by surface edits (random deletion, masking, reordering, replacement by a synonym). Evaluating eight language models, their results show that at higher levels of abstraction, T5 (Raffel et al., 2020), BERT, SpanBERT (Joshi et al., 2020), and ELECTRA (Clark et al., 2020) show the lowest distance (normalized for each model) between the two terms of analogical pairs, depending on the exact distance metric used.

Hu et al. (2023) translate Raven's visual "progressive matrix" tests generated by Zhang et al. (2019) into a text-based abstraction, upon which they find that pretrained OPT (Zhang et al., 2022) and GPT-3 (Brown et al., 2020) language models in the hundred-million to hundred-billion parameter size range approach or surpass human performance as they scale. These progressive matrices consist in finding a correct visual solution given example rows of varying geometric images. Specifically, given a 3x3 grid of images, the ninth image in the bottom right must be chosen among a set of candidates in order such that the progression of images in the final row matches that of the first two rows. Webb et al. (2023) find that GPT-3 performs at or above human level in solving progressive matrices, the letter string analogies introduced by (Hofstadter, 1995), proportional verbal analogies, and approach human scores on analogical story problems, all of these described in text directly in the model's input space. We show an example of the text abstraction for these progressive matrices in Figure 1.4.

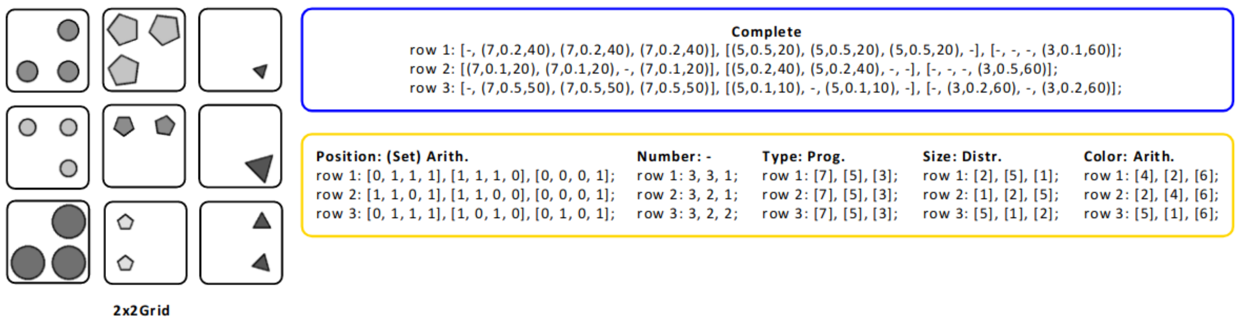

Figure 1.4. Example of text-based abstraction of a progressive matrix problem Figure taken from Hu et al. (2023).

Chapter 2

Sentence analogy test set

In this chapter we present a dataset of relational pairs of sentences which we shall use in this work as a benchmark for solving proportional analogies. There are several considerations to make in determining what sorts of analogies we'd like to solve. We have seen, in Chapter 1, proportional analogies constructed from relational pairs of formal (syntactic or morphological), semantic (e.g. NLI, paraphrase), or encyclopedic (definitions) nature. We would like to include in our analogies these kinds of relation, as well as some common sense relations such as opposition of meaning or sentiment, cause and effect, a change in temporal context, or "abstract" ones closer to the structure-mapping kind, where the same underlying roles are played by different entities in different contexts.

While we have shown several analogy datasets of natural language sequences in Section 1.5, many of those works had not been published at the time of our experimentation, and others are either not available publicly, may not provide desirable analogies by virtue of expressing unclear or inconsistent relations—such as the semantico-formal analogies of Lepage (2019) or the paraphrases and discourse pairs of Barbero and Afantenos (2023)—or are of undesirable construction such as the template-based analogies of Zhu and de Melo (2020). For this reason, we curate and indeed construct our own sentence pairs to ensure an (1) adequate diversity of relations and (2) that they constitute relatively "valid" analogies, by the author's manual vetting process (which, of course, introduces its own predispositions). Our sentence analogy test set (which we will shorten to SATS) is a collection of pairs of sentences obeying such relations.

We construct 32 relation sets (listed in Table 2.1) of 50 pairs of sentences each (for reason of having an equal number of pairs per relation, as well as limiting the amount of labor required), with pairs formed by meaning-preserving formal changes, meaning-altering formal changes, semantic structure-preserving changes, and changes requiring world knowledge. This is described in more depth in Section 2.1. Example pairs for each relation can be

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-extraposition		Semantic	cause-effect
	Syntactic	qa2d-declarative-howmany		Semantic	description-state
	Syntactic	qa2d-declarative-when		Semantic	home-outdoors
	Syntactic	qa2d-declarative-who		Semantic	simple-implicative-entailment
Validation	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

Table 2.1. SATS relations by category and split

seen in Tables 2.5 to 2.8. This results in 3,200 sentences, of which 3,024 unique sentences. As we've shown in Chapter 1, analogies can then be constructed by shuffling each pair of sentences of a same relation set against each other. In total, from 50 pairs we obtain 2,500 analogies making 80,000 analogies out of our 32 relations, or 2,450 and 78,400 respectively when we exclude $a : b :: a : b$. For the purposes of evaluating methods where parameters are fit on this data, we form training, validation, and test splits along relation set lines. We do this in order to gauge analogy-solving ability which generalizes beyond relations that have been observed. This is in opposition to, for example, fitting on analogies from the first 25 pairs of each relation and evaluating generalization to unseen pairs, which we consider would allow only for much weaker conclusions to be drawn. In an attempt to provide a dataset analogous to BATS (Gladkova et al., 2016), we somewhat similarly bin our relation sets into coarse types we call Encyclopedic, Lexical, Semantic, and Syntactic for later aggregation of results, and obtain sentence pairs matching BATS word pairs where possible for the hypernym-animal, misc-hypernym, person-occupation, meronym-substance, capital-country, and country-language relations.

2.1. Dataset construction

For most relations, we proceed by manually constructing sentence pairs, whereas others are manually retrieved or selected from an existing dataset. Specifically, we collect 6 relation sets from the Question to declarative sentence (QA2D) (Demszky et al., 2018) dataset, and 8 from Wikipedia. We show statistics on our data in Table 2.3 and example sentences in Tables 2.5 to 2.8.

QA2D was created from question answering (QA) datasets for the purpose of inducing NLI pairs between declarative statements (the hypothesis) and the passages which serve as the basis for questions (the premise). They use the preexisting passages and questions with rule-based and neural models to obtain declarative sentences. While we are not interested in NLI, we can make good use of pairs of declarative sentences and questions, since they obey a relatively fixed syntactic relation. For example, for a passage "I saw Gonzague yesterday. He was at the cafe.", a declarative-question pair could be *Gonzague was at the cafe yesterday.* : *When was Gonzague at the cafe?* An analogy could then be made with another declarative-"when" question pair, such as *Napoleon was defeated at Waterloo in 1815.* : *When was Napoleon defeated at Waterloo?*

Syntactic relations are curated in the following way. For the QA2D-derived relation sets, we make one relation set for several types of question (what, when, where, who, how many, how much), with sentence A being the answer and sentence B the question. This is done in order to reduce ambiguity, since one question can be answered by many premises, whereas a question can only be formed from a declaration in limited ways. Other relations are formed by taking a "canonical" sentence and applying a meaning-preserving change in its form as follows. For one relation, we take pairs of active and passive sentences (e.g. *The car destroyed a lamppost* : *A lamppost was destroyed by the car*). A second relation involves the extraposition of a phrase, where the sense of the sentence remains the same, but a syntactic constituent is shifted to the end of a clause, as in the analogy *To suggest that is treason* : *It is treason to suggest that* :: *When they decided can't be known* : *It can't be known when they decided.* A third set involves the movement of a particle associated to a phrasal verb to the end of the sentence (e.g. *I* ***threw out*** *all those old boxes* : *I* ***threw*** *all those old boxes* ***out***). Finally, we have a set of pairs of causal sentences using "because", where we invert the clauses and use "so" instead (e.g. *I ate all the cookies because I was hungry* : *I was hungry, so I ate all the cookies*). It should be noted that such relation sets involve very little if no change in a sentence's vocabulary, which makes solving analogies by retrieval extremely easy if there is a small or otherwise unchallenging candidate pool.

For the encyclopedic relation sets, we sourced sentences from Wikipedia, taking each sentence in a pair as the first sentence found from the most relevant article on the English language Wikipedia, sourced manually. For example, a pair from the person-occupation set may be (Strauss, composer). In such a case we would take the first sentence from the article for Richard Georg Strauss, and the first sentence from the article for "composer". This is similar to the DSBATS dataset of Zhang and Lepage (2023) described in Section 1.5.4. We do not require each sentence to be unique, since the encyclopedic relations are generally many-to-one, e.g. multiple individuals have the same occupation, many different animals share the same clade, and so on. See, for example, the following analogy from the meronym-substance relation: *A lens is a transmissive optical device which focuses or disperses ...* : *Glass is a non-crystalline, often transparent amorphous solid ...* :: *A mirror or looking glass is an object that reflects an image.* : *Glass is a ...* We show the number of unique sentences (both a and b together, thus out of 100 sentences) for such relations in Table 2.2. As mentioned earlier, we inspire ourselves from several of the encyclopedic and lexicographic relation sets in BATS (Gladkova et al., 2016), taking pairs of sentences that correspond to their word pairs where possible for the hypernym-animal, misc-hypernym, person-occupation, meronym-substance, capital-country, and country-language BATS relations.

Relation	Number of unique sentences
country-language	73
hypernym-animal	63
member-band	99
meronym-substance	82
misc-hypernym	73
person-occupation	70
simple_implicative-entailment	95

Table 2.2. Number of unique sentences for many-to-one relations

In our so-called lexical category, we pair sentences by changing the tense and other temporal markers, i.e. from present to past or future, or from past to future, from present to past (e.g. *We're eating here today* : *We were eating here yesterday*). We also create some relational pair sets by taking sentences with numerals and their approximations (*I have 62 files* : *I have about 60 files*), or instead by spelling the numbers out in words (*I have 62 files* : *I have sixty-two files*). We also take pairs of sentences with common English idioms and their replacement with literal descriptors. For example, *He hit the hay* : *He went to sleep* :: *Stop rubbing it in* : *Stop making me feel worse.*

The semantic category involves relations which are less obvious or more abstract, such as pairs of sentences in relations of cause and effect (*A rock is thrown at a window* : *The window shatters* :: *Your foot gets caught on a vine while running* : *You fall forwards violently*), descriptions and the state of being they describe (*He makes a phone call and a yellow car shows up a few minutes later.* : *He calls a cab.* :: *A cloud of steam rises off a cup of water.* : *The water is hot.*), sentences opposite in sentiment—such as in the pair *I care so much* : *I worry so much*—in meaning (e.g. the pair *Sometimes people lie* : *People always tell the truth*), or in formality (*Got a sec?* : *Do you have a moment?* :: *This beater's a hunk of* *junk.* : *This car is old and broken.*). Another set consists of sentences describing a scene or action at home paired with equivalent sentences occurring outdoors, for example *I could probably find it in my drawer at home* : *I could probably find it on my desk at work.* Finally, we construct entailment pairs using implicative verbs or phrasal verbs which imply a truth value about an associated verbal phrase, whether positive or negative, depending on the verb (Karttunen, 2012). For example, "She admitted to eating the cookie" implies she ate it, whereas "They neglected to inform students" implies they did not inform students, despite both verbs being left unnegated. Similarly, for some verbs, negation does not necessarily imply the opposite truth value. We construct pairs of sentences which both imply and negate one another, providing some built-in distractors. For example, see the following analogy made from phrasal verb-based NLI pairs: *They lacked the sense to speak their minds* : *They didn't speak their minds* :: *They didn't lack the sense to speak their minds* : *They spoke their minds.*

Number of characters

	Mean ± Std	Min	10%	25%	50%	75%	90%	Max
Encyclopedic	164 ± 80	20	74	103	152	209	267	541
Lexical	32 ± 12	11	18	24	31	39	48	106
Semantic	35 ± 15	7	18	23	32	44	55	91
Syntactic	56 ± 25	15	30	39	52	68	84	203
Total	69 ± 66	7	22	30	45	76	161	541

Table 2.3. SATS sentence length statistics

2.2. Limitations and bias

While the crude binning of our relation sets is useful for overall analysis, it remains somewhat arbitrary, as whether some relations ought to be considered semantic or lexical is not always clear. Indeed, different words hold different meanings, though whether time, for example, is much more about a choice of words or verb conjugation rather than a change in context is dependent on our framing and construction of the relation. Idioms might be considered lexical items, since they are set phrases with semantics that cannot usually be obtained by composing that of their individual parts (e.g. to spill the beans means to tell a secret, as long as one is already aware of this sense). Either way, the categorization of our relation sets does not generally impact the rest of our methodology or experimentation outside of providing a denser view of some results or statistics.

Encyclopedic	Lexical	Semantic	Syntactic
0.16	0.57	0.39	0.55

Table 2.4. Jaccard similarity of SATS pairs by category

It is worth noting some particularities of our data. As can be seen in Table 2.3, our encyclopedic sentences are much longer than the other categories, due to their direct extraction from Wikipedia, often containing footnote indicators, citations, or pronunciation hints, which can be seen in the examples of Table 2.5. While these may be considered noise, we opt not to preprocess them in any way. Sentences from other categories are generally quite short, and in pairs have a high proportion of shared vocabulary. The QA2D sentences, by nature of the question-answering task they are drawn from, chiefly concern historical fact, statistics, and so on, though the relation represented is not encyclopedic. These sentences are somewhat longer than those we construct, though they similarly have high surface overlap. We show the average Jaccard similarity of sentence pairs in Table 2.4, with a per-relation breakdown shown in Table A.1, calculated as follows for two whitespace-tokenized sentences whose sets of unique words are A and B:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.2.1}$$

Manually constructing sentences introduces the bias of the writer, whether by writing style or lack thereof, a preference toward certain words or describing certain contexts, or an overall lack of diversity thereof. It is evident from these points that constructed sentences do not represent a natural usage of language. These sentences are not generally conceived with a communicative goal in mind, quite unlike most uses of language. Despite this, we believe SATS can serve to test reasoning on proportional sentence analogies, though some confusion may result during evaluation from the non-uniqueness of many of the relations we study, as some relations may be fulfilled by any number of different sentences of mildly or greatly differing wording or construction. In the time since we've begun, recent work has presented analogy datasets or methods for creating them, as shown in Sections 1.5.4 and 1.5.5, which likely do not suffer from the above limitations.

Relation	A	B
capital-country	Abuja (...)[4] is the capital and eighth most populous city of Nigeria.	Nigeria (... Listen), officially the Federal Republic of Nigeria, is a country in West Africa.
country-language	Syria (Arabic: ..., romanized: Sūriyā), officially the Syrian Arab Republic (Arabic: ..., romanized: al-Jumhūrīyah al-'Arabīyah as-Sūrīyah), is a Western Asian country located in the Eastern Mediterranean and the Levant.	Arabic (... (listen) or ... (listen) or ...) is a Semitic language that first emerged in the 1st to 4th centuries CE.
hypernym-animal	The chimpanzee (Pan troglodytes), also known simply as chimp, is a species of great ape native to the forest and savannah of tropical Africa.	The Hominidae (...), whose members are known as the great apes[note 1] or hominids (...), are a taxonomic family of primates that includes eight extant species in four genera: Pongo (the Bornean, Sumatran and Tapanuli orangutan); Gorilla (the eastern and western gorilla); Pan (the chimpanzee and the bonobo); and Homo, of which only modern humans remain.
invention-creator	In plumbing, a trap is a U-shaped portion of pipe designed to trap liquid or gas to prevent unwanted flow; most notably sewer gases from entering buildings while allowing waste materials to pass through.	Thomas Crapper (baptised 28 September 1836; died 27 January 1910) was an English plumber and businessman.
member-band	Bradford Phillip Delson (born December 1, 1977) is an American musician, best known as the lead guitarist and one of the founding members of the rock band Linkin Park.	Linkin Park is an American rock band from Agoura Hills, California.
meronym-substance	A penny is a coin (pl. pennies) or a unit of currency (pl. pence) in various countries.	A metal (from Greek ... métallon, "mine, quarry, metal") is a material that, when freshly prepared, polished, or fractured, shows a lustrous appearance, and conducts electricity and heat relatively well.
misc-hypernym	Cake is a flour confection made from flour, sugar, and other ingredients, and is usually baked.	Dessert is a course that concludes a meal.
person-occupation	Richard Georg Strauss (German: [...]; 11 June 1864 – 8 September 1949) was a German composer, conductor, pianist, and violinist.	A composer is a person who writes music.

Table 2.5. Examples of encyclopedic SATS sentence pairs per relation, with phonetic transcriptions replaced with ellipses due to typesetting errors

Relation	A	B
idiom-literal	They have a bone to pick with him.	They have a complaint with him.
numeral-spelled	Back in the 90s I was in a very famous TV show.	Back in the nineties I was in a very famous TV show.
numeric-approximation	I have 265 friends.	I have about 250 friends.
past-future	The past was hopeful.	The future will be hopeful.
present-future	She knows we were there.	She will know we were there.
present-past	She isn't on her way to the gala.	She wasn't on her way to the gala.

Table 2.6. Examples of lexical SATS sentence pairs per relation

Relation	A	B
active-passive	Several people trespassed on the property.	The property was trespassed on by several people.
because-so	The forest was razed because they needed the lumber for ships.	They needed the lumber for ships so the forest was razed.
canonical-extraposition	To do so is immoral.	It's immoral to do so.
canonical-verb-particle-movement	She threw up last night's dinner.	She threw last night's dinner up.
qa2d-declarative-howmany	The Prelude field was estimated to contain 3 trillion cubic feet of natural gas reserves .	The Prelude field was estimated to contain how many cubic feet of natural gas reserves ?
qa2d-declarative-howmuch	80 % of the population of Salvado , Bahia is black or mixed race .	How much of the population of Salvador , Bahia is black or mixed race ?
qa2d-declarative-what	The famous French leader Napolean had established the Polish state at this time .	What famous French leader had established the Polish state at this time ?
qa2d-declarative-when	In September 1829 , Chopin returned to Warsaw .	When did Chopin return to Warsaw ?
qa2d-declarative-where	The tracheae is located in the body cavity .	Where is tracheae located ?
qa2d-declarative-who	The South primarily initiated the clashes along the 38th parallel .	Who primarily initiated the clashes along the 38th parallel ?

Table 2.7. Examples of syntactic SATS sentence pairs per relation

Relation	A	B
cause-effect	A woman has not eaten in many hours.	The woman is hungry.
description-state	A man salivates at the thought of food.	A man is hungry.
home-outdoors	He gave himself a haircut in the bathroom.	He went for a haircut at the barbershop.
informal-formal	We gotta get going.	We need to start moving.
phrasal-implicative-entailment	She didn't miss the occasion to dress him up.	She dressed him up.
sentence-opposite	These beautiful flowers grow so fast.	These shriveled husks can't grow.
sentiment-good-bad	An exciting tale that makes its three hours go by in a flash.	A mind-numbing ordeal that makes its three hours drag out forever.
simple-implicative-entailment	He didn't predict that they wouldn't go.	They didn't go.

Table 2.8. Examples of semantic SATS sentence pairs per relation

Chapter 3

Experiments

We introduced in Sections 1.4 and 1.5 common methods for solving proportional analogies notated $a : b :: c : d$ the terms of which are units of natural language, whether words or sequences. These methods typically obtain for each term a vector representation in keeping with the distributional hypothesis of meaning in language (Firth, 1957). The vector representations of the premise (a,b,c) are operated on to obtain one which is hoped to be nearest to that of the conclusion d, at which point a solution can be retrieved from a corpus or list of candidates, or decoded as a sequence. In a different framework, the entire problem can be posed in natural language and provided as a prompt to a language model in the hopes of generating the solution d by conditioning on input containing (a,b,c). We thus conduct two kinds of experiments, introduced below and described in depth in this chapter: we solve proportional sentence analogies by (1) retrieval in a VSM by operating on sentence embeddings, and (2) by generating the desired solution, conditional on the premise sentences. We analyze the results obtained from these experiments in Chapter 4.

We conduct experiments to assess the suitability of vector representations of natural language for solving sentence analogies by retrieval in line with the techniques shown in Chapter 1. As vector representation baselines, we examine fixed and contextual embeddings presented in Section 3.1. We examine a variety of previously introduced vector space solving methods—described in Section 3.3—which are both unparameterized and having parameters which must be fit. We introduce in Section 3.2 a vector autoencoder which we finetune from a pretrained language model, whose bottleneck vector we use as a sentence representation. We additionally finetune this autoencoder on our analogies with a vector solver in an end-to-end fashion (see Section 3.3.4). We evaluate the combination of embeddings and vector solving methods on the retrieval-based analogy task (Section 3.4).

In addition to retrieval, we experiment with solving analogies by generating the desired sequence. Although we use the end-to-end autoencoder and vector solver mentioned above, departing from solvers defined by vector operations, we also finetune an encoder-decoder language model to conditionally generate the solution to an analogy in a sequence-to-sequence framework (described in Section 3.5.2). In addition to this, we attempt to generate solutions to our proportional analogies in a few-shot setting, i.e. by providing a prompt with instructions and examples for the task, requiring no further parameter tuning (Section 3.5.3). Relying on the scaling of language model abilities with parameter size, we evaluate models of different scales from the million to billion-plus parameter regime.

All analogy solvers and language model inference and training routines are implemented using the PyTorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2020) libraries, with models fit by gradient descent using reverse mode automatic differentiation.

3.1. Pretrained models

3.1.1. FastText

Grave et al. (2018) provide 300-dimensional word embeddings fit using CBOW on data from Wikipedia and Common Crawl.[1] These embeddings incorporate sub-word information, allowing to represent out-of-vocabulary words. We sum word embeddings over a whole sentence split by whitespace to obtain its embedding, using their FastText Python module[2] for the purpose.

We initially considered comparing these word embeddings to a bag-of-words (BoW) representation, though informal initial experimentation showed it to perform similarly enough. For this reason, we found it was not worth the increased computation time resulting from the massive dimensionality—equal to the vocabulary size—of BoW vectors.

3.1.2. BERT

We introduced in Section 1.5.1 the BERT model (Devlin et al., 2019), whose pretraining scheme intends to learn general-purpose parameters for later finetuning by masked language modeling and next sentence prediction. We take the average of all token encodings as a sequence's embedding, both here and in subsequent transformer encoders. We use the Base non-case-sensitive model,[3] which has 110M parameters. Its hidden layers and hence final encoding vectors are 768-dimensional, which will be the same for subsequently described models of "Base" size.

[1] https://commoncrawl.org

[2] https://fasttext.cc/docs/en/python-module.html

[3] https://huggingface.co/bert-base-uncased

3.1.3. RoBERTa

Liu et al. (2019) present RoBERTa, a transformer pretraining method wherein they modify the BERT training procedure to remove the next sentence prediction task, use larger batches of longer sequences, and dynamically generate [MASK] tokens for each training example. Our experiments use the RoBERTa Base checkpoint,[4] which has 125M parameters.

3.1.4. DeBERTa

DeBERTa is another development on BERT and RoBERTa which set state-of-the-art records for natural language understanding benchmarks (He et al., 2020). Instead of typical additive positional embeddings—vectors added to vocabulary embeddings in order to incorporate temporal information into the transformer's otherwise order-invariant computation—DeBERTa computes attention scores with separate matrices for content and position. This model has a Base size of 100M parameters.[5]

The third version of DeBERTa further improved upon the state of the art by replacing the masked language modeling objective with a replaced token discrimination task. This task originates in the ELECTRA model (Clark et al., 2020), which is trained in two parts. A generator model is trained on the MLM task, and provides a probability distribution on the masked tokens of each training batch. From this they sample a replacement token which a discriminator model must detect in a binary classification task. Both models have their own parameters with the exception of the embeddings, which they prevent the discriminator's gradient from updating. The final model uses the discriminator's parameters, and has 86M parameters for its Base size[6].

3.1.5. Sentence-BERT

Here, Sentence-BERT refers to any of a collection of encoder transformers which are finetuned to cluster semantically related input sequences by pooling their final encodings and provided by the Sentence-Transformers team (Reimers and Gurevych, 2019). With their Sentence-Transformers Python library,[7] we use the all-mpnet-base-v2 checkpoint[8] (133M parameters) based on the MPNet model, a transformer encoder with a pretraining objective based on masked and permuted language modeling which we may associate with BERT and

[4] https://huggingface.co/roberta-base

[5] https://huggingface.co/microsoft/deberta-base

[6] https://huggingface.co/microsoft/deberta-v3-base

[7] https://www.sbert.net/

[8] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

XLNet (Yang et al., 2020) respectively. This is finetuned with a contrastive objective on 1 billion semantically related pairs of text sequences.

3.1.6. Instructor

The Instructor model (Su et al., 2023) aims to provide general-purpose embeddings by directing it using instructions, being trained on a contrastive objective on the mean-pooled encodings of instruction-input sequences over 330 datasets from various domains. For example, we might use this example (taken from the original authors): "`Represent the Review sentence for classifying emotion as positive or negative:`" for sentiment classification.

We use the Instructor Large checkpoint[9] (1.5B parameters, 1024-dimensional hidden state) with the prompt "`Represent the sentence for solving analogies of surface form and meaning:`". While we would prefer to provide the type of relation in the prompt, we obtained degenerate results in our initial attempts to do so, which we will leave as a limitation of this work.

3.1.7. Flan-T5

The Text-To-Text-Transfer-Transformer (T5) model (Raffel et al., 2020) is an encoder-decoder transformer which is first pretrained on an unsupervised denoising task, followed by finetuning on a multitude of supervised tasks. All downstream tasks are reframed in a text-to-text setting, where an instruction prompt coupled with input is given and a language modeling objective optimized against the output. For example, a translation task would be framed as "Translate from English to German: I am thirsty". The decoder applies cross-attention to the encoded inputs and should autoregressively generate "Ich habe Durst". The model uses a SentencePiece tokenizer (Kudo and Richardson, 2018) with a vocabulary of 32,000 tokens.

Flan-T5 (Chung et al., 2022) improves upon this by further finetuning on an additional 1,836 instruction-based tasks. We use the Flan-T5 Base model as a baseline as well as for our finetuned vector autoencoder. For our finetuned sequence-to-sequence model, we use both the Base and Large sizes. In our few-shot experiment, we use the Flan-T5 model at all its parameter sizes of 250M (Base),[10] 780M (Large), 3B (XL), and 11B (XXL).

---

[9] https://huggingface.co/hkunlp/instructor-large

[10] https://huggingface.co/google/flan-t5-base

3.2. Finetuned Flan-T5 autoencoder

We would additionally like to move beyond retrieval by instead generating sequences when solving proportional analogies using functions of real vectors. To achieve this, we finetune the pretrained Flan-T5 Base model parameters using a slightly modified transformer architecture where we mean-pool the encodings at the final layer of the encoder portion of the transformer. From this we obtain a bottleneck vector from which the decoder should autoregressively generate the desired sequence, as depicted simplistically in Figure 3.1. That is, the transformer's cross-attention is only performed against this pooled vector, which we use as our sentence embedding. Such a model would permit us to operate on the bottleneck vector and, rather than retrieving the solution, generate it. However, the model described in this section is trained simply as an autoencoder; it is not in its own right meant for solving proportional analogies. In the generative experiments described in Section 3.5, we do not use this autoencoder, but its end-to-end finetuned solver variant (see Section 3.3.4).

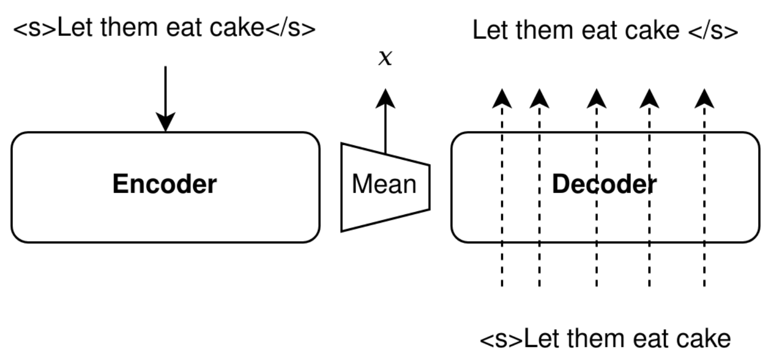

Figure 3.1. A visualization of the mean-pooled Flan-T5 autoencoder architecture x refers to the bottleneck vector which serves as an embedding. The arrows depict the input and output of a single autoregressive decoding step. <s> and </s> stand in for the beginning of sequence and end of sequence tokens.

The model is fit to autoencode sentences—from a dataset described below—with a NLL loss on batches of 72 sentences padded up to a length limit of 500 tokens, i.e. each sequence in the batch is one sentence. We use a learning rate of 3×10^{-5} and the Adafactor optimizer (Shazeer and Stern, 2018) which the original Flan-T5 model is pretrained with. A training artifact is the initial use of a linear warmup from zero of the learning rate with a peak at the 10,000th batch out of 100,000. Seeing that this underfit, we left the learning rate constant until 281,600 batches. Ultimately we reached a validation loss (lower is better) of 0.372 on 1024 held out sentences compared to the pretrained Flan-T5's 0.30. This training was done in a single epoch, with no intentional repetition of data, on a dataset described below. Due to this training procedure, the model did not overfit (and indeed did not finish converging due to the abundance of data).

We source sentences from the Online Language Modelling (Thrush et al., 2022) dataset of Wikipedia articles dated 20-12-2022,[11] which we interleave with sentences drawn from the Reddit comments dataset[12] provided by the Sentence-Transformers team (Reimers and Gurevych, 2019) using the Huggingface Datasets library (Lhoest et al., 2021). We take the first million training examples from both sources in a round-robin fashion and use a Python wrapper package for Microsoft's Blingfire library[13] to split them into sentences. Reddit comments are chosen as a data source for their range of styles, which we can characterize as colloquial, formal, argumentative, or conversational, and which often concern varied topics and everyday concerns. Wikipedia is useful for its formal style and encyclopedic information. However, we expect the model not to learn (and indeed forget from its pretraining) discourse patterns and the overall relatedness of such facts as a result of splitting examples into individual sentences.

3.3. Vector solvers

We discussed in Sections 1.4 and 1.5 a number of methods for solving proportional analogies whose terms have a distributed vector representation. We will apply some of these to the representations extracted from the models listed in Sections 3.1 and 3.2.

Two unsupervised solving methods are used: 3CosAdd, and, as a crude baseline, the mean of the premise vectors $\hat{d} = \frac{a+b+c}{3}$. The former method has significant representation in the literature, and appears to work best insofar as vector representations of distributional semantics might actually permit analogies as parallelograms in the vector space. However, that point is debated, as discussed previously.

Alongside these we use two parametric solvers which are fit on analogies from SATS (see Chapter 2). The first is a dense feedforward neural network composed of residual blocks. The second is the Abelian group network of Abe et al. (2021) described in Section 1.4.6. We report their architecture and hyperparameters used, which were settled on by an informal manual search.

3.3.1. Feedforward solver

Inspired by the feedforward neural network solver of Wang and Lepage (2020), we similarly use one to parameterize a vector solver $\hat{d} = g(a,b,c)$—where (a,b,c) represent their

[11] https://huggingface.co/datasets/olm/olm-wikipedia-20221220

[12] https://huggingface.co/datasets/sentence-transformers/reddit-title-body

[13] https://github.com/microsoft/BlingFire

respective sentences' embeddings—whose architecture is a five layer stack of residual blocks f using the Gaussian error linear unit (GELU) activation function (Hendrycks and Gimpel, 2016) and layer normalization (Ba et al., 2016):

$$x_1 = \mathrm{LN}\left(\mathrm{GELU}\left(W_1 x + b_1\right) + x\right) \tag{3.3.1}$$

$$f(x) = \mathrm{LN}\left(\mathrm{GELU}\left(W_2 x_1 + b_2\right) + x_1 + x\right) \tag{3.3.2}$$

All hidden states retain the input dimensionality. To admit as input the concatenation of the premise vectors, the first block has the dimension of $a \circ b \circ c$, followed by an affine transformation which reduces it to the original embedding dimension.

3.3.2. Abelian solver

Let us recall Equation 1.4.5. The Abelian solver uses an invertible neural network to transform embeddings into a space of the same dimensionality before applying the vector offset in that space. The solution can be found from the resulting vector by passing it through the inverse of the neural network.

We use the FrEIA Python module (Ardizzone et al., 2018) which implements invertible blocks compatible with PyTorch, each combining the operations of affine coupling (Dinh et al., 2015), a permutation matrix sampled from the special orthogonal group, activation norm (Kingma and Dhariwal, 2018), and taking an inner function parameterized as a feed-forward neural network. We compose each such inner network of two affine layers with a GELU activation function:

$$f(x) = W_2 \mathrm{GELU}\left(W_1 x + b_1\right) + b_2 \tag{3.3.3}$$

3.3.3. Training

We train one solver of each type for each embedding model listed in Section 3.1 by fitting it on analogies using the training splits described in Chapter 2, including identity analogies $a : b :: a : b$. We use the Adafactor optimizer with a learning rate of 3×10^{-5} and a batch size of 8. We additionally add Gaussian noise on the order of 10^{-2} to the input vectors. We use as loss function the negative cosine similarity:

$$\mathrm{loss}(d,\hat{d}) = -\frac{d^\intercal \hat{d}}{\|d\| \cdot \|\hat{d}\|} \tag{3.3.4}$$

The solvers are trained for two epochs over the 30,000 training analogies created from our data, over which time we observe overfitting. For that reason the best checkpoint is selected on the validation split using the harmonic mean of the top-1 and top-5 accuracy for retrieval by cosine similarity, to encourage choosing a model that at least predicts the neighbourhood of the solution. Retrieval metrics are discussed in Section 3.4.

3.3.4. End-to-end decoder solver

In addition to solvers trained on the embeddings of each of the baseline models in Section 3.1, we further finetune the Flan-T5 mean-pooled autoencoder presented in Section 3.2 to decode the solution to our analogies found with the solvers in Section 3.3 by training them jointly in an end-to-end fashion.

For each sentence analogy $a : b :: c : d$, we individually encode each of the premises (a,b,c) using the mean-pooled embedding from the encoder portion E of our autoencoder. From these three embeddings, we predict a vector $x = f(E(a),E(b),E(c))$ using the solver f, which may be, as stated, any of vector arithmetic, the average premise embedding, or the parameterized feedforward or Abelian solver architectures presented in Sections 3.3.1 and 3.3.2. We then autoregressively decode a sentence from this "solved" vector x, just as in Section 3.2, by performing cross-attention on the vector. This architecture is depicted in Figure 3.2.

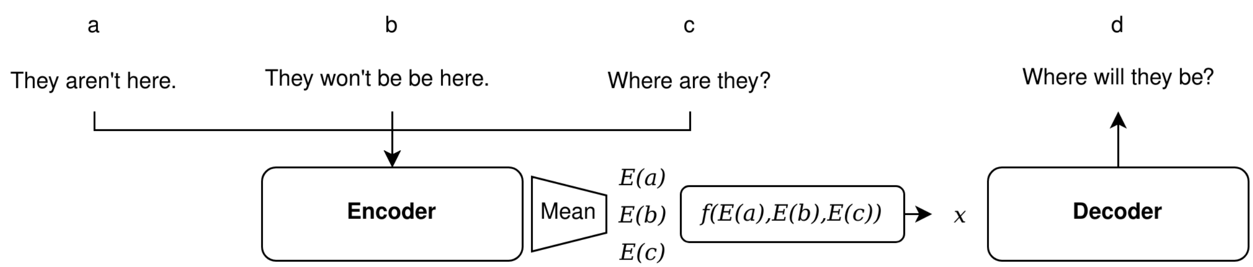

Figure 3.2. A visualization of the end-to-end solver and decoder architecture f is the vector solving method used, e.g. the offset method. Note that although not depicted, the output is still decoded autoregressively.

For each choice of f, we finetune this end-to-end solver for a single epoch on the same training split for SATS shown in Chapter 2 with a batch size of 64, using an NLL loss between predicted and true solution sentences, updating parameters with the Adafactor optimizer by a learning rate of 10^{-4}. Indeed, as all the solvers mentioned previously are differentiable, they can easily be inserted into the computational graph and loss gradients propagated through them. Observing overfitting, which is expected given the relative data duplication resulting from analogies composed of pairs of sentences, we select the best checkpoint on the validation split by METEOR score (Banerjee and Lavie, 2005).

3.4. Retrieval task

The models and vector solvers described above will now allow us to solve analogy tests by retrieval in a VSM. Let us restate Equation 1.4.2, assuming we have predicted a vector x

for our sentence analogy $a : b :: c : d$, represented by each term's embeddings:

$$\hat{d} = \underset{y \in C}{\operatorname{argmax}} \frac{x^\intercal y}{\|x\| \cdot \|y\|} \tag{3.4.1}$$

Considering all candidates in our candidate set C, we want $\hat{d}$, the one that maximizes its cosine similarity with x. Letting S be the set of embeddings of all unique sentences in SATS, the candidate set is usually $C = S \setminus \{a,b,c\}$, i.e. the premises (a,b,c) are excluded. This has the side effects of first occasionally excluding valid candidates, if the relation to which (a,b) and (c,d) belong is many-to-one, and secondly of artificially increasing accuracy. Indeed, in an analogy $a : b :: c : d$ constructed from pairs of terms in a certain relation, paired terms tend have the most similar embeddings anyhow. Thus excluding the premises makes uncertain whether we are evaluating the ability to use the embeddings to solve analogies or to simply cluster and retrieve related pairs in the absence of sufficient confounding data, as discussed in Section 1.4. We will see in our analysis in Section 4.1 that d is the nearest neighbor to c in a majority of cases unless distractors are added to the candidate set (see Table 4.3).

For this reason, we report the retrieval accuracy under four different conditions. First we use the traditional, "easy" analogy test, where premises are excluded. Second, we introduce a set of distractors constructed automatically as shall be described below from the unique sentences in SATS. Third, we remove the easing constraint excluding premises. Fourth, we both keep the premises and add the distractors in our candidate set.

For each of the 3,024 unique sentences in SATS, we split it into words by whitespace and construct a number of distractors by either randomly swapping pairs of words (e.g. from the phrasal implicative entailment relation, "**it.** didn't cross **He**"), randomly removing words (e.g. from the idiom-literal relation, "Stop ∅ **(being)** a wet blanket."), or by replacing them by a randomly chosen nearest neighbour (e.g. from the phrasal implicative entailment relation, "**her (She)** didn't confirm the deal.") from a VSM of word embeddings. Denoting length of the whitespace-tokenized sequence as L, we swap words a number of times equal to $\min\{\binom{L}{2}, 5\}$. Words are uniformly chosen for deletion $\min\{L, 5\}$ times, as well as for replacement. When replacing, we use the Python module for FastText (Grave et al., 2018) to use their English Common Crawl CBOW subword embeddings (see Section 3.1.1). We take the 20 nearest neighbours in the vocabulary of fewer than 25 characters, from which we uniformly sample. In this manner we generate 41,356 distractor sentences after removing any duplicates.

3.5. Generative task

As we have stated, in addition to solving analogies by retrieval, we would also like to generate solutions. First, we can use the end-to-end vector solver and autoencoder shown in Section 3.3.4 in order to jointly encode the terms of the analogy into vectors, apply a solving method to the premise vectors, and decode the solved vector into a sentence. However, we additionally experiment with framing the analogy test as sequence-to-sequence task, for which we finetune a model (described below in Section 3.5.2), and with few-shot inference without any finetuning (described in Section 3.5.3).

Different methods are used when decoding, depending on the model. For the end-to-end vector solver and the sequence-to-sequence models, we use both greedy decoding, which is prone to repetition of subsequences, and η-sampling, a truncation method with superior human plausibility judgements and which escapes repetitive patterns more often (Hewitt et al., 2022). In the few-shot setting we only used η-sampling. We set $\eta = 6 \times 10^{-4}$ based on the best hyperparameters found by Hewitt et al. for the GPT-2 Medium (355M) and Large (774M) models.

3.5.1. Metrics

By generating solutions, we hope to remove the difficulties in evaluating this ability by retrieval, which is limited by the number and properties of alternative candidates. However, we introduce the difficulty of an adequate sequence evaluation metric. For all model outputs, we report the word error rate (WER), METEOR score (Banerjee and Lavie, 2005), BLEU score (Papineni et al., 2002), exact match accuracy, and the copy-a, copy-b, and copy-c exact match rates. We use the Huggingface Evaluate library (Von Werra et al., 2022) for all but the copy rates. There are limitations to the metrics we use, of course. N-gram or other surface metrics, exact match included, discount what may be valid predictions if they have a different form, by use of synonyms, paraphrase, or otherwise. To alleviate this, methods like BERTScore (Zhang et al., 2020) attempt to apply the distributional representations learned by language models. However, it has been found that even such methods are biased to the presence of important content words, though more minute (yet invalidating) differences may be masked by surface features (Hanna and Bojar, 2021).

WER[14] is a sequence alignment-based metric, equal to the normalized edit distance, where, for S substitutions, D deletions, I insertions, and L the length of the reference sequence tokenized by whitespace, we compute

$$\mathrm{WER} = \frac{S + D + I}{L}, \tag{3.5.1}$$

[14] https://huggingface.co/spaces/evaluate-metric/wer

which can be greater than 1 if the number of errors is greater than the length of the reference.

BLEU score measures the geometric mean rate of token subsequence overlap for n-grams appearing in the prediction versus the reference, adjusted for length. We use $n = 4$ for the maximum order of n-grams and the provided whitespace and regex tokenizer.[15]

METEOR[16] matches word unigrams, additionally by related morphological forms and synonymy, calculating a final score based on a weighted harmonic mean of precision and recall, with a penalty to upweight well-ordered predictions.

3.5.2. Finetuning Flan-T5 for sequence-to-sequence analogies

We leverage the Flan-T5 Base (250M parameters) and Large (780M parameters) models for finetuning on SATS analogies as a sequence-to-sequence task. For an analogy $a : b :: c : d$, the input to the encoder portion of the transformer is the tokenized sentence a and the tokenized sentence b spliced with a separator token. For the separator token we use one of the special tokens used by the T5 tokenizer, so that a tokenized pair is formatted as a<extra_id_0>b. When inferring d, the decoder is prompted with the sentence c, performing cross-attention on the encoded input pair.

Given a pretrained checkpoint, we train two models, which we call "generator" and "solver", using two differents finetuning schemes. Given a quadruple $a : b :: c : d$, the solver-type model is trained to encode the concatenated sequence (a,b) and, performing cross attention, autoregressively decode d while being prompted with c, which is exactly the same as the task we wish to solve at test time (i.e. solve the analogy by finding the correct fourth sequence).

The generator-type model is trained to decode all of (c,d) concatenated together, with the intuition that a better training signal might be obtained by fitting a model which generates a pair (c,d) analogous to the input (a,b). Indeed, we could hope to then sample analogous pairs from a model trained in this manner, since for each pair (a,b) in a given SATS relation set, there are 49 other pairs of the same relation which serve as a target output (c,d) (see Chapter 2). However, the small amount of data repeated in this way may result in somewhat degenerate training dynamics.

These models are trained using the same SATS splits as elsewhere. We do not include trivial analogies $a : b :: a : b$ in order to reduce the likelihood of copying one of the input sentences, resulting in 29,400 training quadruples. Models are trained for 3 epochs with a batch size of 64 using the Adafactor optimizer with a constant learning rate of 3×10^{-5} for the Base models, 10^{-5} for the Large models, and a weight decay rate of 10^{-2}. We use an

[15] https://huggingface.co/spaces/evaluate-metric/bleu

[16] https://huggingface.co/spaces/evaluate-metric/meteor

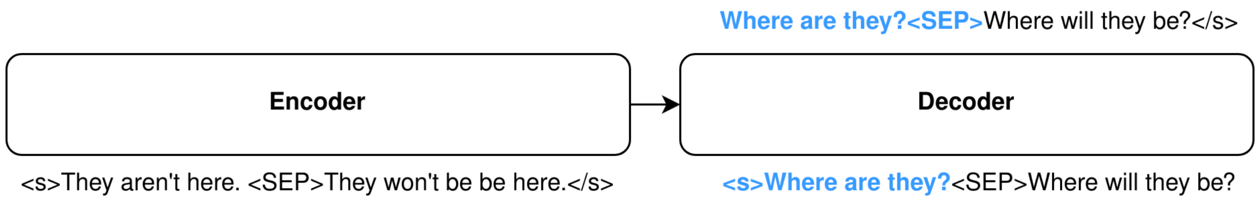

Figure 3.3. Depiction of the sequence-to-sequence analogy task
The bolded blue text refers to the prompt, i.e. sentence c, given as input, but which is not generated during inference. As described, during training the "generator"-type model autoregressively generates (c,d), unlike the "solver"-type model, which is trained to generate d while prompted with c. The encoder-decoder architecture depicted refers to the standard transformer architecture used by the Flan-T5 model.

NLL loss, which we also use to pick the best checkpoint on the validation set, which we observe always occurs within the first epoch of training.

3.5.3. Few-shot Flan-T5 solver

We noted in Section 1.5.5 that it has been observed that large language models can perform well by prompting them in a few-shot fashion, and that recent work attempts to solve analogies in this setting. Given that the Flan-T5 model is trained on text-to-text instruction-prompted tasks, we attempt to induce it to solve SATS analogies this way.

Informally, we experimented with different prompts and input formats. Our concerns for this are to reliably extract a solution from generated output, which may contain unrelated artifacts or reasoning preceding an answer, and, chiefly, to induce the language model to fulfill the task. Without wishing to truly turn to prompt engineering, we settled on what appeared to be an effective prompt template, shown in Figure 3.4. Searching for prompts informally, we found that we were able to elicit an appropriate response separated by the `(FINAL ANSWER)` substring using this prompt with ad-hoc analogies not found in SATS, which we found sufficient for further experimentation.

As noted earlier, we use the Flan-T5 Base, Large, XL, and XXL size models for this experiment. While the affordability of graphics processing units with sufficient memory for usage of large language models is relatively prohibitive, we use the Huggingface Transformers library's integration of 4-bit quantization methods (Dettmers et al., 2022) combined with the offloading of embeddings, parts of the encoder, and the language model head to CPU memory. These optimizations allow us to perform these experiments on a single NVIDIA RTX 3060.

```
Question: If "The car made it." becomes "Any car could make
it." then "I want that apple." becomes what? It seems something
definite has the article any used instead. (FINAL ANSWER) I want
any apple.
Question: If "Tables are great!" becomes "Chairs are great!"
then "Tables suck." becomes what? It appears the topic must
change from tables to chairs. (FINAL ANSWER) Chairs suck.
Question: If "He absolutely wouldn't" becomes "I don't think
he would." then "Drink it right away." becomes what? The
sentence changes from being very certain to being hesitant or
uncertain. (FINAL ANSWER) Drink it soon if you feel like it.
Question: If "[A]" becomes "[B]" then "[C]" becomes what? First
identify the change between the first two sentences, then apply it
to the third.
```

Figure 3.4. Prompt used for few-shot solving of proportional analogies

Chapter 4

Analysis

In this chapter we discuss the results obtained from the analogies solved as described in Sections 3.4 and 3.5. Results are shown in following with the data splits shown in Chapter 2. Trivial analogies of the form $a : b :: a : b$ are not included in any results shown.

4.1. Retrieval

We presented a number of embedding models in Chapter 3. These include the pretrained models, the autoencoder (AE-)Flan-T5, and the finetuned end-to-end (E2E-)AE-Flan-T5 models. We also presented the vector solvers we use in Section 3.3. These are:

(1) the arithmetic solver, also known as the vector offset method, or 3CosAdd when used to retrieve solutions by cosine similarity in a VSM;
(2) the Abelian solver, a trained model using an invertible neural network which transforms the premise vectors individually into a space where the vector offset method can be meaningfully applied;
(3) the feedforward solver, a conventional neural network which takes all three premises and regresses to a vector similar to the conclusion;
(4) the mean of the three premise vectors, a simple baseline.

For each model-solver combination, we evaluate the solving and retrieval accuracy demonstrated under different candidate set conditions: either with premises removed, premises included, premises removed and distractors added, or premises and distractors both included.

We will note straightaway a few details. First is that the feedforward end-to-end solver underfits the task when selecting its best checkpoint on the validation set and attains zero accuracy even on the training set. Second is that the end-to-end Abelian and arithmetic solvers perform virtually identically, which we will discuss further later. Third is, as expected, that the mean solver baseline is surpassed by virtually every other option.

4.1.1. Pairing consistency score

Before viewing the accuracies obtained, let us review the pairing consistency score (PCS) introduced by (Fournier et al., 2020), which we mentioned in Section 1.4.5. This score measures the linear regularity of a relation for a particular embedding space. For each of our relation sets of $n = 50$ paired sentences (a,b), calling the original offsets between them O, we sample $N = 50$ shuffled pairs whose offsets we call O_k for $1 < k < N$. For one set of offsets, we can call its set of offset similarities $sim(O)$, equivalent to the upper triangular matrix where $sim(O)_{ij} = \frac{o_i^\intercal o_j}{\|o_i\| \cdot \|o_j\|}$ (for $0 < i < j < n$). Considering each offset similarity therein as a binary class probability, we can take as a task to classify whether a pair of offsets belongs to the true set O or a false one O_k. PCS is computed from this as the average area under the receiver operating characteristic (AUROC) curve, which plots the proportion of false positives by true positives over an increasing positive class threshold:

$$\text{PCS} = \frac{1}{N} \sum_{k=1}^{N} \text{AUROC}\left(sim(O), sim(O_k)\right) \tag{4.1.1}$$

We compute PCS for all embedding models and report it in Table 4.1, averaged over relations of a category.

	Encyclopedic	Lexical	Semantic	Syntactic
e2e-ae-flan-t5-base (abelian)	0.52	0.85	0.64	0.94
e2e-ae-flan-t5-base (arithmetic)	0.52	0.85	0.64	0.94
e2e-ae-flan-t5-base (ff)	0.51	0.57	0.51	0.57
e2e-ae-flan-t5-base (mean)	0.52	0.79	0.62	0.72
ae-flan-t5-base	0.52	0.81	0.58	0.80
flan-t5-base	0.55	0.83	0.62	0.86
deberta-v3-base	0.51	0.68	0.58	0.65
deberta-base	0.58	0.83	0.67	0.89
roberta-base	0.54	0.83	0.66	0.87
bert-base-uncased	0.62	0.84	0.63	0.81
instructor	0.74	0.83	0.62	0.80
all-mpnet-base-v2	0.82	0.80	0.61	0.74
fasttext	0.57	0.86	0.62	0.87

Table 4.1. Pairing consistency scores by model and category

Higher is better and 0.5 is chance level. Given for the test split as the end-to-end models are trained.

It can be seen that most models have a much higher than chance PCS for the lexical and syntactic categories of relations, with the exception of the feedforward end-to-end model and DeBERTa-V3. Indeed, the former severely underfit during training (and otherwise overfit if not for model selection on the validation set), and could not solve even a single training analogy. However, DeBERTa-V3 is a state-of-the-art model differing only notably by its replaced token detection task rather than MLM or contrastive learning. This may be a hint that linear regularity is not to be expected, and is perhaps only a side effect of fitting parameters to a particular objective. Otherwise, it appears no embedding space captures the semantic relations in a linear fashion, and the only ones that capture the encyclopedic ones are Sentence-BERT (all-mpnet-base-v2), and, to a lesser extent, the Instructor model. Notably, both are trained on a contrastive loss objective including pairs from Wikipedia and question-answer tasks, whereas it has been advanced by Ri et al. (2023) that contrastive loss elicits parallel offset dimensions for analogical quadruples.

4.1.2. Candidate sets

	Encyclopedic		Lexical		Semantic		Syntactic	
	$+(a,b,c)$	Both	$+(a,b,c)$	Both	$+(a,b,c)$	Both	$+(a,b,c)$	Both
e2e-ae-flan-t5-base	0.03	0.02	0.28	0.27	0.02	0.02	0.30	0.28
ae-flan-t5-base	0.01	0.01	0.20	0.19	0.02	0.02	0.28	0.22
flan-t5-base	0.03	0.01	0.20	0.19	0.02	0.01	0.37	0.24
deberta-v3-base	0.01	0.01	0.11	0.08	0.02	0.02	0.08	0.06
deberta-base	0.04	0.02	0.16	0.15	0.02	0.01	0.12	0.09
roberta-base	0.03	0.01	0.28	0.26	0.08	0.04	0.23	0.19
bert-base-uncased	0.05	0.03	0.13	0.12	0.01	0.01	0.12	0.07
instructor	0.07	0.01	0.25	0.17	0.02	0.01	0.04	0.02
all-mpnet-base-v2	0.12	0.01	0.13	0.10	0.01	0.01	0.03	0.02
fasttext	0.03	0.00	0.40	0.00	0.03	0.00	0.46	0.00

Table 4.2. SATS test accuracy by category using the Abelian solver

Shown for candidate sets $+(a,b,c)$: with premises included, and Both: with distractors and premises.

We show a subset of accuracies in Table 4.2. Accuracies broken down by solver type, model, and choice of candidate set are shown in Table 4.8. A full breakdown of accuracies are shown in Figures A.1 to A.4. It appears at once that the Abelian and arithmetic solvers perform better than the feedforward one. Indeed, when premises are excluded, they appear to achieve remarkable accuracy (see Table 4.8). However, as is expected, the inclusion of distractors drastically lowers this accuracy, which is generally lowerbounded by simply including the premises. The inclusion of both, naturally, shows that very few models if any at all can be said to solve our analogies. In Chapter 2 we pointed out that our encyclopedic sentences are much longer than those of other relations (see Table 2.3). The inclusion of distractors then appears to make retrieval impossible for encyclopedic relations. However, a quick examination of Sentence-BERT's top retrieval candidates shows that c ranks highest, and when including distractors, it is c's distractors that rank highest, rather than d's. We show one such example in Table 4.4 when retrieving the solution to an encyclopedic analogy using Sentence-BERT. We observe similar behaviour in all models except DeBERTa-V3, for which the predicted vector retrieves nonsensical alternatives (see Table 4.6).

4.1.3. Pair and offset similarity

The obtained accuracies can be compared to those from the simple baseline where, for each base SATS pair (a,b), we try to retrieve b from a among all unique sentences in SATS, excluding a from candidates since it will always be nearest itself. We report this baseline in Table 4.3, finding similar issues as brought up by Linzen (2016) for word analogies. Indeed the nearest neighbour to a is b in 76% of the cases for Sentence-BERT across all categories, 56% for the Abelian E2E-AE-Flan-T5, and 48% for RoBERTa, none too far from the best accuracies they show when solving analogies. Even when including distractors, these nearest neighbour rates are on the order of retrieval accuracies, and often surpass them if we compare the baseline with the accuracies for the Abelian solver in Table 4.2.

	Encyclopedic		Lexical		Semantic		Syntactic	
	–	+distr.	–	+distr.	–	+distr.	–	+distr.
e2e-ae-flan-t5-base (Abelian)	0.00	0.00	0.84	0.08	0.44	0.00	0.97	0.14
e2e-ae-flan-t5-base (arithmetic)	0.00	0.00	0.84	0.10	0.46	0.00	0.96	0.16
e2e-ae-flan-t5-base (feedforward)	0.00	0.00	0.02	0.00	0.02	0.00	0.11	0.04
e2e-ae-flan-t5-base (mean)	0.02	0.00	0.80	0.09	0.40	0.00	0.48	0.12
ae-flan-t5-base	0.00	0.00	0.82	0.08	0.39	0.01	0.84	0.05
flan-t5-base	0.02	0.00	0.82	0.11	0.42	0.01	0.84	0.16
deberta-v3-base	0.01	0.00	0.57	0.27	0.20	0.08	0.57	0.26
deberta-base	0.00	0.00	0.82	0.21	0.40	0.01	0.86	0.15
roberta-base	0.01	0.00	0.76	0.14	0.30	0.02	0.88	0.20
bert-base-uncased	0.02	0.00	0.80	0.12	0.36	0.01	0.81	0.18
instructor	0.22	0.00	0.92	0.09	0.78	0.00	0.99	0.18
all-mpnet-base-v2	0.46	0.00	0.88	0.12	0.76	0.00	0.96	0.14
fasttext	0.02	0.00	0.75	0.00	0.32	0.00	0.76	0.04

Table 4.3. Nearest neighbour retrieval baseline on SATS test split (a,b) pairs

Shown additionally for added distractor candidates.

Evidently, the cosine similarity of a SATS pair (a,b) has a disproportionate effect on whether we can retrieve the solutions to our analogies by this method, and it appears to grow in tandem with word overlap, which we visualize in Figure 4.1. As shown by Fournier et al. (2020), the analogy score $sim(d,\, c+b-a)$ can be decomposed into terms proportional to the within-pair similarity $sim(a,b)$, the offset similarity $sim(b-a,\, d-c)$, and the similarity of c to the offset, which are subject to spurious properties of the geometry of the embedding used. They introduce an offset concentration score (OCS), i.e. the average similarity of offsets of a same relation. We show in Figure 4.2 that as accuracy for a relation increases, the product of the average within-pair similarity and OCS increases, a behaviour we observe equally for the arithmetic solver as for the Abelian one.

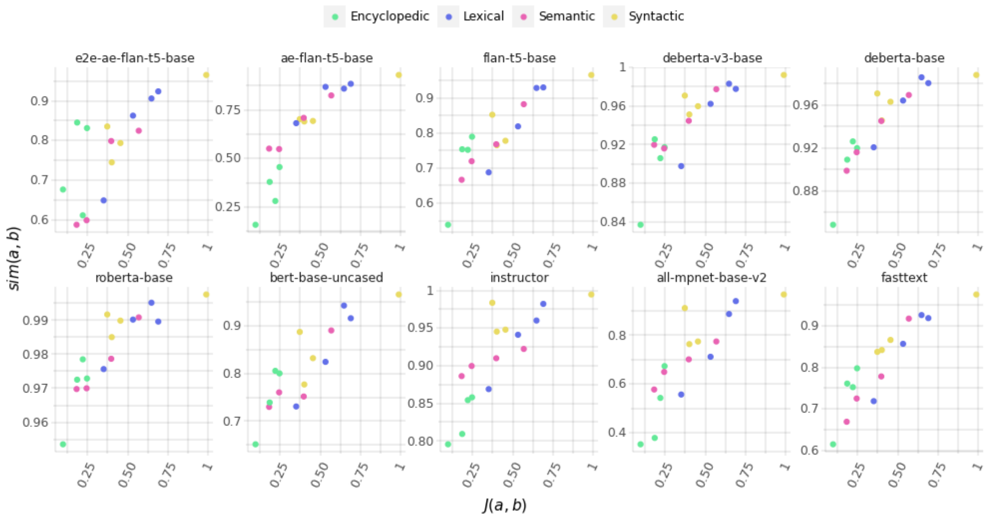

Figure 4.1. Within-pair similarity versus Jaccard similarity, averaged per SATS test relation

The arithmetic solver variant of E2E-Flan-T5-Base is used.

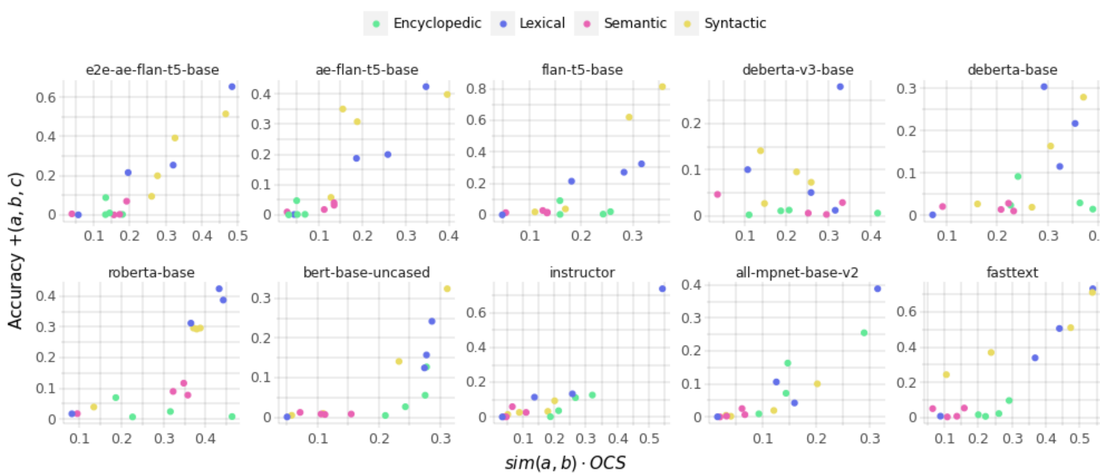

Figure 4.2. Arithmetic solver retrieval accuracy with premises included versus the product of average within-pair similarity and OCS per SATS test relation

We report accuracy when candidates include premises but not distractors so as not to zero out all encyclopedic relations.

4.1.4. Takeaways

We can conclude in agreement with Fournier et al. (2020) that the offset method fails to solve analogies, and that insofar as it succeeds, it is only due to particularities in the vector space. Indeed, in most cases the predicted vector is closest to c or surface edits thereof (see Tables 4.4 and 4.7). While accuracy increases along with pair similarity and offset similarity, we can discount the former as a meaningful factor if we're interested in solving analogies using parallel lines, and we can throw out the latter wholesale: OCS can be trivially high if one "side" of the relation is embedded in a smaller neighbourhood than the other, or if we increase the distance between equally sized neighbourhoods.

It is less obvious why we should not notice any meaningful difference in accuracy between the vector offset method and the Abelian solver. We report accuracies on all data splits in Table 4.5, wherein it can be seen that for most models, the Abelian solver severely underfits the training set, obtaining low accuracies when including premises and distractors, whereas the feedforward solver simply does not generalize despite generally fitting the training set. It

a : b :: c : d

Gene Simmons (born Chaim Witz; Hebrew: (...); born August 25, 1949) is an Israeli-American musician, singer and songwriter. **:** Kiss (stylized as ...) is an American rock band formed in New York City in 1973 by Paul Stanley, Gene Simmons, Ace Frehley, and Peter Criss. **::** Bradford Phillip Delson (born December 1, 1977) is an American musician, best known as the lead guitarist and one of the founding members of the rock band Linkin Park. **:** Linkin Park is an American rock band from Agoura Hills, California.

Neither	+(a,b,c)	+(a,b,c) +distractors
Linkin Park is an American rock band from Agoura Hills, California.	Bradford Phillip Delson (born December 1, 1977) is an American musician, best known as the lead guitarist and one of the founding members of the rock band Linkin Park.	Bradford Phillip Delson 1, 1977) is an musician, best known as the lead guitarist and one of the founding of the rock band Linkin Park.
Def Leppard are an English rock band formed in 1977 in Sheffield.	Kiss (stylized as ...) is an American rock band formed in New York City in 1973 by Paul Stanley, Gene Simmons, Ace Frehley, and Peter Criss.	Bradford Phillip Delson (born December 1, 1977) is an American musician, best known of the lead guitarist and one as the founding members of the rock band Linkin Park.

Table 4.4. Example analogy and top retrieved solutions under different candidate sets for Sentence-BERT using the arithmetic solver

The example analogy is drawn from the `member-band` relation, which is part of the test split. When including the premises as candidates, c followed by b are more similar to the offset prediction than the true solution. When including distractors, the top predictions are all similar to c. Special characters are replaced with an ellipsis due to typesetting issues.

is unclear what better kind of hyperparameter selection could alleviate this. Including distractors when evaluating on the validation set, however, ought to be fruitful. Nevertheless, for the two models where the Abelian solver mostly fit the training set (Flan-T5 and Instructor), their validation and test accuracies are no more impressive than the others. While it is possible that it ails mainly from a lack of data—a conclusion supported by the feedforward solver's lack of generalization—it may be due to a property of Abelian groups themselves for the purpose of solving proportional analogies of this nature.

	Training			Validation			Test		
	Abelian	Arithmetic	FF	Abelian	Arithmetic	FF	Abelian	Arithmetic	FF
e2e-ae-flan-t5-base	0.24	0.25	0.00	0.17	0.17	0.00	0.15	0.14	0.00
ae-flan-t5-base	0.71	0.05	1.00	0.12	0.10	0.04	0.11	0.08	0.04
flan-t5-base	0.97	0.12	0.90	0.12	0.07	0.06	0.11	0.08	0.07
deberta-v3-base	0.04	0.07	0.39	0.02	0.05	0.00	0.04	0.06	0.00
deberta-base	0.08	0.08	0.61	0.04	0.04	0.02	0.07	0.07	0.03
roberta-base	0.12	0.11	0.87	0.13	0.12	0.07	0.13	0.12	0.08
bert-base-uncased	0.08	0.06	0.90	0.03	0.03	0.05	0.06	0.05	0.08
instructor	0.82	0.07	1.00	0.12	0.12	0.01	0.05	0.05	0.01
all-mpnet-base-v2	0.24	0.03	0.64	0.05	0.06	0.02	0.03	0.03	0.02
fasttext	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4.5. Retrieval accuracy per model on all SATS splits

Accuracy given for candidate set with premises and distractors.

Otherwise, it appears that the end-to-end solver not only performs best among all models, but equally whether using the Abelian or arithmetic solver. This is most likely due to both architectures ultimately performing the vector offset method on the mean-pooled encodings, while the parameters of the encoder are allowed to change. The addition of an extra transformation in the form of an invertible neural network appears to change little about this fact. A limitation of this model is that the autoencoder it is based on may have forgotten most long-range dependencies and representations that can be obtained from them, including of encyclopedic knowledge and discourse relations. Indeed, being a simple autoencoder, we expect that its bottleneck vector represents little other than surface features. An improvement might be to use an architecture such as the transformer sequential denoising autoencoder (TSDAE) of Wang et al. (2021), which, by using a denoising and masked language modeling objective, likely learns a representation based in distributional semantics.

Finally, performance at this task appears to us to have little correlation with the capability of a model, since the embeddings obtained from state-of-the-art models (though mean-pooling may lose information for many of them) are outperformed by the end-to-end model. This, while it performs best, is best only marginally, for a handful of relations where pairs have sizeable word overlap, and subject to all the issues we have discussed above.

a : b :: c : d		
Four people stand on a flimsy wooden plank. : The wooden plank breaks. :: There are very loud noises outside while you try to sleep. : You sleep poorly.		
Neither	+(a,b,c)	+(a,b,c) +distractors
If recommit fails , then what choice needs to be made by minority representatives ?	If recommit fails , then what choice needs to be made by minority representatives ?	There are very loud noises while outside you try to sleep.
Advertisement is effective so this flavor is popular.	Advertisement is effective so this flavor is popular.	The person who won that prize last year.It is speaking.
Your piping is being repaired by the plumbers.	Your piping is being repaired by the plumbers.	When were the deported allowed to return ?
How many people arrived without an invitation ?	How many people arrived without an invitation ?	If recommit fails , then what choice needs to be made by minority representatives ?
They all ordered food for delivery.	They all ordered food for delivery.	.They all ordered food for delivery.

Table 4.6. Top 5 retrieved solutions under different candidate sets for Deberta-V3-Base using the arithmetic solver

The example analogy is drawn from the `cause-effect` relation, which is part of the test split. Candidates are noticeably irrelevant in all cases, hinting at DeBERTa-V3's particular embedding geometry, although when distractors are additionally included one corresponding to c ranks first.

a : b :: c : d		
There's a loud noise and you flinch. : You are startled by a noise. :: A family huddles on a couch with popcorn and turn on the TV. : A family is watching a movie together.		
Neither	+(a,b,c)	+(a,b,c) +distractors
A family is watching a movie together.	A family huddles on a couch with popcorn and turn on the TV.	A family huddles on a couch with popcorn and turn on the TV.
An air conditioner is turned on in a hot room.	You are startled by a noise.	a family huddles on A couch with popcorn and turn on the TV.
A cloud of steam rises off a cup of water.	A family is watching a movie together.	6.A family huddles on a couch with popcorn and turn On the TV.
They gather on the couch to watch a movie every weekend.	An air conditioner is turned on in a hot room.	A family huddles on a couch with popcorn on the TV.
The fire was started by a cigarette.	A cloud of steam rises off a cup of water.	A family huddles on a couch turn popcorn and with on the TV.

Table 4.7. Top 5 retrieved solutions under different candidate sets for the end-to-end arithmetic solver Flan-T5 model

The example analogy is drawn from the `description-state` relation, which is part of the test split. When including the premises as candidates, c followed by b are more similar to the offset vector than the solution. After including distractors, though, the next top-4 candidates are distractors for c.

Solver type	Model	Neither	+distractors	+(a,b,c)	Both
Abelian	e2e-ae-flan-t5-base	0.60	0.25	0.16	0.15
	ae-flan-t5-base	0.51	0.20	0.13	0.11
	flan-t5-base	0.59	0.23	0.16	0.11
	deberta-v3-base	0.20	0.12	0.06	0.04
	deberta-base	0.59	0.21	0.08	0.07
	roberta-base	0.57	0.27	0.15	0.13
	bert-base-uncased	0.61	0.17	0.08	0.06
	instructor	0.74	0.13	0.09	0.05
	all-mpnet-base-v2	0.82	0.11	0.07	0.03
	fasttext	0.44	0.00	0.23	0.00
Arithmetic	e2e-ae-flan-t5-base	0.60	0.25	0.15	0.14
	ae-flan-t5-base	0.54	0.17	0.09	0.08
	flan-t5-base	0.61	0.21	0.10	0.08
	deberta-v3-base	0.34	0.20	0.08	0.06
	deberta-base	0.59	0.20	0.08	0.07
	roberta-base	0.57	0.27	0.15	0.12
	bert-base-uncased	0.61	0.16	0.07	0.05
	instructor	0.80	0.13	0.08	0.05
	all-mpnet-base-v2	0.82	0.10	0.07	0.03
	fasttext	0.57	0.00	0.21	0.00
FF	e2e-ae-flan-t5-base	0.00	0.00	0.00	0.00
	ae-flan-t5-base	0.27	0.06	0.07	0.04
	flan-t5-base	0.36	0.10	0.15	0.07
	deberta-v3-base	0.01	0.00	0.01	0.00
	deberta-base	0.22	0.03	0.15	0.03
	roberta-base	0.23	0.12	0.12	0.08
	bert-base-uncased	0.32	0.12	0.13	0.08
	instructor	0.38	0.01	0.09	0.01
	all-mpnet-base-v2	0.50	0.04	0.15	0.02
	fasttext	0.01	0.00	0.01	0.00
Mean	e2e-ae-flan-t5-base	0.18	0.00	0.00	0.00
	ae-flan-t5-base	0.23	0.00	0.00	0.00
	flan-t5-base	0.19	0.00	0.00	0.00
	deberta-v3-base	0.05	0.01	0.00	0.00
	deberta-base	0.17	0.00	0.00	0.00
	roberta-base	0.15	0.00	0.00	0.00
	bert-base-uncased	0.21	0.00	0.00	0.00
	instructor	0.20	0.00	0.00	0.00
	all-mpnet-base-v2	0.28	0.00	0.00	0.00
	fasttext	0.13	0.00	0.00	0.00

Table 4.8. SATS test retrieval accuracies under different candidate sets, by solver type and model

4.2. Generation

We generate predictions for our proportional sentence analogies using the methods described in Section 3.5, for our end-to-end solver decoder (E2E-AE-Flan-T5), the sequence-to-sequence finetuned (so-called Solver and Generator) Flan-T5 models, and the pretrained Flan-T5 from sizes Base to XXL, which is prompted with few-shot examples to solve proportional analogies.

4.2.1. Results

		Exact Match		WER		METEOR		BLEU	
	Sampled	False	True	False	True	False	True	False	True
(a,b) Baseline		—		0.86		0.54		0.23	
Generator	Base	0.00	0.00	1.39	1.49	0.36	0.26	0.10	0.07
	Large	0.01	0.00	1.08	1.06	0.47	0.41	0.17	0.14
Solver	Base	0.01	0.00	1.14	1.10	0.36	0.31	0.12	0.10
	Large	0.02	0.01	0.88	0.92	0.48	0.42	0.20	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
Arithmetic	Base	0.04	0.03	1.00	0.92	0.40	0.38	0.15	0.14

Table 4.9. SATS test split generation metrics

Scores are shown in table Table 4.9. As shown in Section 4.1, the end-to-end solver behaves identically with either the arithmetic or Abelian solver, and outright fails with the feedforward one. Since this holds equally in the generative setting, we only report results for the arithmetic solver for brevity. In addition to the solutions generated by models by greedy decoding and η-sampling, we report metrics for the baseline where, for each base SATS pair (a,b), a as the prediction for b. We do so because, as noted in Chapter 2 and elsewhere, our relational pairs tend to be extremely similar in form. Thus, it can be difficult to distinguish predictions that are noisy but otherwise valid solutions to the analogy from predictions which are similar to one of the premises, though those which are otherwise unrelated to the solution should score poorly regardless. Success might be indicated by better metrics than this baseline.

From the metrics shown in Table 4.9, it is evident that our methods perform poorly at the task, with some variation, which we can note. The end-to-end vector solver performs best by exact match rate, though doesn't have the overall best metrics. This is not all too surprising, as the architectural bias of the vector offset solver should behave favorably for relations where, as discussed previously, embeddings show higher offset similarity and within-pair similarity, though the vector predicted by offset may not necessarily be understood by the decoder, even if it results in good retrieval accuracy. The few-shot results for Flan-T5-XXL, even with its 11 billion parameters, show that it achieves somewhat less than this level.

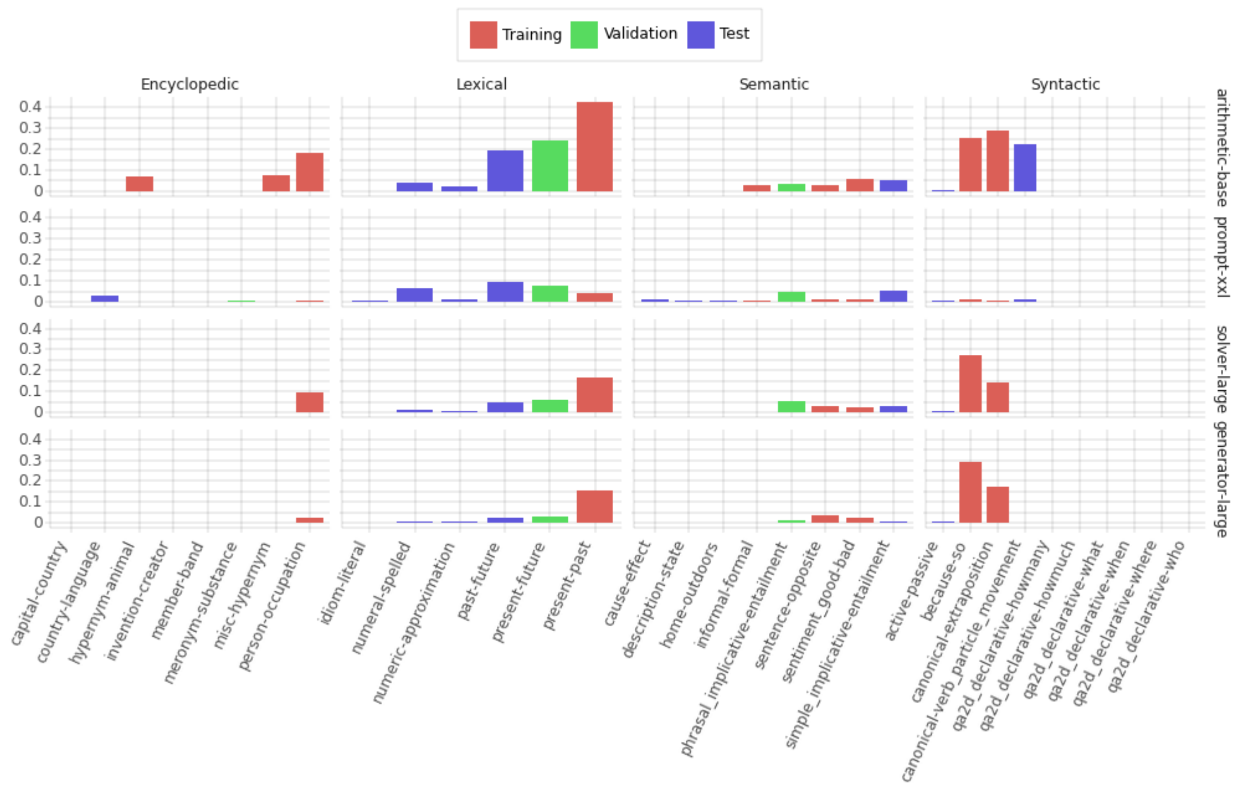

Figure 4.3. Exact match accuracy comparison of best models

Between the finetuned sequence-to-sequence models, the Solver model, which predicts d prompted with c, outperforms the one trained to generate the whole (c,d) pair, though this difference narrows for the Large models. We would hope that a model capable of generating a pair of sentences analogical to an input pair should complete d well, but this expectation is dashed by the overall poor results seen even on the training split, demonstrated in Figure 4.3. Indeed, no trained model appears to have captured the training data very well. It is a significant limitation of this work that the inherent data duplication resulting from analogies constructed by shuffling pairs results in overfitting. Thus, our models in fact underfit the task when we select the best checkpoint. The prompted Flan-T5's performance increases with parameter size. At its largest range, the Flan-T5-XXL checkpoint (11B parameters) and reaches an overall performance on the level of the end-to-end solver. Encouragingly, it reaches albeit low accuracy across most relations, rather than it being concentrated as much. Figure 4.4 shows the scale of exact match compared to copy rates on the test split.

		Copy a Rate		Copy b Rate		Copy c Rate	
	Sampled	False	True	False	True	False	True
Generator	Base	0.00	0.00	0.00	0.00	0.20	0.06
	Large	0.00	0.00	0.00	0.00	0.30	0.15
Solver	Base	0.04	0.01	0.03	0.01	0.13	0.06
	Large	0.00	0.00	0.01	0.00	0.27	0.14
Prompted	Base	—	0.01	—	0.02	—	0.07
	Large	—	0.01	—	0.01	—	0.24
	XL	—	0.05	—	0.06	—	0.09
	XXL	—	0.02	—	0.05	—	0.06
Arithmetic	Base	0.00	0.00	0.06	0.05	0.03	0.02

Table 4.10. SATS test split copy rates

Copying c appears to peak in the double digits for all "full-attention" Flan-T5 models at around 20% for all Large models (see Table 4.10), though increasing parameter size reduces this once more in the few-shot setting. Interestingly, the the XL checkpoint sees a peak in the copy rate for encyclopedic relations, in a way that mimics the Large checkpoint for other categories (see the breakdown per category in Table 4.11). Perhaps there are parameter thresholds at which the model has enough capacity to capture the long-range structure of the prompt, depending on the length of the input sequence—since the longest encyclopedic analogies may have hundreds of words—though lacks the ability to solve them.

	Encyclopedic		Lexical		Semantic		Syntactic	
	Exact Match	Copy	Exact Match	Copy	Exact Match	Copy	Exact Match	Copy
Base	0.001	0.108	0.001	0.146	0.001	0.130	0.000	0.071
Large	0.002	0.200	0.007	0.395	0.005	0.308	0.001	0.176
XL	0.007	0.356	0.017	0.211	0.008	0.170	0.001	0.149
XXL	0.006	0.210	0.049	0.101	0.019	0.070	0.003	0.100

Table 4.11. Few-shot exact match and summed copy rates for different Flan-T5 model sizes, per relation category

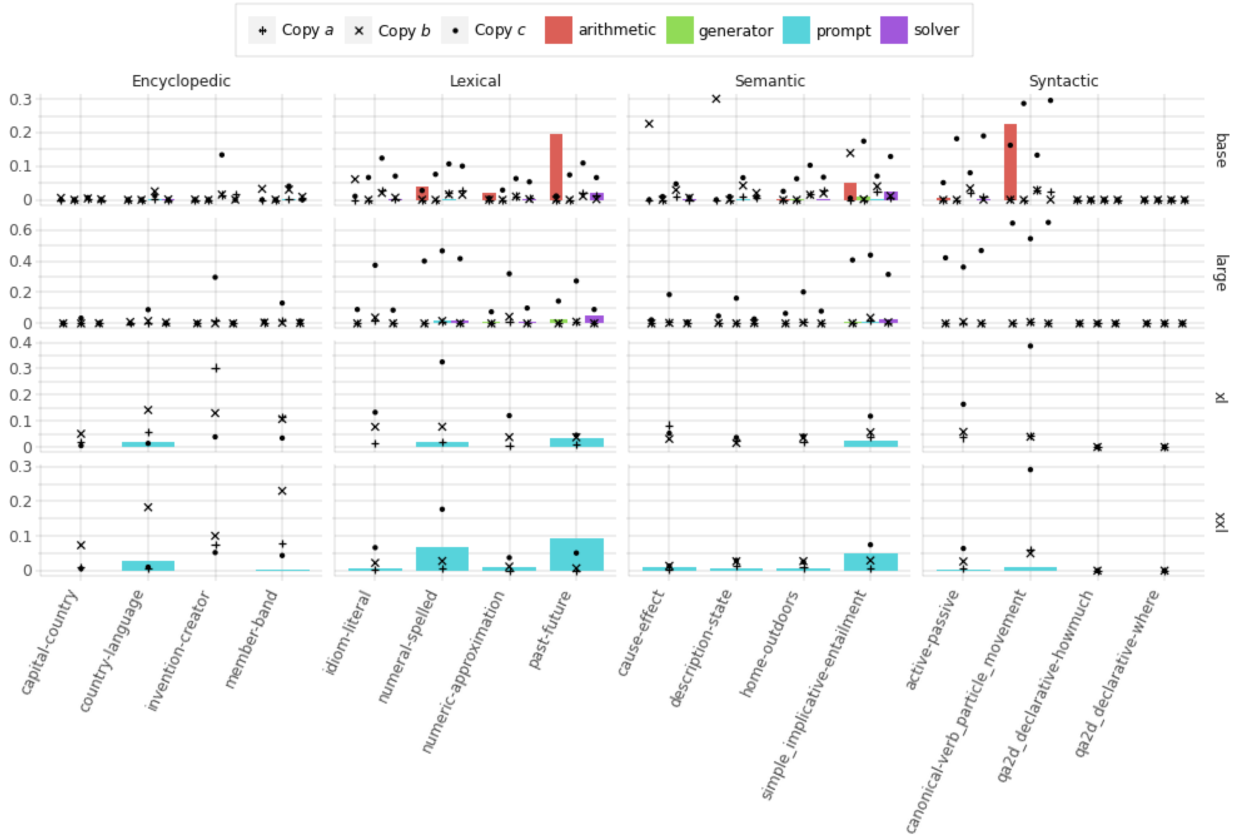

Figure 4.4. Test exact match accuracy per model by parameter size, copy rates overlaid

4.2.2. Limitations

We can identify some shortcomings of our experimental and data processing methodology. As can be seen in Figures 4.3 and 4.4, we get zero exact match and copy rates for the QA2D relations. It is an unfortunate consequence of not preprocessing the formatting of those sentences, that they have artifacts, and that all tokens, punctuation included, are whitespaced. We can say the same for the encyclopedic sentences found from Wikipedia, which have substantial formatting artifacts. We would expect successful models to treat these somewhat like noise, or, if they attempt to emulate the artifacts, fail to perfectly copy them. For similar reasons, we should expect the exact match rate to reach zeros for encyclopedic relations as well. However, as noted in Chapter 2, all but two of those relations, `person-occupation` and `capital-country`, are many-to-one (see Table 2.2). Subsequently, a high b copy rate should artificially inflate this metric. For Flan-T5-XL and XXL on the encyclopedic set, the copy b rate is 12.3% and 13.6% respectively, whereas for the Base and Large models it is 2.6% and 0.5% respectively.

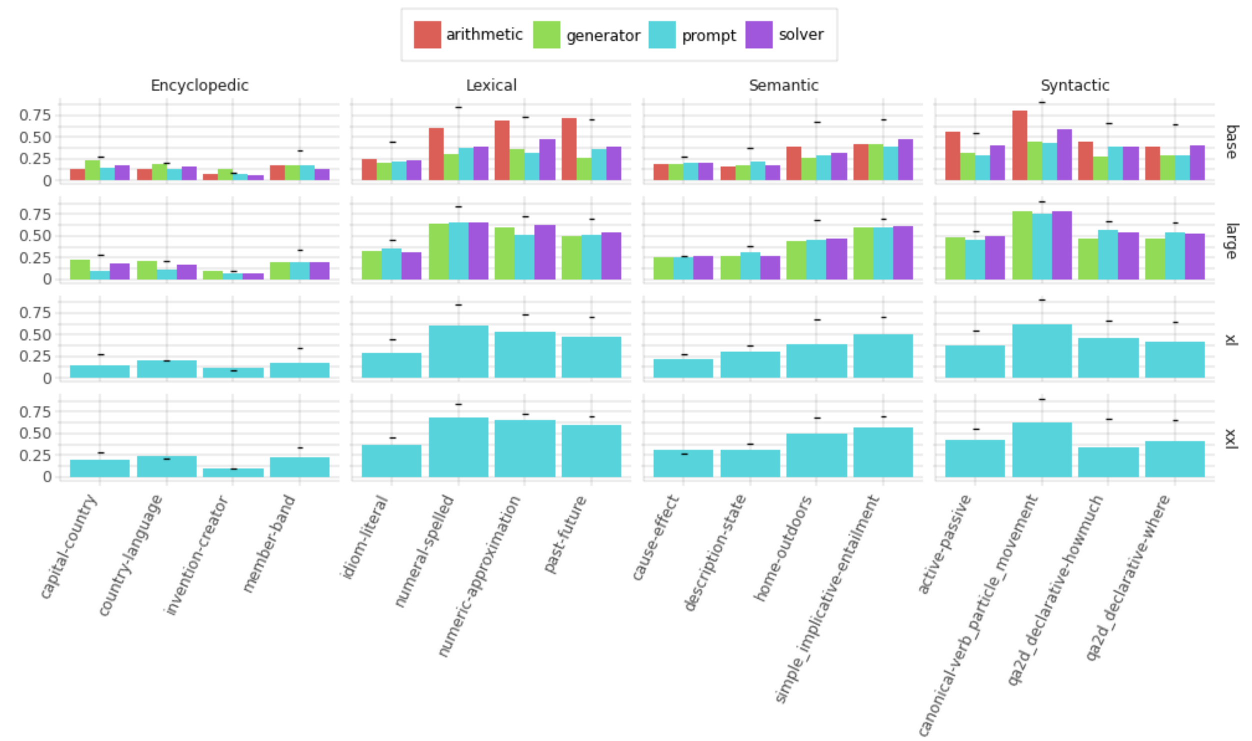

Figure 4.5. Test METEOR (higher is better) per model by parameter size with METEOR of (a,b) pair baseline overlaid as a dash

We plot METEOR in Figure 4.5 in comparison to the (a,b) baseline. In virtually all cases, we can see that our predictions are substantially worse than the pair baseline. Combined with copy rates shown elsewhere, it appears our models are generally not solving our analogies. Without fully evaluating by human judgement, examining some examples can provide some clarity, even if not representative of all cases. We can see from a few selected examples in Table 4.14 that our arithmetic end-to-end solver often decodes almost perfectly memorized training examples, or else nonsense which is somewhat reduced by sampling. In one analogy from the `qa2d-declarative-howmuch` relation, it appears to approach a reasonable solution, though both sentence pairs are regarding electricity production in Estonia. Even in this hopeful case, it does not maintain those parts of sentence c which should not be modified, an important element of the relation.

In Table 4.12 we can see that the full output from Flan-T5-XXL in response to the prompt, before splitting the solution, includes an explanation acknowledging the relation in question. Ultimately, the output repeats the separator and repeats a sequence similar to c, and hence fails the analogy test. For comparison, we used the OpenAI Platform to access the davinci-002[1] and ChatGPT-3.5 models, using the same prompt in Figure 3.4

[1]Part of the updated legacy models in 2023, at https://platform.openai.com/.

(see Table 4.13). ChatGPT, which benefits from instruction training with human feedback (Ouyang et al., 2022), answers appropriately, though without using the desired separator, whereas davinci-002 copies *b* approximately in one case and exactly in the other. This highlights two things: that prompting alone may not be a successful method, and that our metrics may fail to identify cases where it is.

capital-country						
Lisbon [...] is the capital and the largest [...]	:	Portugal, officially the Portuguese Republic [...]	::	Manila [...] is the capital of the Philippines	:	The Philippines [...] is an archipelagic country in Southeast Asia.
(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						
qa2d-declarative-what						
Greenpeace claimed that Shell requested Google to ban a video footage [...]	:	Greenpeace claimed that Shell requested Google to ban what ?	::	"Mauri" is Moors in English	:	What is Mauri in English ?
(FINAL ANSWER) The subject has been removed from the sentence and replaced with what. (FINAL ANSWER) Mauri.						

Table 4.12. Flan-T5-XXL analogy examples before splitting by the answer separator

person-occupation						
John Christopher Depp [...] is an American actor [...]	:	An actor or actress is a person who portrays a character in a performance.	::	Christopher Columbus [...] was an Italian[b] explorer [...]	:	An explorer is a person involved in exploration
(ChatGPT-3.5) An explorer or navigator is a person who completes voyages across the Atlantic Ocean, sponsored by monarchs, opening the way for widespread exploration and colonization of the Americas.						
(davinci-002) (FINAL ANSWER) An actor or actress is a person who portrays a character in a performance. They may have used stage makeup to enhance their natural appearance or wear a mask to hide all traces of their identity. (Changed from Christopher Columbus to actor and female actresses.) Join the mailing list: The Good Riddles Mission						
qa2d-declarative-what						
The website , thetime.co.uk , is aimed at daily readers .	:	The website , thetimes.co.uk , is aimed at what kind of reader ?	::	Metaphysics , epistemology , axiology and the philosophical approaches make up field education and a field applied philosophy .	:	What fields make up field education and a field applied philosophy ?
(ChatGPT-3.5) What makes up the field of education and applied philosophy?						
(davinci-002) (FINAL ANSWER) The website , thetimes.co.uk , is aimed at what kind of readers ?						

Table 4.13. davinci-002 and ChatGPT-3.5 analogy examples

We manually extracted the generated solutions from OpenAI Playground, using default hyperparameters, and the ChatGPT UI. We use the same prompt as shown in Figure 3.4. The ChatGPT solution was explicitly presented as such, surrounded by reasoning steps and descriptions related to the task.

qa2d-declarative-howmuch						
1.562 billion kilowatt hours of electricity is exported by Estonia annually .	:	How much electricity is exported by Estonia annually ?	::	75 % of consumed electricity is produced by Estonia itself .	:	How much of consumed electricity does Estonia produce by itself ?
(Greedy decoding) How much electricity is generated by Estonia?						
(Sampled) How much electricity does Estonia produce annually by?						
member-band						
Kurt Donald Cobain (1967–1994) was an American singer, songwriter and artist.	:	Nirvana was an American rock band formed in Aberdeen, Washington, in 1987.	::	Paul David Hewson [...] is an Irish singer-songwriter, activist, and philanthropist.	:	U2 are an Irish rock band from Dublin, formed in 1976.
(Greedy decoding) In Aberdeen, Aberdeen formed a rock group formed in the Aberdeen rock group Aberdeen, Aberdeen, Aberdeen, Aberdeen, Aberdeen, [...]						
(Sampled) Aberdeen formed a rock group formed in Aberdeen, Aberdeen's Aberdeen in mid-nineteenth century AD, which formed the Rockies.						
capital-country						
Taipei [...] is the capital[a] and a special municipality of Taiwan.	:	Taiwan [...] is a country[22] in East Asia, at the junction [...]	::	Berlin [...] is the capital and largest city of Germany [...]	:	Germany [...] is a country in Central Europe.
(Greedy decoding) A physicist is a scientist who specializes in the field of physics, which encompasses the interactions of matter and energy at all length and time scales in the universe.						
(Sampled) An author is someone who writes music., which incorporates many elements of their work, typically from the heart, each of its constituent parts, from the heart, and the heartâsest part, from the heart to the back.						

Table 4.14. Arithmetic E2E-Flan-T5-Base analogy examples

Chapter 5

Conclusion

In this work, we studied a number of methods for solving proportional analogies between sentences. Leveraging pairs of sentences between which a certain relation holds, we introduce a manually curated sentence analogy test set. This dataset is composed of relations of syntax, semantics, style, and world knowledge. We experiment with some conventional methods for solving proportional analogies by retrieval in a vector space model using embeddings, distributed vector representations of text which we derive largely from pretrained language models, inspired by their previous usage in word analogies, and their continued use in recent times for sentence analogies. We further study the adequacy of conventional methods by decoding the solutions to proportional analogies from those vector representations rather than retrieving them in a VSM. We do so juxtaposed against the relatively novel frameworks of solving analogies as a sequence-to-sequence task and of few-shot prompting with examples.

Several limitations of our methodology can be traced to the nature of our data, affecting the adequacy of our models and the strength of our analysis. In light of this, the first improvement should be of the quality of data by the expansion of the relations it contains and increased diversity of its content. Many datasets of pairs of text sequences exist which can be combined together, and several novel analogical datasets have recently been proposed. While further work could incorporate hyperparameter search, coupled with more data, to ensure current methods are not in fact sufficient, we find it may be more fruitful to explore different model architectures and features to include. Some architectural biases could be favorable to the task, such as the Offset Network of Mao and Lepage (2023).

We find, in agreement with previous studies of word analogies using vector representations (Linzen, 2016; Rogers et al., 2017), that the vector offset method does not recover the solutions to sentence analogies, instead often recovering the premises themselves, unless both the pairs' vectors and the offsets between pairs are respectively similar, which it has been shown can happen for spurious reasons (Fournier et al., 2020). It does not either appear desirable to us to enforce a vector representation which obeys these properties, likely at the cost of the goodness of its representation. We note this most saliently by the utter failure of the embeddings obtained from DeBERTa-v3, which is otherwise an impressive state-of-the-art model. Retrieval when solving analogies in vector space remains at low accuracy even when relations for a given embedding have a high pairing consistency score, which measures the linear distinguishability of pairs' offsets. This latter score also being low for DeBERTa-v3, it appears that whether or not a representation encodes any regularities linearly is a matter of its particular architecture and training objective.

Ultimately, it appears to us that this task often reduces to measuring the clustering of related pairs. We find similar results comparing the vector offset method with an operation parameterized as an Abelian group network (Abe et al., 2021), which has been shown to be a universal approximator of Abelian Lie groups, indicating the relevance of an operation's algebraic properties to the task as argued by Rogers et al. (2017) and discussed in Section 1.4.4. On the other hand, training a feedforward network to the task, we find it overfits and also suffers from retrieving premises, which may be a consequence of our dataset, which is both limited and number and contains an adverse amount of duplicate data.

Separately, we might wish to write off the notion solving analogies in vector space. Outside of our usage of summed word embeddings and our bottleneck vector-based models, no important representations of language learned by recent models manifest in a single vector. Indeed, the most promising attempts at human-like in-text reasoning are those involving extremely large transformer language models, which represent inputs as a sequence of vectors. In our case, taking as a sentence embedding the average of these token representations likely leaves information on the table. We expect that a successful attempt at analogical reasoning in vector space in this fashion would require an immense amount of appropriate proportional analogy data and substantial computation, all for the purpose of learning the embedding geometry defined by a specific model's distribution of mean-pooled embeddings (or as defined by other pooling methods). This seems to us counterproductive. If we are interested in solving analogies, whether proportional or not, which can be based on semantics, style, or other "fuzzy" characteristics, then we should exploit models which have learned general patterns which may represent them.

Despite the promise of generation as a means of solving analogies, this vein of experimentation suffered from its own failures. We find in all cases that models copying the premise sentences is an issue. Our analysis of generated solutions also suffers from the difficulty of choosing an appropriate automatic evaluation metric for sequences which could be valid in their meaning despite differing in form. Nevertheless, we manage to find some insights by quantitative analysis. Ultimately, qualitative analysis may reveal deeper insights than previous automatic evaluation metrics, as the tasks being evaluated outpace them. Indeed, human evaluation has been used recently in many applications of LLMs due to the difficulties of automating their evaluation, a quandary whose solution may lie in none other than LLMs themselves (Chang et al., 2023).

Unfortunately, we find finetuning a pretrained language model to solve analogies as a sequence-to-sequence task to be unsuccessful, likely due to our small and degenerate training data used. This likely results in forgetting the distribution learned by pretraining. Prompting pretrained language models with up to 11 billion parameters to solve analogies in a few-shot setting in our experiments was overall unsuccessful and in aggregate less successful than the end-to-end solver. However, at the largest range of parameter sizes, we find a marked improvement where Flan-T5-XXL performs arguably on par with trained models, and qualitatively presents superior behavior. Thus, prompt engineering, prompt tuning, or other in-context approaches such as those of Patel et al. (2023) for bidirectional encoders could be most constructive, given the relatively promising results obtained for Flan-T5-XXL.

Finally, although it is intended only as a qualitative, exploratory, and especially informal experiment, the single analogy test given to the davinci-002 and ChatGPT-3.5 models offers some important insights for future research into solving proportional analogies. Despite its 175B parameters, davinci-002 falls into the same premise-copying trap as the smaller models we evaluated. In contrast, ChatGPT-3.5 provided essentially a perfect response. While the choice of prompt (and lack of engineering or learning thereof) is a confounding factor, it may point toward the preferential usage of similarly trained instruction-following models (Ouyang et al., 2022) to elicit responses that leverage learned abstract patterns. We would recommend evaluating the analogical reasoning capabilities of such models, especially publicly accessible ones which may appear in light of the release of human feedback data by Köpf et al. (2023). We expect this vein of research to demonstrate the most human-like analogical reasoning ability.

References

Abe, K., Maehara, T., and Sato, I. (2021). Abelian Neural Networks.

Afantenos, S., Lim, S., Prade, H., and Richard, G. (2022). Theoretical study and empirical investigation of sentence analogies. In Couceiro, M. and Murena, P.-A., editors, *IJCAI-ECAI Workshop: Workshop on the Interactions between Analogical Reasoning and Machine Learning (IAMRL 2022) @ IJCAI-ECAI 2022*, volume 3174 of *CEUR Proceedings*, pages 15–28, Vienna, Austria. CEUR-WS.org.

Airenti, G. (2019). The Place of Development in the History of Psychology and Cognitive Science. *Frontiers in Psychology*, 10:895.

Allen, C. and Hospedales, T. (2019). Analogies Explained: Towards Understanding Word Embeddings. *arXiv:1901.09813 [cs, stat]*.

Almarwani, N. and Diab, M. (2021). Discrete Cosine Transform as Universal Sentence Encoder. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 419–426, Online. Association for Computational Linguistics.

Alsaidi, S., Decker, A., Lay, P., Marquer, E., Murena, P.-A., and Couceiro, M. (2021). A Neural Approach for Detecting Morphological Analogies.

Ardizzone, L., Bungert, T., Draxler, F., Köthe, U., Kruse, J., Schmier, R., and Sorrenson, P. (2018). Framework for easily invertible architectures (FrEIA).

Ashworth, E. J. and D'Ettore, D. (2021). Medieval theories of analogy. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2021 edition.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Goldstein, J., Lavie, A., Lin, C.-Y., and Voss, C., editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Barbero, T. and Afantenos, S. D. (2023). Some preliminary results on analogies between sentences using contextual and non-contextual embeddings. In Couceiro, M., Afantenos, S. D., and Murena, P.-A., editors, *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*, pages 34–45. CEUR-WS.org.

Barbot, N., Miclet, L., and Prade, H. (2019). Analogy between concepts. *Artificial Intelligence*, 275:487–539.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(null):1137–1155.

Blank, D. S. (1997). *Learning to See Analogies: A Connectionist Exploration*. Indiana University.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2022). On the Opportunities and Risks of Foundation Models.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Brown, W. R. (1989). Two Traditions of Analogy. *Informal Logic*, 11(3).

Chalmers, D. J., French, R. M., and Hofstadter, D. R. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence*, 4(3):185–211.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2023). A survey on evaluation of large language models.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling Instruction-Finetuned Language Models.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Demszky, D., Guu, K., and Liang, P. (2018). Transforming Question Answering Datasets Into Natural Language Inference Datasets.

Dettmers, T., Lewis, M., Shleifer, S., and Zettlemoyer, L. (2022). 8-bit optimizers via block-wise quantization. *9th International Conference on Learning Representations, ICLR*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: Non-linear Independent Components Estimation.

Drozd, A., Gladkova, A., and Matsuoka, S. (2016). Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530, Osaka, Japan. The COLING 2016 Organizing Committee.

Ethayarajh, K. (2019). Rotate King to get Queen: Word Relationships as Orthogonal Transformations in Embedding Space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3501–3506, Hong Kong, China. Association for Computational Linguistics.

Ethayarajh, K., Duvenaud, D., and Hirst, G. (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.

Falkenhainer, B., Forbus, K. D., and Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63.

Firth, J. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32.

Fournier, L., Dupoux, E., and Dunbar, E. (2020). Analogies minus analogy test: Measuring regularities in word embeddings. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 365–375, Online. Association for Computational Linguistics.

French, R. M. (2002). The computational modeling of analogy-making. *Trends in Cognitive Sciences*, 6(5):200–205.

Gao, T., Yao, X., and Chen, D. (2022). SimCSE: Simple Contrastive Learning of Sentence Embeddings.

Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7(2):155–170.

Gentner, D. and Markman, A. (1992). Analogy– Watershed or Waterloo? Structural alignment and the development of connectionist models of analogy. In Hanson, S., Cowan, J., and Giles, C., editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.

Gick, M. L. and Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, 12(3):306–355.

Giorgi, J., Nitski, O., Wang, B., and Bader, G. (2020). Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*.

Gittens, A., Achlioptas, D., and Mahoney, M. W. (2017). Skip-Gram - Zipf + Uniform = Vector Additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Gladkova, A., Drozd, A., and Matsuoka, S. (2016). 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, pages 8–15, San Diego, California. Association for Computational Linguistics.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Guu, K., Hashimoto, T. B., Oren, Y., and Liang, P. (2018). Generating Sentences by Editing Prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Hadamard, J. (1945). *The Mathematician's Mind: The Psychology of Invention in the Mathematical Field*. Princeton University Press, Princeton, N.J.

Halford, G. S., Wiles, JANET., Humphreys, M. S., and Wilson, W. H. (1993). Parallel distributed processing approaches to creative reasoning: Tensor models of memory and analogy. *AI and Creativity*, pages 57–60.

Hanna, M. and Bojar, O. (2021). A fine-grained analysis of BERTScore. In Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-jussa, M. R., Federmann, C., Fishel, M., Fraser, A., Freitag, M., Graham, Y., Grundkiewicz, R., Guzman, P., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Kocmi, T., Martins, A., Morishita, M., and Monz, C., editors, *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

He, P., Liu, X., Gao, J., and Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention.

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. McGill University.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*.

Hewitt, J., Manning, C., and Liang, P. (2022). Truncation sampling as language model desmoothing. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoffmann, DT., Behrmann, N., Gall, J., Brox, T., and Noroozi, M. (2022). Ranking info noise contrastive estimation: Boosting contrastive learning via ranked positives. In *AAAI Conference on Artificial Intelligence*.

Hofstadter, D. R. (1995). *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought.* Basic books.

Hofstadter, D. R., Mitchell, M., et al. (1995). The copycat project: A model of mental fluidity and analogy-making. *Advances in connectionist and neural computation theory*, 2:205–267.

Holyoak, K. J. and Thagard, P. (1989). Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, 13(3):295–355.

Hu, X., Storks, S., Lewis, R., and Chai, J. (2023). In-Context Analogical Reasoning with Pre-Trained Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1953–1969, Toronto, Canada. Association for Computational Linguistics.

Huang, J. Y., Huang, K.-H., and Chang, K.-W. (2021). Disentangling Semantics and Syntax in Sentence Embeddings with Pre-trained Language Models.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Karttunen, L. (2012). Simple and phrasal implicatives. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 124–131, Montréal, Canada. Association for Computational Linguistics.

Keane, M. (2013). Deconstructing analogy. *CogSc-12 ILCLI International Workshop on Cognitive Science*.

Kim, T., Yoo, K. M., and Lee, S.-g. (2021). Self-Guided Contrastive Learning for BERT Sentence Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2528–2540, Online. Association for Computational Linguistics.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., ES, S., Suri, S., Glushkov, D., Dantuluri, A., Maguire, A., Schuhmann, C., Nguyen, H., and Mattick, A. (2023). OpenAssistant conversations – democratizing large language model alignment.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Blanco, E. and Lu, W., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Langlais, P., Yvon, F., and Zweigenbaum, P. (2009). Improvements in analogical learning: Application to translating multi-terms of the medical domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 487–495, USA. Association for Computational Linguistics.

Lepage, Y. (2004). Analogy and Formal Languages. *Electronic Notes in Theoretical Computer Science*, 53:180–191.

Lepage, Y. (2019). 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*, pages 163–179, Berlin, Heidelberg. Springer-Verlag.

Levine, A. S. (1950). Construction and use of verbal analogy items. *Journal of Applied Psychology*, 34(2):105–107.

Levy, O. and Goldberg, Y. (2014). Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. (2021). Datasets: A community library for natural language processing. In Adel, H. and Shi, S., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. (2020). On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Lim, S., Prade, H., and Richard, G. (2021). Classifying and completing word analogies by machine learning. *International Journal of Approximate Reasoning*, 132:1–25.

Linzen, T. (2016). Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, Berlin, Germany. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Manning, C. D. (2022). Human Language Understanding & Reasoning. *Daedalus*, 151(2):127–138.

Mao, W. and Lepage, Y. (2023). Embedding-to-embedding method based on autoencoder for solving sentence analogies. In Malburg, L. and Verma, D., editors, *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*, pages 15–26. CEUR-WS.org.

Miclet, L., Bayoudh, S., and Delhay, A. (2008). Analogical Dissimilarity: Definition, Algorithms and Two Experiments in Machine Learning. *Journal of Artificial Intelligence Research*, 32:793–824.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Mill, J. S. (1843). *A System of Logic: Ratiocinative and Inductive*. in the version presented by Jonathan Bennett at www.earlymoderntexts.com.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the*

Association for Computational Linguistics, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Murena, P.-A., Al-Ghossein, M., Dessalles, J.-L., and Cornuéjols, A. (2020). Solving Analogies on Words based on Minimal Complexity Transformation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 1848–1854, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.

Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In Hajič, J., Carberry, S., Clark, S., and Nivre, J., editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Newell, A., Simon, H. A., et al. (1972). *Human Problem Solving*, volume 104. Prentice-hall Englewood Cliffs, NJ.

Oppenheimer, R. (1956). Analogy in science. *American Psychologist*, 11(3):127.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Patel, A., Li, B., Rasooli, M. S., Constant, N., Raffel, C., and Callison-Burch, C. (2023). Bidirectional Language Models Are Also Few-shot Learners.

Paterson, D. (1925). *Preparation and Use of New-Type Examinations: A Manual for Teachers*. World book Company.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages

1532–1543, Doha, Qatar. Association for Computational Linguistics.

Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130.

Pitis, S. (2016). An Alternate Arithmetic for Word Vector Analogies.

Prade, H. and Richard, G. (2017). Analogical Proportions and Analogical Reasoning - An Introduction. In Aha, D. W. and Lieber, J., editors, *Case-Based Reasoning Research and Development*, volume 10339, pages 16–32. Springer International Publishing, Cham.

Prasad, R., Webber, B., and Lee, A. (2018). Discourse annotation in the PDTB: The next generation. In Bunt, H., editor, *Proceedings of the 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 87–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Prasad, Rashmi, Webber, Bonnie, Lee, Alan, and Joshi, Aravind (2019). Penn Discourse Treebank Version 3.0.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1).

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Reitman, W. R., Grove, R. B., and Shoup, R. G. (1964). Argus: An information-processing model of thinking. *Behavioral Science*, 9(3):270–281.

Ri, N., Lee, F.-T., and Verma, N. (2023). 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)*, pages 164–173, Toronto, Canada. Association for Computational Linguistics.

Rogers, A., Drozd, A., and Li, B. (2017). The (too Many) Problems of Analogical Reasoning with Word Vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148, Vancouver, Canada. Association for Computational Linguistics.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1985). Learning internal representations by error propagation.

Rumelhart, D. E. and Mcclelland, J. L. (1986). Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1. Foundations.

Salton, G. (1989). Automatic text processing: The transformation, analysis, and retrieval of.

Schluter, N. (2018). 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)*, pages 242–246, New Orleans, Louisiana. Association for Computational Linguistics.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Shazeer, N. and Stern, M. (2018). Adafactor: Adaptive Learning Rates with Sublinear Memory Cost.

Spearman, C. (1927). The abilities of man: Their nature and measurement. *Humana Mente*, 2(8):557–560.

Sternberg, R. J. (1977). Component processes in analogical reasoning. *Psychological Review*, 84(4):353–378.

Su, H., Shi, W., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., Yih, W.-t., Smith, N. A., Zettlemoyer, L., and Yu, T. (2023). One Embedder, Any Task: Instruction-Finetuned Text Embeddings.

Su, J., Cao, J., Liu, W., and Ou, Y. (2021). Whitening Sentence Representations for Better Semantics and Faster Retrieval.

Sultan, O. and Shahaf, D. (2022). Life is a circus and we are the clowns: Automatically finding analogies between situations and processes. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3547–3562, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thibodeau, P. H., Flusberg, S. J., Glick, J. J., and Sternberg, D. A. (2013). An emergent approach to analogical inference. *Connection Science*, 25(1):27–53.

Thrush, T., Ngo, H., Lambert, N., and Kiela, D. (2022). Online language modelling data pipeline.

Tuggy, D. (1985). The inflectional/derivational distinction. *Work Papers of the Summer Institute of Linguistics, University of North Dakota Session*, 29(1).

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, LIX(236):433–460.

Turney, P. D. and Littman, M. L. (2003). Learning Analogies and Semantic Relations.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [cs]*.

Von Werra, L., Tunstall, L., Thakur, A., Luccioni, S., Thrush, T., Piktus, A., Marty, F., Rajani, N., Mustar, V., and Ngo, H. (2022). Evaluate & evaluation on the hub: Better best practices for data and model measurements. In Che, W. and Shutova, E., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 128–136, Abu Dhabi, UAE. Association for Computational Linguistics.

Wang, K., Reimers, N., and Gurevych, I. (2021). TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning.

Wang, L. and Lepage, Y. (2020). Vector-to-Sequence Models for Sentence Analogies. In *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 441–446, Depok, Indonesia. IEEE.

Webb, T., Holyoak, K. J., and Lu, H. (2023). Emergent analogical reasoning in large language models. *Nature Human Behaviour*.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H.-h., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022). Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Wijesiriwardene, T., Wickramarachchi, R., Gajera, B., Gowaikar, S., Gupta, C., Chadha, A., Reganti, A. N., Sheth, A., and Das, A. (2023). ANALOGICAL - A Novel Benchmark for Long Text Analogy Evaluation in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3534–3549, Toronto, Canada. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2020). XLNet: Generalized Autoregressive Pretraining for Language Understanding.

Zhang, C., Gao, F., Jia, B., Zhu, Y., and Zhu, S.-C. (2019). RAVEN: A dataset for relational and analogical visual rEasoNing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Q. and Lepage, Y. (2023). Improving sentence embedding with sentence relationships from word analogies. In Malburg, L. and Verma, D., editors, *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*, pages 43–53. CEUR-WS.org.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. (2022). OPT: Open Pre-trained Transformer Language Models.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT.

Zhu, X. and de Melo, G. (2020). Sentence Analogies: Exploring Linguistic Relationships and Regularities in Sentence Embeddings.

Appendix A

Tables and figures

Category	Relation	Jaccard similarity
Encyclopedic	capital-country	0.25
	country-language	0.19
	hypernym-animal	0.15
	invention-creator	0.10
	member-band	0.22
	meronym-substance	0.13
	misc-hypernym	0.14
	person-occupation	0.08
Lexical	idiom-literal	0.35
	numeral-spelled	0.69
	numeric-approximation	0.65
	past-future	0.53
	present-future	0.57
	present-past	0.61
Semantic	cause-effect	0.18
	description-state	0.24
	home-outdoors	0.57
	informal-formal	0.35
	phrasal-implicative-entailment	0.43
	sentence-opposite	0.44
	sentiment-good-bad	0.53
	simple-implicative-entailment	0.40
Syntactic	active-passive	0.37
	because-so	0.63
	canonical-extraposition	0.69
	canonical-verb-particle-movement	0.99
	qa2d-declarative-howmany	0.49
	qa2d-declarative-howmuch	0.45
	qa2d-declarative-what	0.47
	qa2d-declarative-when	0.45
	qa2d-declarative-where	0.40
	qa2d-declarative-who	0.56

Table A.1. Average Jaccard similarity between SATS pairs per relation

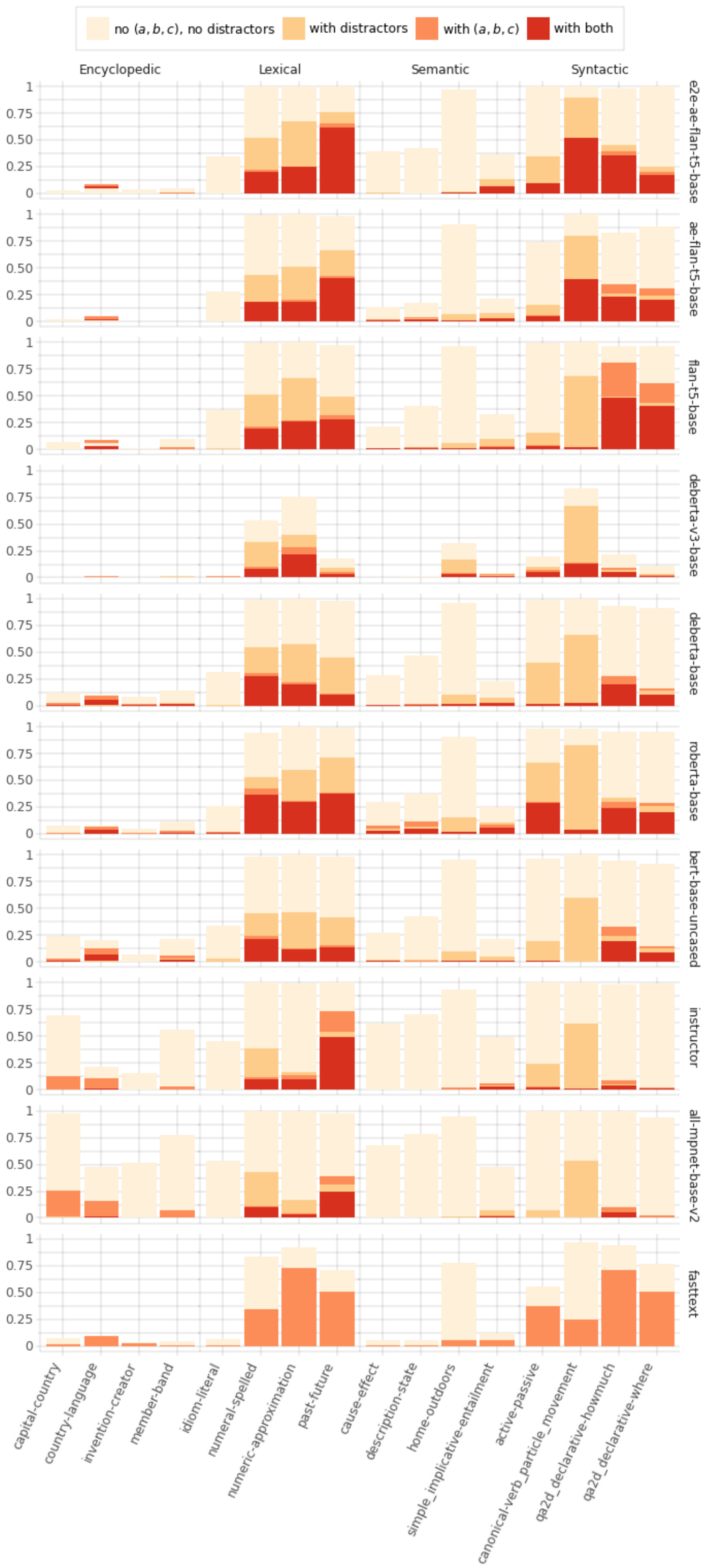

Figure A.1. Retrieval accuracy on SATS test split using the Abelian solver

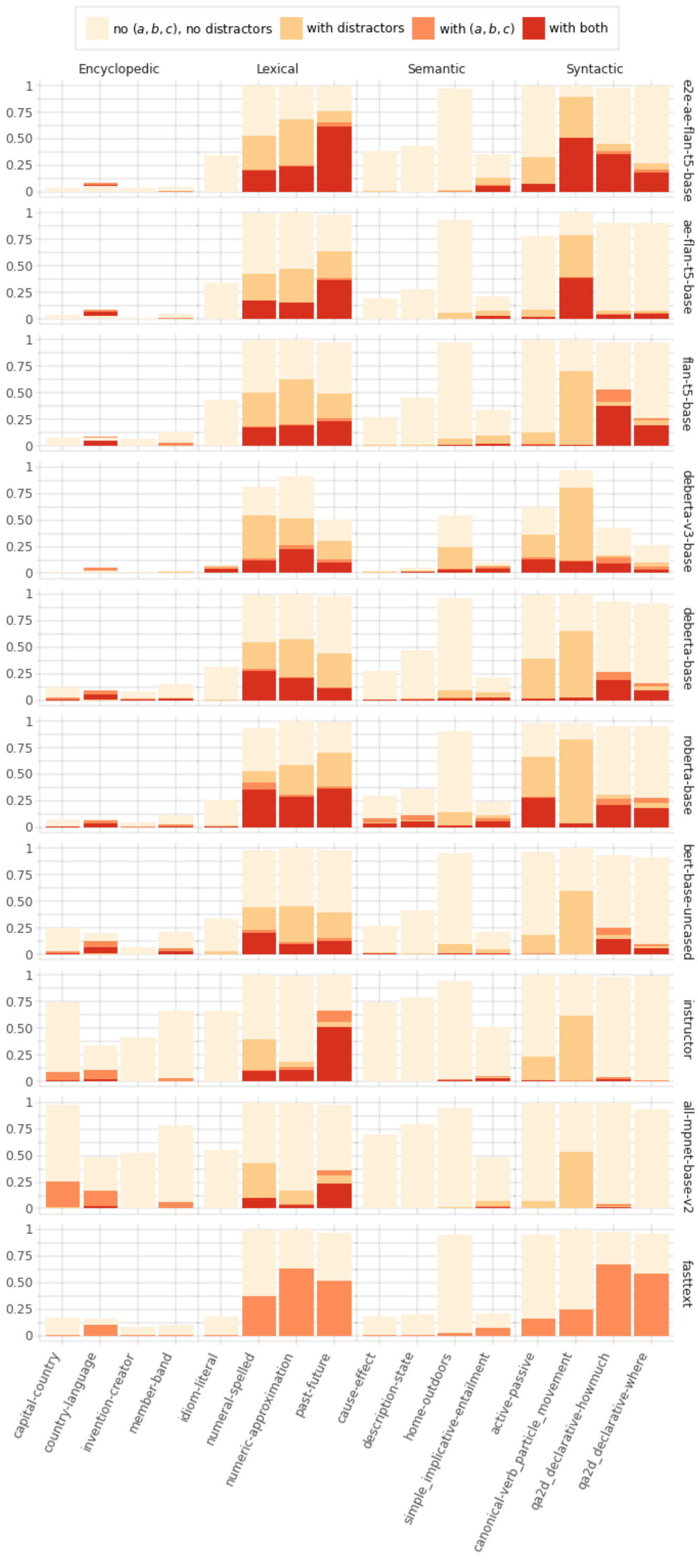

Figure A.2. Retrieval accuracy on SATS test split using the arithmetic solver

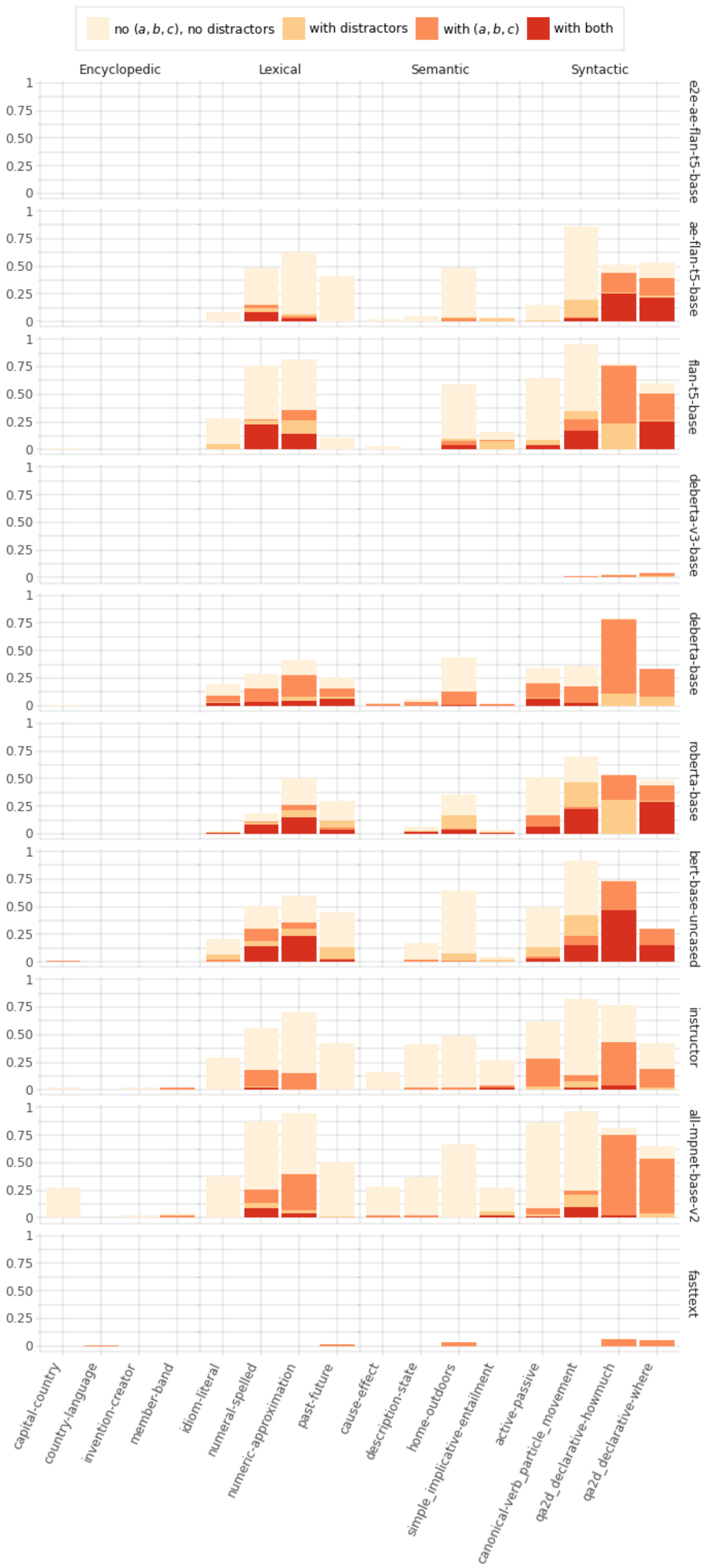

Figure A.3. Retrieval accuracy on SATS test split using the feedforward solver

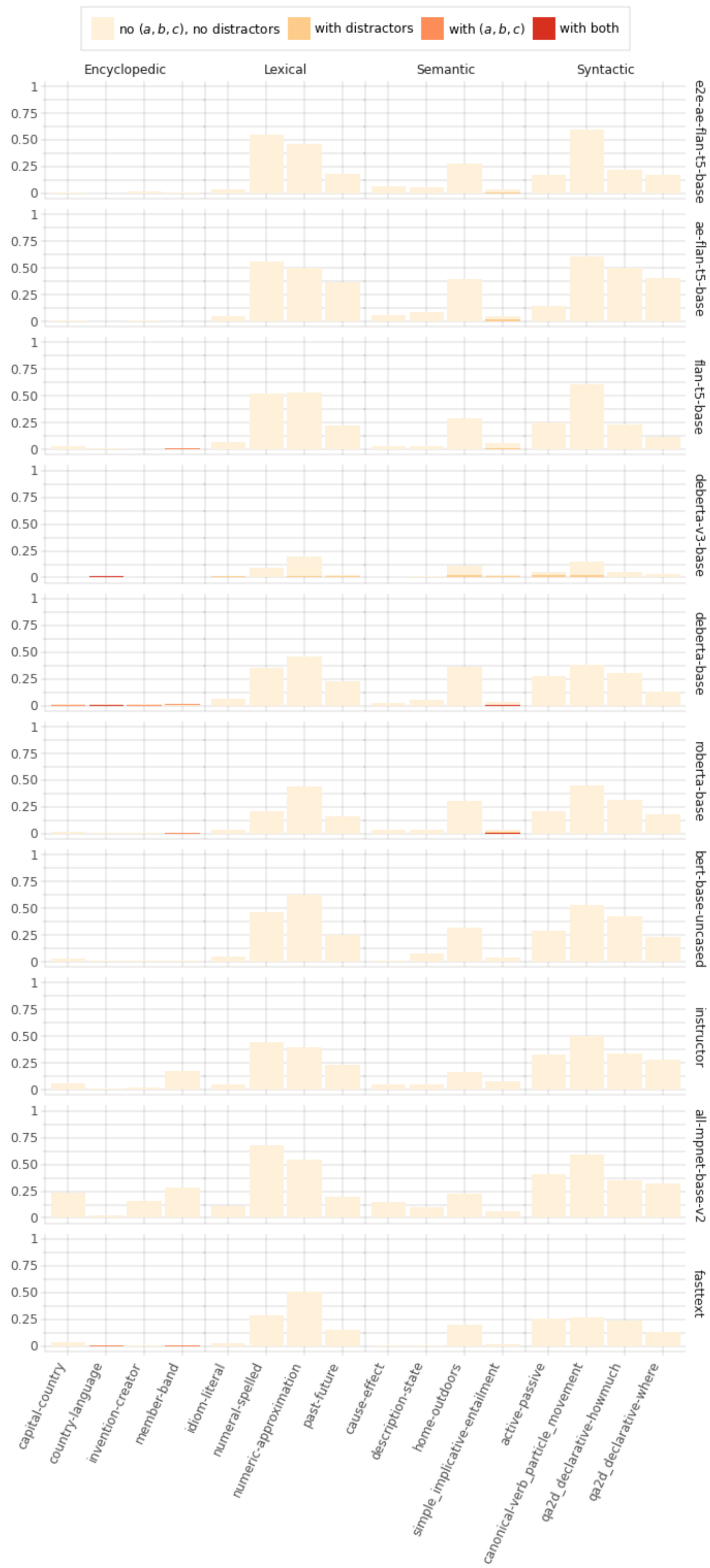

Figure A.4. Retrieval accuracy on SATS test split using the mean premise solver