

VASR: Visual Analogies of Situation Recognition

Yonatan Bitton, Ron Yosef, Eli Strugo, Dafna Shahaf, Roy Schwartz, Gabriel Stanovsky

The Hebrew University of Jerusalem

{yonatan.bitton,ron.yosef,eli.strugo,dafna.shahaf,roy.schwartz1,gabriel.stanovsky}@mail.huji.ac.il

Abstract

A core process in human cognition is *analogical mapping*: the ability to identify a similar relational structure between different situations. We introduce a novel task, Visual Analogies of Situation Recognition, adapting the classical word-analogy task into the visual domain. Given a triplet of images, the task is to select an image candidate B' that completes the analogy (A to A' is like B to what?). Unlike previous work on visual analogy that focused on simple image transformations, we tackle complex analogies requiring understanding of scenes.

We leverage situation recognition annotations and the CLIP model to generate a large set of 500k candidate analogies. Crowdsourced annotations for a sample of the data indicate that humans agree with the dataset label $\sim 80\%$ of the time (chance level 25%). Furthermore, we use human annotations to create a gold-standard dataset of 3,820 validated analogies. Our experiments demonstrate that state-of-the-art models do well when distractors are chosen randomly ($\sim 86\%$), but struggle with carefully chosen distractors ($\sim 53\%$, compared to 90% human accuracy). We hope our dataset will encourage the development of new analogy-making models. Website: <https://vasr-dataset.github.io/>

1 Introduction

The ability to draw analogies, flexibly mapping relations between superficially different domains, is fundamental to human intelligence, creativity and problem solving (Hofstadter and Sander 2013; Depeweg, Rothkopf, and Jäkel 2018; Goodman, Tenenbaum, and Gerstenberg 2014; Fauconnier 1997; Gentner, Holyoak, and Kokinov 2001; Carey 2011; Spelke and Kinzler 2007). This ability has also been suggested to be key to constructing more general and trustworthy AI systems (Mitchell 2021; McCarthy et al. 2006). An essential part of analogical thinking is the ability to look at different *situations* and extract abstract patterns. For example, a famous analogy is between the solar system and the Rutherford-Bohr model of the atom. Importantly, while the surface features are very different (atoms are much smaller than planets, different forces are involved, etc.), both phenomena share deep structural similarity (e.g., smaller objects revolving around a massive object, attracted by some force).

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

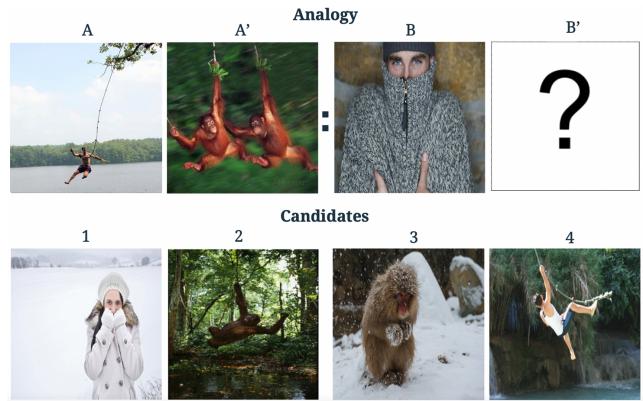


Figure 1: An example of visual analogy from the VASR dataset. The task is to select an image which best completes the analogy. The answer is found in the footnote.

Most computational analogy works to date have focused on text (Mikolov, Yih, and Zweig 2013; Allen and Hospedales 2019), often studying SAT-type analogies (e.g., walk:legs :: chew:mouth). In works involving analogies between *situations* (Falkeneheimer, Forbus, and Gentner 1986; Evans 1964; Winston 1980; Gentner 1983), both entities and relations need explicit structured representations, limiting their scalability. In the visual domain, works also focused on SAT-type questions (Lovett and Forbus 2017; Lake, Salakhutdinov, and Tenenbaum 2015; Depeweg, Rothkopf, and Jäkel 2018), synthetic images (Lu et al. 2019; Reed et al. 2015) or images depicting static objects, where the analogies focus on object properties (color, size, etc.) (Tewel et al. 2021; Sadeghi, Zitnick, and Farhadi 2015), rather than requiring understanding of a full scene.

In this work we argue that images are a promising source of *relational* analogies between situations, as they provide rich semantic information about the scenes depicted in them. We take a step in that direction and introduce the Visual Analogies of Situation Recognition (VASR) dataset. Each instance in VASR is composed of three images (A , A' , B) and $K = 4$ candidates (see Figure 1). The task is to se-

Answer: 3. Between A and A' , *man* changed to *monkey*. Thus, from B to B' , a *man* feeling cold changes to a *monkey* feeling cold.



Figure 2: Two images and their situation recognition annotations from imSitu. In this example, both images share the same annotations except for the *item* role (*boat* → *tractor*).

lect the candidate B' such that the relation between B and B' is most analogous to the relation between A and A' . To solve the analogy in Figure 1, one needs to understand the key difference between A and A' (the main entity is changed from *man* to *monkey*) and map it to B (“A *man* feeling cold” is changed to “A *monkey* feeling cold”). Importantly, VASR focuses on situation recognition that requires understanding the full scene, the different roles involved and how they relate to each other.

To create VASR, we develop an automatic method that leverages situation recognition annotations¹ to generate silver analogies of different kinds.² We start with the imSitu corpus (Yatskar, Zettlemoyer, and Farhadi 2016), which annotates frame roles in images. For example, in the image on the left of Figure 2, the *agent* is a *truck*, the *verb* is *hauling*, and the *item* (or *theme*) is a *boat*. We search for instances $A : A' :: B : B'$ where: (1) $A : A'$ are annotated similarly except for a single different role; (2) $B : B'$ exhibit the same delta in frame annotation. For example in Figure 2, the images are annotated the same except for *item* that is changed from *boat* to *tractor*. The corresponding $B : B'$ images pairs should similarly have *boat* as an *item* role in B , and *tractor* as an *item* in B' , while all other roles are identical between them. We use several filters aiming to keep pairs of images that have a single main salient difference between them, and carefully choose the distractors to adjust the difficulty of the task. This process produces over 500,000 instances, with diverse analogy types (activity, tool being used, etc.).

To create a gold standard and to evaluate the automatic generation of VASR, we crowd-source a portion of 4,170 analogies of the silver annotations using five annotators. On the test set, we find that annotators are very likely (93%) to agree on the analogy answer, and reach high agreement with the auto-generated label (79%). For human evaluation, we crowd-source additional annotations from new annotators who did not participate in the data generation part, evaluating a sample of 10% of the gold-standard test set, finding that they solve it with high accuracy (90%).

We evaluate various state-of-the-art computer vision models (ViT (Dosovitskiy et al. 2020), Swin Transformer (Liu

¹Often referred to as visual semantic role labeling (Gupta and Malik 2015).

²We use the term “silver labels” to refer to labels generated by an automatic process, which, unlike gold labels, are not validated by human annotators.

et al. 2021), DeiT (Touvron et al. 2021) and ConvNeXt (Liu et al. 2022)) in zero-shot settings using arithmetic formulations, following similar approaches in text and in vision (Mikolov, Yih, and Zweig 2013). We find that they can solve analogies well when the distractors are chosen randomly (86%), but all struggle with well-chosen difficult distractors, achieving only 53% accuracy on VASR, far below human performance. Interestingly, we show that training baseline models on the large silver corpus is comparable with zero-shot performance and far below human performance, leaving room for future research.

Our main contributions are: (1) we present the VASR dataset as a resource for evaluating visual analogies of situation recognition; (2) we develop a method for automatically generating silver-label visual analogies from situation recognition annotations; (3) we show that current state-of-the-art models are able to solve analogies with random candidates, but struggle with more challenging distractors.

2 Related Work

The VASR dataset is built using annotations of situation recognition from imSitu, described below. In addition, we discuss two works most similar to ours, which tackle different aspects of analogy understanding in images.

Situation Recognition. Situation recognition is the task of predicting the different semantic role labels (SRL) in an image. For example in Figure 1, image A depicts a frame where the *agent* is a *person*, the *verb* is *swinging*, the *item* is a *rope*, and the *place* is a *river*. The imSitu dataset (Yatskar, Zettlemoyer, and Farhadi 2016) presented the task along with annotated images gathered from Google image search, and a model for solving this task. Each annotation in imSitu comprises of *frames* (Fillmore, Johnson, and Petrucc 2003), where each noun is linked to WordNet (Miller 1992), and objects are identified in image bounding boxes.³ We use these annotations to automatically generate our silver analogy dataset.

Analogy. Analogies have been studied in multiple contexts. Broadly speaking, computational analogy methods can be divided into symbolic methods, probabilistic program induction, and neural approaches (Mitchell 2021).

In the context of analogies between *images*, there have been several attempts to represent *transformations* between pairs of images (Memisevic and Hinton 2010; Reed et al. 2015; Hertzmann et al. 2001; Forbus et al. 2011). The transformations studied were usually stylistic (texture transfers, artistic filters) or geometric (topological relations, relative position and size, 3D pose modifications).

More recently, DCGAN (Radford, Metz, and Chintala 2016) has shown capabilities of executing vector arithmetic on images of faces, e.g. (man with glasses - man without glasses + woman without glasses ≈ woman with glasses). Another work, focusing on zero-shot captioning (Tewel et al. 2021), presented a model based on CLIP and GPT-2 (Radford et al. 2019) for solving visual analogies, where the input

³Follow-up work (Pratt et al. 2020) added bounding boxes to imSitu.

consists of three images and the answer is textual. We evaluate their model in our experiments.

Perhaps most similar to our work is VISALOGY (Sadeghi, Zitnick, and Farhadi 2015). In this work, the authors construct two image analogy datasets—a synthetic one (using 3D models of chairs that can be rotated) and a natural-image one, using Google image search followed by manual verification. However, even in the natural-image case, the analogies in VISALOGY are quite restricted; images mostly contain a single main object (e.g., a dog) and analogies based on attributes (e.g., color) or action (e.g., run). The VASR dataset contains analogies that are much more expressive, requiring understanding the full scene (see Figure 15 in Appendix 6). Importantly, the VISALOGY dataset is not publicly available, which makes VASR, to the best of our knowledge, the only publicly available benchmark for visual situational analogies with natural images.

3 The VASR Dataset

To build the VASR dataset, we leverage situation recognition annotations from imSitu. We start by finding likely image candidates based on the imSitu gold annotated frames (§3.1). We then search for challenging answer distractors (§3.2). Following, we apply several filters (§3.3) in order to keep pairs of images with a single salient difference between them. We then select candidates for the gold test set (§3.4), and crowdsource the annotation of a gold dataset (§3.5). Finally, we provide the dataset statistics (§3.6).

3.1 Finding Analogous Situations in imSitu

We start by considering the imSitu dataset containing situation recognition annotations of 125,000 images. We search for images $A : A'$ that are annotated the same, except for a single different role (e.g., the *agent* role in Figure 1 is changed from *man* to *monkey*). We extract image pairs that have the same situation recognition annotation yet differ in one of the following roles: agent, verb, item, tool, vehicle and victim. This process yields ~ 7 million image pairs. However, many of these pairs are not analogous because they do not have a *single* salient visual difference between them (as exemplified in Figure 3), due to partial annotation of the imSitu images. To overcome this, we apply several filters, described in Section 3.3, keeping $\sim 23\%$ of the pairs. Next, for each $A : A'$ pair we search for another pair of images, $B : B'$, which satisfy a single condition, namely that they exhibit the same difference in roles. Importantly, note that $B : B'$ can be very different from $A : A'$, as long as they adhere to this condition.

3.2 Choosing Difficult Distractors

Next, we describe how we compose VASR instances out of the analogy pairs collected in the previous section. The candidates are composed of the correct answer B' and three other challenging distractors. Our experiments (§4) demonstrate the value of our method for selecting difficult distractors compared to randomly selected distractors. Figure 4 illustrates this difference.



Figure 3: An image pair with *multiple* salient visual differences (dog breed, activity, and more). We aim to filter these cases, keeping pairs with *single* main salient difference.



Figure 4: Compared to random distractors (on the left), VASR includes difficult distractors (on the right).

Specifically, we want distractors that would impede shortcuts as much as possible. Namely, the correct answer should involve two reasoning steps: (1) understanding the key difference between $A : A'$ (the agent role *man* changed to *monkey* in Figure 1); (2) Map it to B . For the first reasoning step, we include distractors that are similar to B but that do not have the same value in the changed role in A' (candidates 1, 4 in Figure 1 do *not* depict a *monkey*). For the second reasoning step, we include distractors with the changed role in A' but in a different situation than B (candidate 2 in Figure 1, which does show a *monkey*, but in a different situation). To provide such distractors, we search for images that are annotated similarly to A' and B . For the similarity metric, we use an adaption of the Jaccard similarity metric between the images annotations. We calculate the number of joint values divided by the size of the union between the key sets of both images.⁴ We start by extracting multiple suitable distractors (40 in *dev* and *test*, 20 in *train*). We later select the final 3 distractors using the filtering step described below (§3.3).

3.3 Filtering Ambiguous Image Pairs

We note that our automatic process is subject to several potential sources of error. One of them is the situation recognition annotations. The imSitu corpus was not created with analogies in mind, and as a result salient differences between

⁴https://en.wikipedia.org/wiki/Jaccard_index. For example, for the two dictionaries $\{ \text{'a': 1, 'b': 2} \}$, $\{ \text{'a': 1, 'c': 2} \}$, the adapted Jaccard index is 1/3, because there is one joint value for the same key ('a': 1) and three keys in the union ('a', 'b', 'c')

the images are often omitted, and seemingly less important differences are highlighted. In this section, we attempt to ameliorate the issue and propose different filters to keep only pairs with one salient difference. We stress that there are many more filtering strategies possible, and exploring them is left for future work.

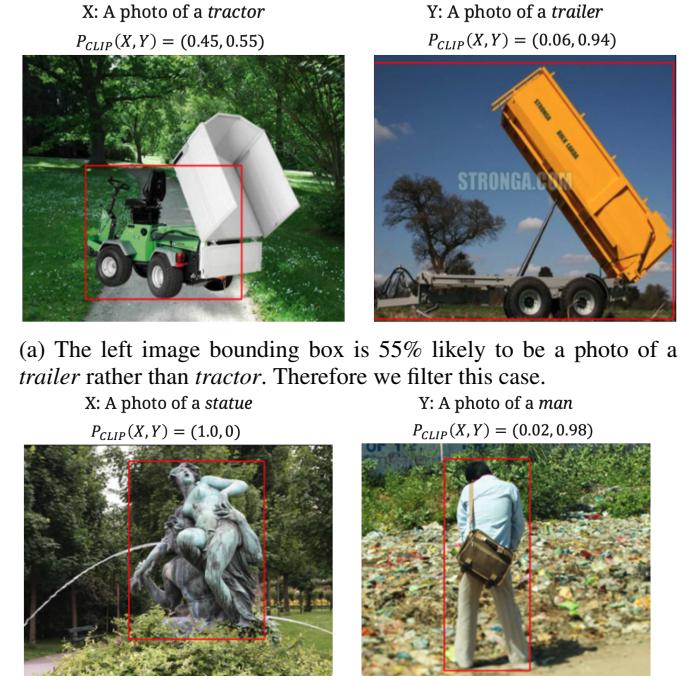
Over-specified annotations We filter image pairs with overly-specific differences. For example, in Figure 3 the frames are annotated identically except for the *agent* which is changed from *beagle* to *puppy*, while a human observer is likely to identify more salient differences (leash color, activity, and more). To mitigate such cases, we use a textual filter by leveraging imSitu’s use of WordNet (Miller 1992) for nouns and FrameNet (Fillmore, Johnson, and Petrucc 2003) for verbs. We identify the lowest common hypernyms for each annotated role (A *beagle* is a type of a *dog*, which is a type of a *mammal*). Next, we only keep instances adhering to one of the following criteria: (1) both instances’ corresponding roles are direct children to the same pre-defined WordNet concept class,⁵ e.g., *businessman* and *businesswoman* are both direct children of *businessperson*; (2) pairs of co-hyponyms, e.g., cat and dog are both animals, but a cat is not a dog and vice-versa; (3) the two instances belong to different clusters of animal, inanimate objects, or humans (e.g., *bike* changed to *cat* or *car* changed to *person*). This process removes 40% of the original pairs. Filtered pairs are likely to be indistinguishable, for example: *beagle* and *puppy*, *cat* and *feline*, *person* and *worker*, and so on.

Another case of over-specific annotations is when a non visually salient object is being annotated. For example in Figure 16 in Appendix 6 the annotated object is a small *boomerang* that might be hard to identify. To mitigate such cases, we leverage bounding-boxes annotations from the SWiG dataset (Pratt et al. 2020) and filter cases where the objects are hard to identify. Images with object size smaller than 2% of the image size are filtered this way, filtering an additional 4%.

Under-specified annotations The imSitu annotation is inherently bound to miss some information encoded in the image. This can result in image pairs A, A' that exhibit multiple salient differences, yet only a subset of them is annotated, leading to ambiguous analogies. For example in Figure 5 top, the left image is described as a *tractor*, and the right image described as a *trailer*. However, the left image can be considered as a *trailer* as well, and it is not clear what is the main difference between this images pair. We aim to filter cases of such ambiguity, where an object can describe the *other* image bounding box. For example, in Figure 5, the top example (a) is filtered by our method and the bottom example (b) is kept. Given two bounding boxes X, Y —each corresponding to different images—and two different annotated objects X_{obj}, Y_{obj} , we compute the CLIP (Radford et al. 2021) probabilities to describe each object bounding box using the prompt of “A photo of a [OBJ]”. We denote

$$P_{X_{img}}(X_{obj}, Y_{obj}) = (P(X_{img}, X_{obj}), P(X_{img}, Y_{obj}))$$

⁵See full list of WordNet concepts in Appendix 6.



(a) The left image bounding box is 55% likely to be a photo of a *trailer* rather than *tractor*. Therefore we filter this case.

(b) Both objects (*statue, man*) better describe their images bounding boxes (in 100% and 98%). Therefore we keep this instance.

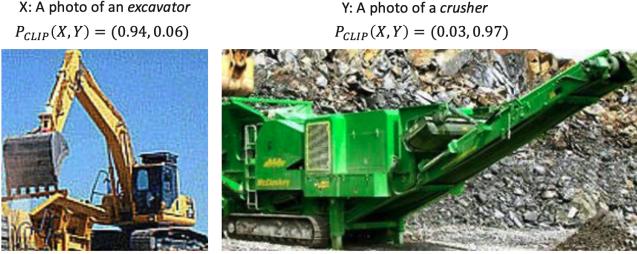
Figure 5: Two examples for our CLIP based vision-and-language filtering. Given two images and annotated objects we compute the probabilities for each object to describe each image. We filter cases where an object can better describe the *other* image than the image it annotates.

(and vice-versa for Y) and filter cases where it is not distinct. For example in the left image in Figure 5, $P_{X_{img}}(X_{obj}, Y_{obj}) = (0.45, 0.55)$. The left image (X) is 55% likely to be a photo of a *trailer* (Y annotation) rather than *tractor* (X annotation), therefore we filter this pair. We filter based on a threshold computed on a development set. We also execute a “mesh filter”, where we combine all object labels of both images, measure the best object for each image, and filter cases where the best describing object for an image bounding box belongs to the other image.

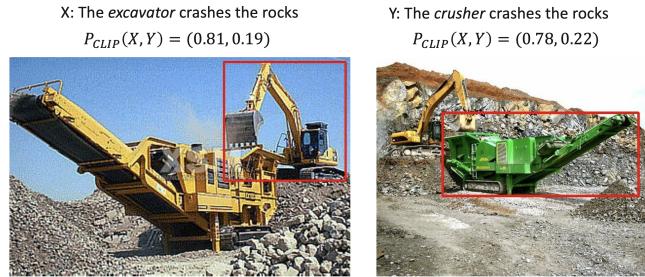
Additionally to the objects and image bounding boxes, we also take into consideration CLIP features extracted from the full image. Examples are presented in Figure 6. Instead of taking a template sentence of “A photo of an [OBJ]”, we use a FrameNet template (Fillmore, Johnson, and Petrucc 2003) to receive a sentence describing the full image. For example the verb “crashing” (Figure 6) has the FrameNet template of: “the AGENT crashes the ITEM...”. We substitute the annotated roles for the image, receiving a synthetic sentence describing the image. The CLIP probabilities are then used to filter indistinctive cases as in bounding-box filtering.

3.4 Building the Test Set

We aim to make the test set both challenging and substantially different from the training set in order to measure model generalizability. To do so, we select challeng-



(a) Based on the bounding box only, no ambiguity between the images and object classes.



(b) Based on the full image, the distinction between the images isn't that clear as in the bounding boxes case on the left.

Figure 6: CLIP-based filtering, bounding box vs. full image. The filter decision needs to consider both signals. Here the left figure is distinctive but the right is not, so we filter it out.

ing test instances according to 3 metrics, defined below. In Section 3.5, we validate these instances via crowd-workers, finding them to be of good quality. The metrics are: (1) an adapted Jaccard similarity metric to compute the difference in annotation between A , A' . We aim to select items with low Jaccard similarity to receive analogies that are *distant* from each other; (2) calculate occurrences of each different key in the training set, in order to prefer rare items. For example $A : A'$ of *girrafe* : *monkey* is preferred over *man* : *monkey* if *girrafe* appeared less than *man* in the training set; (3) High annotation CLIP match: to avoid images with noisy annotations, we use the features computed in Section 3.3 to calculate an “Image SRL score” using a weighted average of: (a) CLIP score of the caption to the image $P_{X_{img}}(X)$; (b) CLIP probability of the caption vs. the caption from the other image pair. For example in the left image in Figure 5 this score is 0.45. We sort our dataset according to these metrics, selecting 2,539 samples for the test set. We evaluate and annotate these candidates with human annotators (§3.5).

3.5 Human Annotation

We pay Amazon Mechanical Turk (AMT) crowdworkers to annotate the ground truth labels for a portion of VASR. We asked five annotators to solve 4,214 analogies.⁶ Workers were asked to select the image that best solves the analogy, and received an estimated hourly pay of 12\$. Total payment

⁶To maintain high-quality work, we have a qualification task of 10 difficult analogies, requiring a grade of at least 90% to enter the full annotation task. The workers received detailed instructions and examples from the project website.

Table 1: AMT annotation results. The annotators are very likely to select the same candidate as the analogy answer, and with high agreement with the auto-generated label.

	Test	Dev	Train
# samples fully annotated	2,539	178	1,492
% of samples with agreement of at least three	93	90	88
% of samples where majority vote agrees with dataset label	79	75	75

to AMT was 1,440\$. Full details and examples of the AMT annotators screen are presented in Appendix 6, Section 6.4.

Table 1 shows some statistics of the annotation process. We observe several trends. First, in 93% of the analogies there was an agreement of at least three annotators on the selected solution, compared to a probability of 41.4% for a random agreement of at least three annotators on a any solution.⁷ Second, in 79% of the instances the majority vote (of at least 3 annotators) agreed with the auto-generated dataset label. Moreover, given that the annotators reached a majority agreement, their choice is the same as the auto-generated label in 85% of the cases. When considering annotators that annotated more than 10% of the test set, the annotator with the highest agreement with the auto-generated label achieved 84% agreement. Overall, these results indicate that the annotators are very likely to agree on a majority vote and with the silver label. The resulting dataset is composed of the 3,820 instances agreed upon with a majority vote of at least 3 annotators.

3.6 Final Datasets and Statistics

The analogies generation process produces over 500,000 analogies using imSitu annotations. We used human annotators (§3.5) to create gold-standard split, with 1,310, 160, 2,350 samples in the *train*, *dev*, *test* (§3.4), respectively. Next, we create a silver *train* of size 150,000 items and a silver *dev* set of size 2,249 items. We sample the silver *train* and *dev* sets randomly, but we balance the proportions of different types of analogies similar to the *test*.

VASR contains a total of 196,269 object transitions (e.g., *book* changed to *table*), of which 6,123 are distinct. It also contains 385,454 activity transitions (e.g., “*smiling*” changed to “*jumping*”), of which 2,427 are distinct. Additional statistics are presented in Appendix 6, Section 6.6. To conclude, we have silver *train* and *dev* sets, and gold *train*, *dev*, and *test* sets. Full statistics are presented in Table 2.

We encourage to focus on solving VASR with little or no training, since solving analogies requires mapping of existing knowledge to new, unseen situations (Mitchell 2021). Evaluation of models should be performed on the (gold) *test* set. To encourage development of models to solve VASR, an evaluation page is available on the website. The ground truth answers are kept hidden, predictions can be sent to our email and we will update the leaderboard. In a few-shot fine-tune setting, we suggest using the gold-standard *train* and *dev* splits, containing 1,470 analogies. For larger fine-tune, we suggest using the silver *train* and *dev* sets, with

⁷Binomial distribution analysis shows that the probability to get a random majority of at least 3 annotators out of 5 is 41.4%.

Table 2: VASR statistics. Rows 1-2 describe the silver data, and rows 3-5 describe the gold-standard data.

		Agent	Verb	Item	Tool	Vehicle	Victim	Total
Silver	Train	82,984	38,331	20,836	6,360	1,343	146	150,000
	Dev	1,704	123	238	146		38	2,249
Gold	Train	558	116	376	170	90		1,310
	Dev	129	7	12	10		2	160
	Test	795	368	554	160	169	304	2,350

152,249 analogies. We also publish the full generated data (over 500K analogies) to allow other custom splits. Next we turn to study state-of-the-art models’ performance on VASR.

4 Experiments

We evaluate humans and state-of-the-art image recognition models in both zero-shot and supervised settings. We show that VASR is easy for humans (90% accuracy) and challenging for models (<55%). We provide a detailed analysis per analogy type, experiments with partial inputs (when only one or two images are available from the input), and experiments with increased numbers of distractors.

4.1 Human Evaluation

We sample 10% of the test set, and ask annotators that did not work on previous VASR tasks to solve the analogies. Samples from the validation process are presented in Appendix 6, Section 6.3. Each analogy is evaluated by 10 annotators and the chosen answer is the majority of 6 annotators.⁸ We find that the human performance on the test set is 90%. Additionally, in 93% of the samples there was an agreement of at least six annotators. This high human performance indicates the high quality of our end-to-end generation pipeline.

4.2 Zero-Shot Models

We compare four model baselines:

1. *Zero-Shot Arithmetic*: Inspired by Word2Vec (Mikolov, Yih, and Zweig 2013), we extract visual features from pre-trained models for each image and represent the input in an *arithmetic* structure by taking the embedding of $B + A' - A$. We compute its cosine similarity to each of the candidates and pick the most similar. We experiment with the following models: ViT (Dosovitskiy et al. 2020), Swin Transformer (Liu et al. 2021), DeiT (Touvron et al. 2021) and ConvNeXt (Liu et al. 2022).⁹ Figure 19 in Appendix 6 illustrates this baseline.
2. *Zero-Shot Image-to-Text* (Tewel et al. 2021) presented a model for solving visual analogy tests in zero-shot setting. Given an input of three images A, A', B , this model uses an initial prompt (“An image of a ...”) and generates the best caption for the image represented by the same *arithmetic* representation we use: $B + A' - A$. We

⁸The probability to receive a random majority vote of at least six annotators out of 10 is 7.9%.

⁹The exact versions we took are the largest pretrained versions available in *timm* library: ViT Large patch32-384, Swin Large patch4 window7-224, DeiT Base patch16 384, ConvNeXt Large.

calculate the CLIP score between each image candidate and the caption generated by the model, and select the candidate with the highest score.

3. *Distractors Elimination*: similar to a multi-choice quiz elimination, this strategy takes the three candidates that are most similar to the inputs A, A', B , eliminates them, and selects the last candidate as the final answer. We use the pre-trained ViT embeddings and compute cosine similarity in order to select the similar candidates.
4. *Situation Recognition Automatic Prediction*: This strategy uses automatic situation recognition model prediction from SWiG (Pratt et al. 2020). It tries to find a difference between $A : A'$ in the situation recognition prediction and map it to B , in a reversed way to the VASR construction. For example in Figure 1 it will select the correct answer *if* both $A : A'$ and $B : B'$ are predicted with the same situation recognition prediction except *man* changed to *monkey*.

4.3 Supervised Models

We also consider models fine-tuned on the silver data. We add a classifier on top of the pre-trained embeddings to select one of the 4 candidates. The first model baseline (denoted *Supervised Concat*) concatenates the input embeddings and learns to classify the answer $(A, A', B) \rightarrow B'$. The second model baseline (denoted *Supervised Arithmetic*) has the same input representation as *Zero-Shot Arithmetic*. To classify an image out of 4 candidates, we follow the design introduced in SWAG (Zellers et al. 2018),¹⁰ which was used by many similar works (Sun et al. 2019; Huang et al. 2019; Liang, Li, and Yin 2019; Dzembzik, Vogel, and Foster 2021). Basically, each of the image candidates is concatenated to the inputs features, followed by a linear network activation and a classifier that selects one of the options. We use the Adam (Kingma and Ba 2015) optimizer, a learning rate of 0.001, batch size of 128, and train for 5 epochs. We take the model checkpoint with the best silver *dev* performance out of the 5 epochs, and use it for evaluation. Figure 20 in Appendix 6 illustrates this model.

4.4 Results and Model Analysis

Table 3 shows our *test* accuracy results. Rows 1-7 show the zero-shot results. The *Zero-Shot Arithmetic* models (R1-R4) achieve the highest results, with small variance between the models, reaching up to 86% with random distractors and around 50% on the difficult ones. The *Zero-Shot Image-to-Text* (R5) achieves lower accuracies on both measures (70% and 38.9%, respectively). The other two models perform at chance level for difficult distractors.¹¹ To conclude, models can solve analogies in zero-shot well when the distractors are random, but struggle with difficult distractors.

¹⁰<https://huggingface.co/transformers/v2.1.1/examples.html?#multiple-choice>

¹¹*Distractors Elimination* strategy is particularly bad with random distractors, as it eliminates the 3 images closest to the input, whereas the solution is often closer to the inputs than random distractors.

Table 3: VASR test set accuracy for several baselines in zero-shot and training. Bold indicates best result in section.

Section	Experiment	Random Distractors	Difficult Distractors	Row
Zero-Shot	ViT	86	50.3	1
	Swin	86	52.9	2
	Arithmetic	DEiT	77.7	3
		ConvNeXt	79	51.2
	Zero-Shot Image-to-Text	70	38.9	5
Training on the Silver Data	Distractors Elimination	0.9	23.4	6
	Situation Recognition Automatic Prediction	31	24.6	7
	Concat	84.1	54.9	8
Partial Inputs	Arithmetic	83.7	47.4	9
	Zero-Shot	A'	84.4	10
		B	77.6	24.7
	Supervised	Single image	82.1	44.8
		Pair of images	83.8	46.3
Humans			90	14

Results on training on the silver data are presented in rows 8-9. *Supervised Concat* representation performs better than the *Supervised Arithmetic*. Interestingly, its performance (54.9%, R8) is only 2% higher than the best zero-shot baseline (*Zero-Shot Arithmetic*, R2), and still far from human performance (R14). This small difference might be explained by the distribution shift between the training data and the test data (§3.4), which might make the trained models over-rely on specific features in the training set. To test this hypothesis, we consider the ViT model’s *supervised* performance on the *dev* set, which, unlike the test set, was not created to be different than the training set. We observe *dev* performance levels similar to the *test* set (56.7% with the difficult and 86.6% with random distractors), which hints that models might struggle to capture the information required to solve visual analogies from supervised data.

Analysis per Analogy Type. We study whether humans and models behave differently for different types of analogies. We examine the *test* performance of both humans and the ViT-based models *Zero-Shot Arithmetic* and *Supervised Concat* per analogy type (Table 4). Humans solve VASR above 80% in all analogy types, except for *tool* (66%). The average performance of both models on all categories is around 50%, except for the *Agent* category, which seems to benefit most from supervision. We propose several possible explanations: First, *Agent* is the most frequent class. This does not seem to be the key reason for this result, as the performance of the second most frequent category, *Item*, is far worse. Second, *Agent* is the most visually salient class and the model learns to identify it. This also does not seem to be the reason, because we see that the bounding-box proportion (objects proportions are in the second row¹²) of the *Vehicle* class (55%) are larger than the *Agent* class (44%), but the performance on it is far worse. Finally, solving *Agent* analogies

¹²For example the “person that is feeling cold” in Figure 1 (image *B*) takes >90% of the image size.

Table 4: Results per analogy types of humans and models baselines. The class with the highest/lowest accuracy for each model is in bold. Data Percentage is the proportion of each class from the *gold* test. Objects Proportion is the mean object size divided by full image size.

	Agent	Item	Verb	Victim	Vehicle	Tool	Total
Data Percentage (%)	34	24	16	13	7	7	100
Objects Proportion (%)	44	27	42	55	18		
Humans	95	98	85	84	83	66	89.9
Arithmetic Zero-Shot	50	48	49	48	56	58	50.3
Trained Concatenation	69	50	44	52	46	44	54.9

could be the most similar task to the pre-training data of the models we evaluate, which mostly include images with a single class, without complex scenes and other participants (e.g., images from ImageNet (Deng et al. 2009)). This hypothesis, if correct, further indicates the value of our dataset, which contains many non-Agent analogies, to challenge current state-of-the-art models. We also find that the *Zero-Shot Arithmetic* and *Supervised Concat* predict the same answer only in 40% of the time. An oracle that is correct if either model is correct reaches an accuracy of 76%, suggesting that these models have learned to solve analogies differently.

Partial Inputs. Ideally, solving analogies should not be possible with partial inputs. We experiment with ViT pre-trained embeddings in two setups: (1) A *Zero-Shot* baseline, where the selected answer is the candidate with the highest cosine similarity to the image embeddings of *A'* or *B*. For example in Figure 1, *A'* depicts a “monkey swinging” and *B* depicts a “person shivering”. The candidates most similar to these inputs are 1 and 2, and both are incorrect solutions; (2) A *supervised* baseline, which is the same as *Supervised Concat*, but instead of using all three inputs, we use a single or a pair of images: *A*, *A'*, *B*, (*A*, *B*), (*A*, *A'*), (*A'*, *B*). Results are presented in Table 3, R10-R13. In *Zero-Shot*, the strategy of choosing an image that is similar to *A'* (R10) reaches close to the full inputs performance with random distractors, but much lower with the difficult distractors. With the *supervised* baseline, we show the best setup of a single image (*B*, in R12) and a pair of images ((*A*, *B*), R13). We observe a similar trend to the zero-shot setting, concluding that it is difficult to solve VASR using partial inputs.

Performance in the Presence of more Distractors Since VASR is generated automatically, we can add more distractors and measure models’ performance. We take the *test* set with the ground-truth answer provided by the annotators and change the number of distractors hyperparameter from 3 to 7, adding distractors to each of the random and difficult distractors splits, changing chance level from 25% to 12.25%. We repeat the zero-shot experiments and present the results in Table 5. The ViT performance on the difficult distractors drops from 50.3% to 27.7%, while on the random distractors the decline is much more moderate, from 86% to 78.7%. We observe a similar trend for the other models. The large drop in performance on the difficult distractors further indicates the importance of a careful selection of the distractors.

Table 5: With random candidates, the models manage to cope even though the task becomes twice as difficult. However, the performance drop is larger with difficult distractors.

Models	4 Candidates		8 Candidates		% Drop	
	Random Distractors	Difficult Distractors	Random Distractors	Difficult Distractors	Random Distractors	Difficult Distractors
ViT	86	50.3	78.7	27.7	8%	45%
Swin	86	52.9	78.2	30.7	9%	42%
DeiT	77.7	47.2	69.3	27.1	11%	43%
ConvNeXt	79	51.2	70.2	29.1	11%	43%

5 Conclusions

We introduced the VASR dataset for visual analogies of situation recognition. We automatically created over 500K analogy candidates, showing their quality via high inter-annotator agreement and their efficacy for training. Importantly, VASR test labels are human-annotated with high agreement. We showed that state-of-the-art models can solve our analogies with random distractors, but struggle with harder ones.

Acknowledgements

We would like to thank Timo Schick, Yanai Elazar, Leshem Choshen, Moran Mizrahi and Oren Sultan for their valuable feedback. This work was supported in part by the Center for Interdisciplinary Data Science Research at the Hebrew University of Jerusalem, and a research grant 2336 from the Israeli Ministry of Science and Technology. It was also supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant no. 852686, SIAM, Shahaf).

References

- Allen, C.; and Hospedales, T. M. 2019. Analogies Explained: Towards Understanding Word Embeddings. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 223–231. PMLR.
- Carey, S. 2011. Précis of the origin of concepts. *Behavioral and Brain Sciences*, 34(3): 113.
- Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 248–255. IEEE Computer Society.
- Depeweg, S.; Rothkopf, C. A.; and Jäkel, F. 2018. Solving bongard problems with a visual language and pragmatic reasoning. *ArXiv preprint*, abs/1804.04452.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv preprint*, abs/2010.11929.
- Dzendzik, D.; Vogel, C.; and Foster, J. 2021. English machine reading comprehension datasets: A survey. *ArXiv preprint*, abs/2101.10421.
- Evans, T. G. 1964. *A program for the solution of a class of geometric-analogy intelligence-test questions*. 64. Air Force Cambridge Research Laboratories, Office of Aerospace Research
- Falkeneheimer, B.; Forbus, K. D.; and Gentner, D. 1986. The structure mapping engine. In *Proceeding of the Sixth National Conference on Artificial Intelligence, Philadelphia, PA*.
- Fauconnier, G. 1997. *Mappings in thought and language*. Cambridge University Press.
- Fillmore, C. J.; Johnson, C. R.; and Petrucc, M. R. 2003. Background to framenet. *International journal of lexicography*, 16(3): 235–250.
- Forbus, K.; Usher, J.; Lovett, A.; Lockwood, K.; and Wetzel, J. 2011. CogSketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, 3(4): 648–666.
- Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2): 155–170.
- Gentner, D.; Holyoak, K. J.; and Kokinov, B. N. 2001. *The analogical mind: Perspectives from cognitive science*. MIT press.
- Goodman, N. D.; Tenenbaum, J. B.; and Gerstenberg, T. 2014. Concepts in a probabilistic language of thought. Technical report, Center for Brains, Minds and Machines (CBMM).
- Gupta, S.; and Malik, J. 2015. Visual semantic role labeling. *ArXiv preprint*, abs/1505.04474.
- Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B.; and Salesin, D. H. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 327–340.
- Hofstadter, D. R.; and Sander, E. 2013. *Surfaces and essences: Analogy as the fuel and fire of thinking*. Basic books.
- Huang, L.; Le Bras, R.; Bhagavatula, C.; and Choi, Y. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. 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)*, 2391–2401. Hong Kong, China: Association for Computational Linguistics.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338.

- Liang, Y.; Li, J.; and Yin, J. 2019. A new multi-choice reading comprehension dataset for curriculum learning. In *Asian Conference on Machine Learning*, 742–757. PMLR.
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.
- Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s. *ArXiv preprint*, abs/2201.03545.
- Lovett, A.; and Forbus, K. 2017. Modeling visual problem solving as analogical reasoning. *Psychological review*, 124(1): 60.
- Lu, H.; Liu, Q.; Ichien, N.; Yuille, A. L.; and Holyoak, K. J. 2019. Seeing the meaning: Vision meets semantics in solving pictorial analogy problems. In *Proceedings of the Annual Conference of the Cognitive Science Society*.
- McCarthy, J.; Minsky, M. L.; Rochester, N.; and Shannon, C. E. 2006. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4): 12–12.
- Memisevic, R.; and Hinton, G. E. 2010. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural computation*, 22(6): 1473–1492.
- Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. 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*, 746–751. Atlanta, Georgia: Association for Computational Linguistics.
- Miller, G. A. 1992. WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.
- Mitchell, M. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1): 79–101.
- Pratt, S.; Yatskar, M.; Weihs, L.; Farhadi, A.; and Kembhavi, A. 2020. Grounded situation recognition. In *European Conference on Computer Vision*, 314–332. Springer.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *ArXiv preprint*, abs/2103.00020.
- Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.
- Reed, S. E.; Zhang, Y.; Zhang, Y.; and Lee, H. 2015. Deep Visual Analogy-Making. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 1252–1260.
- Sadeghi, F.; Zitnick, C. L.; and Farhadi, A. 2015. Visalogic: Answering Visual Analogy Questions. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 1882–1890.
- Spelke, E. S.; and Kinzler, K. D. 2007. Core knowledge. *Developmental science*, 10(1): 89–96.
- Sun, K.; Yu, D.; Chen, J.; Yu, D.; Choi, Y.; and Cardie, C. 2019. DREAM: A Challenge Data Set and Models for Dialogue-Based Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 7: 217–231.
- Tewel, Y.; Shalev, Y.; Schwartz, I.; and Wolf, L. 2021. Zero-Shot Image-to-Text Generation for Visual-Semantic Arithmetic. *ArXiv preprint*, abs/2111.14447.
- Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.
- Winston, P. H. 1980. Learning and reasoning by analogy. *Communications of the ACM*, 23(12): 689–703.
- Yatskar, M.; Zettlemoyer, L. S.; and Farhadi, A. 2016. Situation Recognition: Visual Semantic Role Labeling for Image Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 5534–5542. IEEE Computer Society.
- Zellers, R.; Bisk, Y.; Schwartz, R.; and Choi, Y. 2018. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 93–104. Brussels, Belgium: Association for Computational Linguistics.

6 Appendix

6.1 License and Privacy

All images we use are taken from the SWiG dataset <https://github.com/allenai/swig> licensed under the MIT license. The VASR dataset is thus also licensed under the MIT license. We do not collect or publish players personal information

6.2 Reproducibility Checklist

Checklist

1. This paper Includes a conceptual outline and/or pseudocode description of AI methods introduced: **yes**
2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results: **yes**

3. Provides well marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper: **yes**
4. Does this paper make theoretical contributions? **no**
5. Does this paper rely on one or more datasets? **yes**
6. A motivation is given for why the experiments are conducted on the selected datasets: **yes**
7. All novel datasets introduced in this paper are included in a data appendix: **yes**
8. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes: **yes**
9. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations: **yes**
10. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available: **yes**
11. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying: **Not applicable**
12. Does this paper include computational experiments? **yes**
13. Any code required for pre-processing data is included in the appendix. **yes**
14. All source code required for conducting and analyzing the experiments is included in a code appendix. **yes**
15. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **yes**
16. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from. **yes**
17. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. **yes**
18. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. **yes**
19. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. **yes**
20. This paper states the number of algorithm runs used to compute each reported result. **yes**
21. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. **yes**
22. This paper lists all final (hyper-)parameters used for each model/algorithms in the paper's experiments. **yes**

Models Models are described in Section 4.2. Zero-shot models run in less than two hours, and trained models in less than 24 hours, on a single Tesla K80 GPU. Trained models hyper-parameters are provided in Section 4.3. Full implementation is provided in the attached code.

Statistics Dataset generation method is described in Section 3. Statistics are provided in Section 3.6. A link to a downloadable version of the dataset is available in the code (install.sh file). Complete description of the annotation process is provided in Section 3.5.

Code Full implementation, dependencies, training code, evaluation code, pre-trained models, README files and commands to reproduce the paper results are provided in the attach code.

6.3 Additional VASR Examples

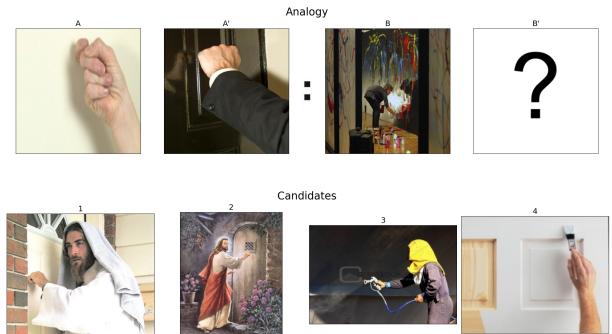


Figure 7: Answer - 4 (*wall* changed to *door*)

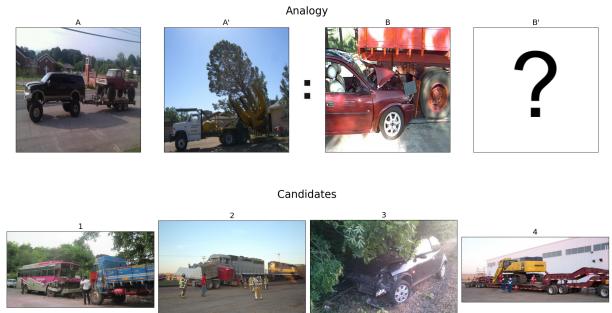


Figure 8: Answer - 3 (*truck* changed to *tree*)

6.4 Human Annotation

Figure 13 shows an example of the Mechanical Turk user-interface. The basic requirements for our annotation task is percentage of approved assignments above 98%, more than 5,000 approved HITs. To be a VASR annotator, we required additional qualification tests: We selected 10 challenging examples from VASR as qualification test. To be qualified we accepted annotators that received a mean accuracy score over 90%. The players received instructions (Section 6.5) and could do an interactive practice in the project website.

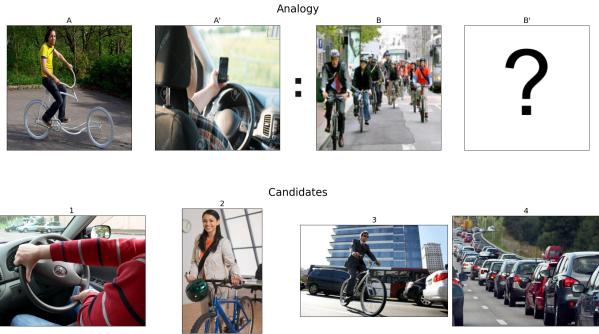


Figure 9: Answer - 4 (*bicycle changed to car*)

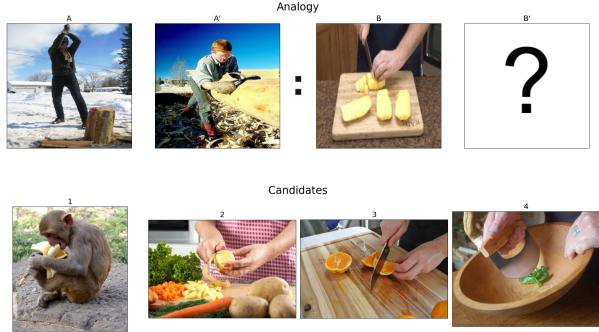


Figure 10: Answer - 2 (*cut changed to peel*)

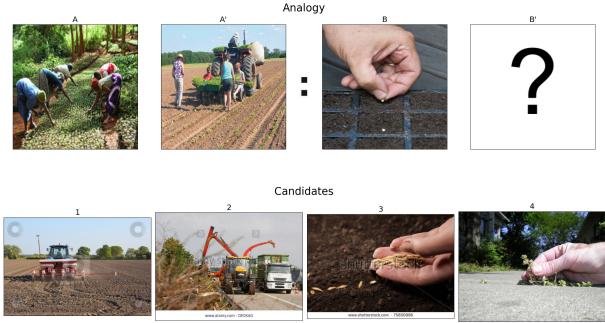


Figure 11: Answer - 1 (*hand changed to tractor*)

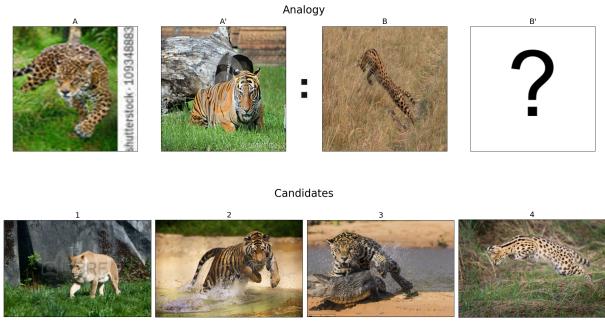


Figure 12: Answer - 2 (*leopard changed to tiger*)

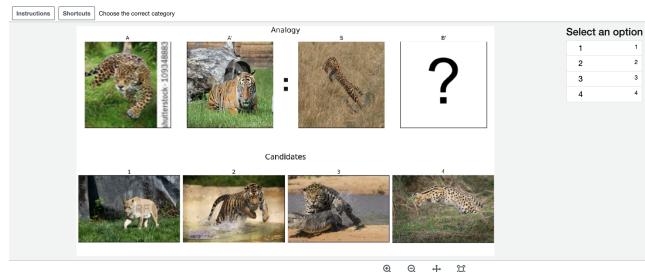


Figure 13: A screenshot from the annotator screen in Amazon Mechanical Turk.

6.5 Annotators Instructions

These are the instructions given to the annotators, accompanied by examples, and option to do an interactive practice in the project website: “In the following you are expected to solve an analogy problem. You will be shown three pictures: A, A’, B. There is some change going from picture A to picture A’ For example, A is a dog yawning and A’ is a baby yawning - the change is dog → baby.

You need to choose an option out of 4 images. Choose the image that best solves the analogy A is to A’ as B is to?

We recommend solving the analogies in computer, not mobile phone, as you’ll need to see the images in large screen to succeed.

In addition, while you are in the HIT interface (after the qualification), we suggest to zoom-out (using Ctrl key and press the - [minus] key) in order to see the image in better resolution.

To enter the full task, there will be a qualification test which requires a score of 100

For additional (interactive!) examples, you may refer to the project website vasr-dataset.github.io. Specifically, in the Explore Page you can learn on the different analogies in the dataset, and in the Test Page you can test yourself on 5 analogies, receiving a score.

To solve it, understand what is the key difference between A and A’, and map it to B.

It’s possible to have several differences between A and A’. Search for the difference that allows you to choose a candidate that solves the analogy.

The difference between the images is one of the roles in the image: (1) who is the agent in the image (man, horse, car, motorcycle, etc); (2) the verb or the activity the agent is doing (e.g.: a man nailing a nail); (3) the tool the agent is using (e.g., a man nailing a nail with a hammer); (4) the item that is effected by the agent (e.g., a man nailing a nail).”

6.6 Additional Statistics

6.7 Additional Figures

6.8 WordNet Concepts

We use the following list, covering most of the objects annotations in imSitu:

animal, person, group, male, female, creation, wheeled vehicle, system of measurement, structure, phenomenon,

	Silver	Gold		
	# Total	# Unique	# Total	# Unique
Objects	113,585	3,490	3,329	1,315
Verbs	38,664	1,989	491	160

Table 6: Analogies Transitions Statistics. For example in Figure 1, *man* changed to *monkey* is counted as a single object transition, and Figure 10, *cut* changed to *peel* is counted as a single verb transition.

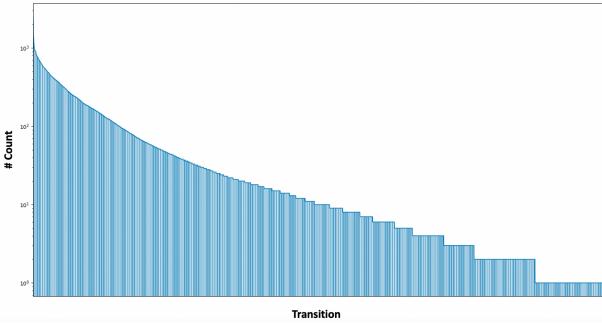


Figure 14: A visualization of all generated transitions (9,543). X axis is the transitions (e.g., *jumping* changed to *swimming*), and Y axis is logarithmic count.



Figure 15: VASR focuses on complex images describing scenes, such as the image on the left (a child feeding a calf), rather than simpler images such as the image on the right.



Figure 16: An example of non-visually salient object (2% of the image), which we aim to filter from VASR.

covering, celestial body, food, furniture, body of water, instrumentality, geographical area, round shape, plant, fire,

Figure 17: An example from VASR website that allows to users to interactively explore the different analogies in VASR. The following example presents an analogy of type *item*.

Figure 18: An example from VASR website that allows users to interactively solve analogies, receiving a grade and a feedback.

tube, educator, liquid, leaf, figure, substance, volcanic eruption, natural elevation, force, bird of prey, bovine, skeleton, male, female, body part, conveyance, utensil, dog, cat, rock, hoop, way, spiritual leader, spring, doll, plant part, piece of cloth, piece of cloth, plant organ, edible fruit, cord, jewelry, baseball, poster, javelin, cement, fabric, snow, football, ice, tape, screen, grave, plate, plastic, egg, collar, ribbon, rope, wool, glass, lumber, cake, powder, sink, balloon, mushroom

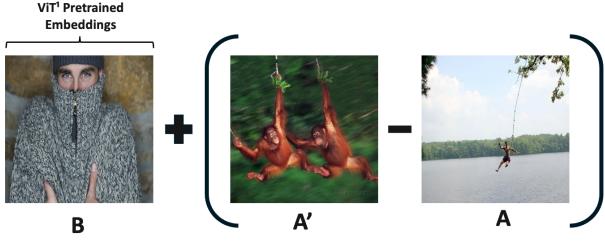


Figure 19: Zero-shot arithmetic model sketch. Given four candidates C_1, C_2, C_3, C_4 ,

$$prediction = \operatorname{argmax}_i (\text{sim}(B + (A' - A), C_i))$$

The pretrained embeddings are obtained from some pretrained model, such as ViT, Swin Transformer, DEiT and ConvNeXt. We perform vector arithmetic $B + A' - A$, and select the candidate that is most similar (cosine-similarity) to the received representation.

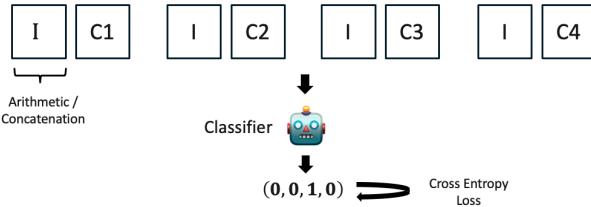


Figure 20: Supervised model sketch. We denote with “I” the input representation, which can be both the arithmetic representation ($B + A' - A$) or the concatenation representation (A, A', B). To classify an image out of four candidates, we concatenate the input to each of the candidates, receiving an output vector, and extracts the cross-entropy loss to train the model.