IMAGINATOR

A Pre-Trained Image+Text Joint Embedding

Anonymous ACL submission

Abstract

Word embeddings, i.e., semantically meaningful vector representation of words, are largely influenced by the distributional hypothesis *"You shall know a word by the company it keeps"* (Harris, 1954), whereas modern prediction-based neural network embeddings rely on design choices and hyperparameter optimisation. Word embeddings like Word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) etc. well capture the contextuality and real-world analogies but contemporary convolution-based image embeddings such as VGGNet (Simonyan and Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), etc. do not capture contextual knowledge. The popular *king-queen* analogy does not hold true for most commonly used vision embeddings.

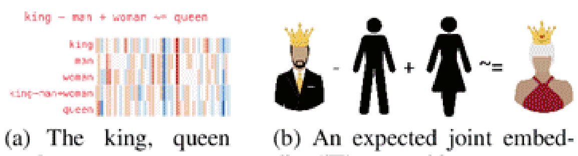
In this paper, we introduce a pre-trained joint embedding (JE), named IMAGINATOR, trained on 21K distinct image objects level from 1M image+text pairs. JE is a way to encode multimodal data into a vector space where the text modality serves as the grounding key, which the complementary modality (in this case, the image) is anchored with. IMAGINATOR encapsulates three individual representations: *(i) object-object collocation, (ii) word-object collocation, and (iii) word-object correlation*. These three ways capture complementary aspects/knowledge of the two modalities which are further combined to obtain the final JEs. We evaluate pre-trained IMAGINATOR JEs on three distinct tasks: (i) image captioning, (ii) Image2Tweet (Jha et al., 2021), and (iii) text-based image retrieval. IMAGINATOR establishes a new standard on the aforementioned downstream tasks by outperforming the current SoTA on all the selected tasks.

Generated JEs are also intrinsically evaluated to assess how well they capture the contextuality and real-world analogies - based on word analogies and using corresponding images. IMAGINATOR will be made publicly available.

1 Joint Modality and Contextuality

Word embeddings are learned representations such that words with similar meanings are represented similarly. Distribution-based compositional word embeddings like Word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) are popular in modern NLP. These are used to extract the notion of relatedness across different words, and capture the overall semantic meaning of a text. Consider the *king-queen* (Mikolov et al., 2013b) word vector analogy (figure 1), which shows how good these word embeddings are at capturing syntactic and semantic regularities in language.

The notion of contextual similarity (i.e., words occurring together) is used in learning the representations, because of which vector arithmetic like `King − Man + Woman = Queen` are possible. See Fig. 1 (Mikolov et al., 2013b). Deriving an analogous representation using images is a challenging task since the concept of relatedness among images is not well-defined. Motivated by this argument, we propose creating JEs that can represent real-world analogies, which can aid in solving several multimodal tasks owing to their distributional semantics.



(a) The king, queen analogy

(b) An expected joint embedding(JE) space: king_{JE} - queen_{JE} $\approx$ $\textit{boy}_{JE}$ - girl_{JE}

Figure 1: CNN-based image embeddings are unable to capture contextuality like existing word embeddings. The *king-queen* vs. *man-woman* analogy has been popularized by (Mikolov et al., 2013b), whereas drawing a similar analogy in image vector space is rather impossible. We argue joint embedding is the alternative.

2 Contemporary Joint Embedding Methods

Canonical Correlation Analysis (CCA)-based methods use similarities to project two inputs onto a vector space. CLIP (Radford et al., 2021) utilizes contrastive pre-training and encodes aligned image and text embeddings with the help of text and visual modality encoders. Stanford's Joint Embedding (Kolluru, 2019) uses VGG-19 (Simonyan and Zisserman, 2014) and GLoVe (Pennington et al., 2014) to generate the image and text encodings using a triplet loss. (Chen et al., 2020) proposed UNITER, trained on a large dataset, which uses an image and text encoder and a transformer to generate the final embeddings. Jia et al. (2021) use a noisy dataset of 1 billion (image, alt-text) pairs and propose a dual architecture for aligning and generating the visual and textual embeddings. This architecture uses contrastive loss for learning. Tan and Bansal (2019) proposed a framework to create a relation between visual and language modalities. This architecture consists of three encoders, one object relation encoder, a language encoder and a cross-modal encoder. Compared to the aforementioned prior works- illustrated in appendix figure 10, the unique differentiating factor with IMAGINATOR is that we focus on the word-level grounding of images while prior works perform embedding generation at the sentence level. Our belief is that this will help us learn rich relational features, i.e., features that are rich encapsulations of words and the corresponding objects they represent via images.

3 IMAGINATOR - learning joint embeddings

Off the shelf, word embeddings like Word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) are prevalently used in modern NLP. Furthermore, pre-trained language models such as BERT (Devlin et al., 2018), GPT (Radford and Narasimhan, 2018), etc. use such pre-trained word embeddings to tackle several downstream NLP tasks. The motivation behind creating IMAGINATOR is something kindred. Researchers can download pre-trained JEs and utilize them for any task they have in hand. Existing techniques have only explored JEs from the sentence-level perspective, which makes it less flexible to repurpose them for other tasks, but most importantly, demands a lot more data for the model to understand and derive meaningful relationships. We thus operate at the word level rather than sentence-level, to help improve the "sharpness" of the data, with the hope that this would, in turn, help synthesize higher relational features that can offer optimal performance on downstream tasks. To that end, we make some simple assumptions and posit arguments on their choice as better alternatives.

3.1 Object vs. word - a unit hypothesis

The smallest meaningful unit of text is a word, which we assume signifies a visual object embedded in an image. Albeit, the common trend is to train end-to-end network on sentence-level, but system may not be able to learn fine grained contextual relations like *king-queen* analogy. This design choice also aligns with our motivation to generate general-purpose JEs suited for a wide variety of downstream tasks (c.f. section 6).

3.1.1 Number of Objects

The number of objects in available datasets like Flickr30k (Young et al., 2014) and COCO (Lin et al., 2014) is limited only to a few hundred. However, if we are interested to learn real-world analogies like *king-queen* analogy we require far more real-life objects to be detected by the system. Detic (Zhou et al., 2022), a recent object detection technique, provides 21K object classes and thus, seems the most pertinent. Results shown in table 8 indicate that an incremented in the number of objects leads to a corresponding increase the accuracy. Table 1 compares the number of objects supported among models popular in the research community.

Model	No. of objects
YOLO X (Ge et al., 2021)	80
Inception V3 (Szegedy et al., 2016)	600
SWIN-L (Liu et al., 2021)	1000
OSCAR Plus (Zhang et al., 2021)	1843
Detic (Zhou et al., 2022)	**21K**

Table 1: Objects detection capabilities of popularly used models.

3.2 Learning JE

Based on the unit hypothesis, we capture three aspects of the input data while generating joint embeddings:

- Object-object collocation: v_{oo}
- Word-object collocation: v_{wo}
- Word-object correlation: v_{wor}

More about each component and their embeddings in the upcoming sections.

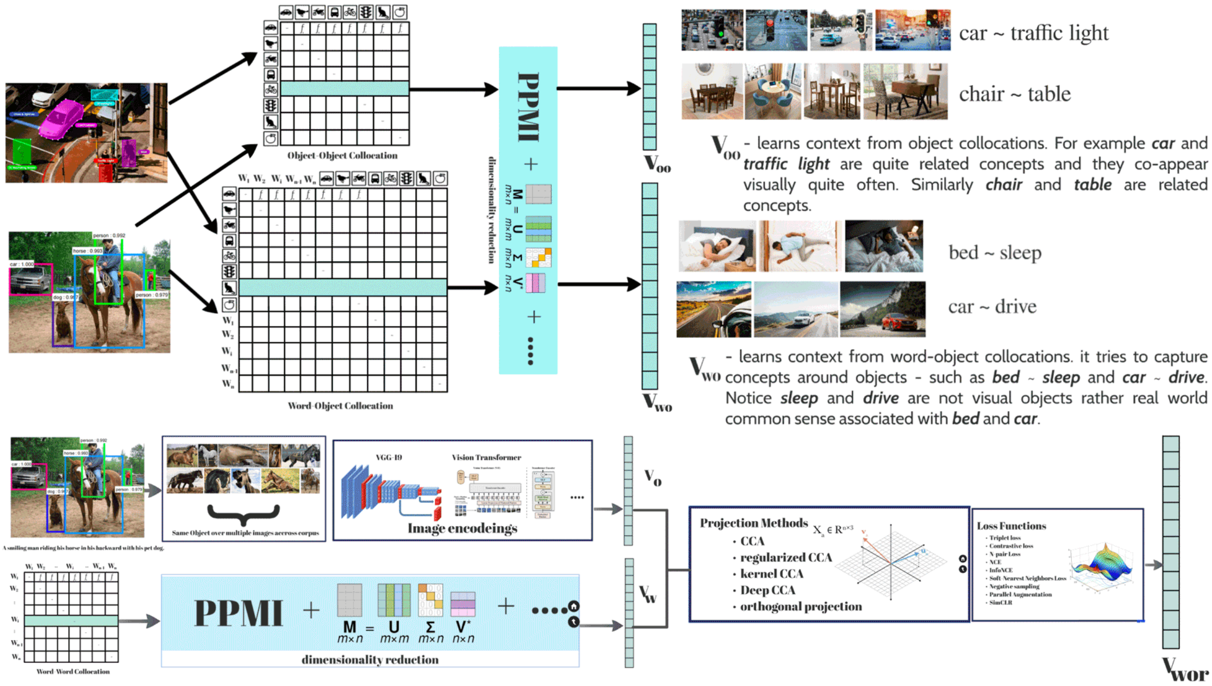


Figure 2: (Top) Architecture for creating text embeddings and v_{oo} and v_{wo}: the rows and columns in the collocation matrix are the words from the text or objects detected from the images from dataset. Each cell of this matrix represents the occurrence count of each row-column pair in the dataset. The two final vectors are generated using PPMI and eigenvalue weighting over the vectors from collocation matrices. (Bottom) Architecture for learning v_{wor}: (left) the averaged VGG19 representation of a particular object across the whole dataset is passed; (right) word2vec representation of the word (i.e., the name of the visual object; for e.g., *horse* in this case).

3.3 Learning v_{oo} and v_{wo}

Figure 2 (top) offers a visual summary of the process of generating object-object collocation embeddings v_{oo} and word-object collocation embeddings v_{wo}. v_{oo} and v_{wo} are learned using an object collocation matrix, where objects refer to the entities detected using an object detection model. Object collocation matrix is a matrix where the rows and columns correspond to objects detected in our images and each cell represents the co-occurrences of the respective two objects. We then take the rows and apply dimensionality reduction techniques like SVD along with Eigenvalue weighting, the vector obtained is then used as the embedding. This yields *object-object collocation*, which encodes how frequently a detected object co-appears with other detected objects in the dataset. On the other hand, *word-object collocation* is built using the objects from object detection on images and the words from the associated text given in the datasets. This might seem similar to object-object collocation at first glance, but a major difference is that the value in each cell represents the number of image captions having the corresponding object and word pair. With this collocation matrix, we get information on how frequently every object co-appears with other words in the dataset.

3.4 Learning v_{wor}

Figure 2 (bottom) unpacks the process of generating the word-object correlation embeddings v_{wor}. v_{wor} is learnt using a different approach when compared with the other two embeddings. Collocation can be defined using the co-occurance of two entities but correlation calls for a deeper understanding of the two entities. Therefore, we get joint embeddings for word-object correlation using word and object vectors.

We generate object embeddings by passing all detected crops of the object from the dataset to VGG19 (Simonyan and Zisserman, 2014). An average of these embeddings across all instances give us the final embedding for the object encoded as a mean representation. The word embeddings are acquired by creating a *word-word collocation* matrix for the text in the dataset, kindred to the aforementioned collocation matrices, where each cell represents the number of co-occurrences of the corresponding word pair.

To obtain the final joint embedding from these

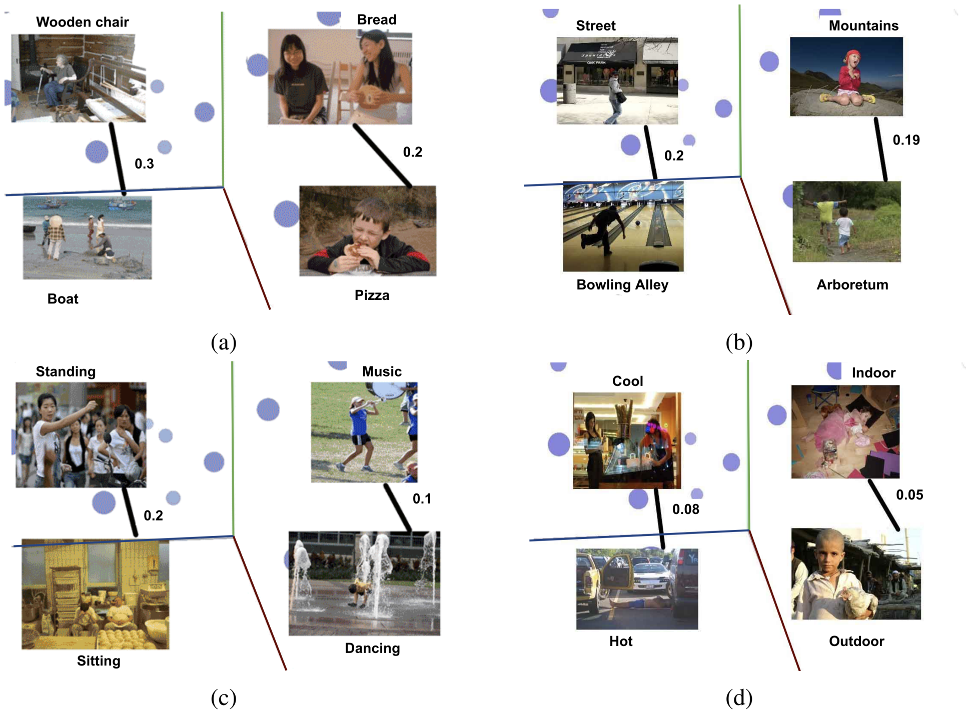


Figure 3: The IMAGINATOR JE vector space captures real-world analogies well. These examples are taken from similar word examples and these depicted distances are vector space euclidean distances.

two vectors, we project the object embedding in the word embedding space instead of projecting both embeddings in a common space (Kolluru, 2019; Radford et al., 2021). The motivation behind this is to maintain the contextuality captured in word embeddings and thus enforce the object embeddings to learn the correlations. Once they learn a correlated vector space, we get the JEs from a weighted average of the projected word and object embedding. We perform experiments to compare several projection methods (such as CCA (Thompson, 2000), Kernel CCA (Hardoon et al., 2004), Deep CCA (Andrew et al., 2013) etc.) and loss functions (InfoNCE (Oord et al., 2018), contrastive loss (Hadsell et al., 2006), and triplet loss (Schroff et al., 2015)). Empercally, we find that orthogonal projection and triplet loss give the best JE results. We believe CCA overfits on our data while orthogonal projection (Artetxe et al., 2018) uses the features based on the dataset size. Please refer to table 8 in Appendix for more on these experiments and their results.

4 Lessons learnt from NLP

Word embeddings are learnt in two major ways: (i) classical count based methods, and (ii) neural network based prediction methods. Levy et al. (2015) argue that the performance gains of neural network based word embeddings are due to certain system design choices and hyperparameter optimizations, rather than the embedding algorithms themselves. Furthermore, they show that these modifications can be transferred to traditional distributional models, yielding similar gains. In contrast to prior reports, they show mostly local or insignificant performance differences between the methods, with no global advantage to any single approach over the others. Therefore, we remain grounded to count-based distributional semantics methods. Raw counts or normalized counts are not useful, rather we choose alternatives like PMI and SVD.

4.1 PPMI and Context Distribution Smoothing

The PPMI (Positive Pointwise Mutual Information) between a word and its context is well known to be an effective association measure in the word similarity literature. (Levy et al., 2015) show that the skip-gram with negative-sampling training method (SGNS) is implicitly factorizing a word-context matrix whose cell values are essentially shifted PMI. Following their analysis, we present two variations of the PMI (and implicitly PPMI) association metric, which we adopt from SGNS. In this section, w and c represent the word and context matrix.

Shifted PMI. The shift caused by $1 < k$ (the number of negative samples in the optimization (w, c): $PMI(w, c) - log(k)$) can be applied to distributional methods through shifted PPMI (Levy and Goldberg, 2014):

The k here, firstly, estimates negative sample distribution and secondly, acts as a prior on the probability of an occurrence of (w, c) in the corpus vs. a negative sample. Shifted PPMI captures the latter, i.e, the prior aspect of k.

$$SPPMI(w, c) = max(PMI(w, c) - log(k), 0) \quad (1)$$

Context Distribution Smoothing (CDS). Word2Vec (Mikolov et al., 2013a) samples negative samples according to a smoothed unigram distribution. This smoothing variation has an analog when calculating PMI directly:

$$PMI_\alpha(w, c) = log\frac{\hat{P}(w, c)}{\hat{P}(w).\hat{P}_\alpha(c)} \quad (2)$$

$$PMI_\alpha(c) = \frac{\#(c)^\alpha}{\Sigma_c \#(c)^\alpha} \quad (3)$$

By enlarging the probability of sampling a rare context (since $\hat{P}_\alpha(c) > \hat{P}(c)$ when c is infrequent), CDS reduces the PMI of (w, c) for a rare context c – thus removing PMI's bias towards rare words.

4.2 SVD and Eigenvalue Weighting (eig)

Word and context vectors derived using SVD of collocation matrices can be represented by:

$$W^{SVD} = U_d \cdot \Sigma_d \quad C^{SVD} = V_d \quad (4)$$

However, in this case, C^{SVD} is orthonormal while W^{SVD} is not. Factorization achieved by SGN is much more symmetric and a similar symmetry can be derived using the following factorization:

$$W = U_d \cdot \sqrt{\Sigma_d} \quad C = V_d \cdot \sqrt{\Sigma_d} \quad (5)$$

Levy et al. (2015) states that while it is not theoretically clear why a symmetric approach performs better for semantic tasks, it works empirically.

For our vector-deriving implementation, we use this as a dimensionality reduction technique. It is similar to SVD but instead of the usual representation: $W = U.\Sigma_d$ and $C = V_d$, eigenvalue weighting uses $W = U.\Sigma_d^{0.5}$ and $C = V_d$. To summarize, after creating the collocation matrix, we derive vectors by initially applying SPPMI with CGS. This is followed by the SVD of the matrices with eigenvalue weighting.

5 Merging v_{oo}, v_{wo}, and v_{wor}

The three vectors can be merged using approaches such as concatenation, averaging or autoencoding (figure 4). Autoencoder is a pertinent research topic where merging of a number of vectors is learnt automatically by a trained model. This approach considers learning the embeddings by considering complementary information from it's source embeddings. In the interest of simplifying this aspect of our design, for our experiments, we use weighted average to combine the embeddings. The weights are decided in an experimental fashion. The best weights we found are 10, 10, and 80 for v_{oo}, v_{wo}, and v_{wor} respectively.

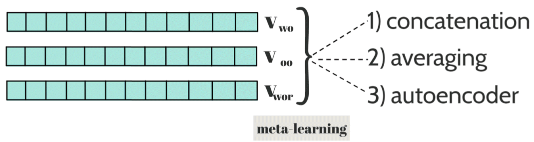


Figure 4: Merging the three representational vectors - v_{oo}, v_{wo}, and v_{wor}

6 Intrinsic evaluation of IMAGINATOR

To be able to make vector arithmetic like `King - Man + Woman = Queen` in a generated word vector space is well known as the intrinsic evaluation paradigm. Contemporary image embeddings are devoid of contextuality, whereas text embeddings are much more meaningful, as shown in figure 1. With joint embeddings, we aim to add a contextual component to improve the semantic richness of the joint embeddings vector space. We use two kinds of intrinsic evaluation setup to evaluate IMAGINATOR: (i) word analogy, and (ii) similar object analogy. Word contextuality is evaluated using different word analogy methods while image contextuality is evaluated using a similarity metric between images of objects that are alike.

6.1 Word Analogies

We adopt the process and all the 10 datasets mentioned in (Jastrzebski et al., 2017) to evaluate IMAGINATOR against word analogy using three intrinsic evaluation tasks: (i) word similarity, (ii) word analogy, and (iii) word categorization. The word embeddings for given similar words from datasets of all three tasks are computed. We use average euclidean distance to derive the final results, as shown in table 2 for embeddings from GloVe (Pennington et al., 2014), CLIP (Radford

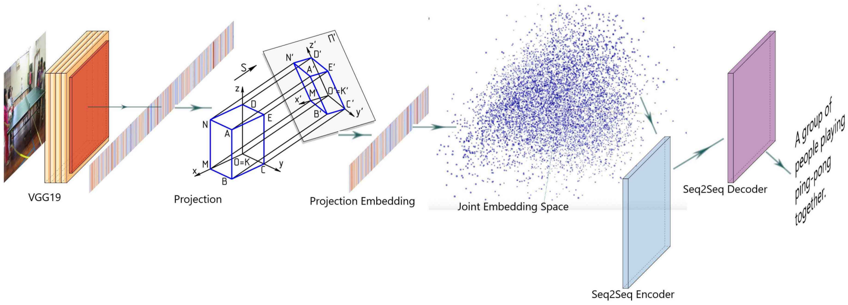


Figure 5: Image captioning using IMAGINATOR.

Dataset	GloVe	CLIP	SJE	ALIGN	Ours
WS353 (Finkelstein et al., 2001)	2.65	0.35	0.19	0.09	0.14
MTurk (Halawi et al., 2012)	1.99	0.29	0.28	0.09	0.20
RG65 (Rubenstein et al., 1965)	0.75	0.38	0.20	0.14	0.13
RW (Pilehvar et al., 2018)	0.96	0.19	0.17	0.10	0.25
SimLex999 (Hill et al., 2015)	2.31	0.18	0.22	0.14	0.12
MEN (Bruni et al., 2014)	0.51	0.22	0.13	0.12	0.11
Google Analogy (Mikolov et al., 2013a)	2.09	0.18	0.12	0.09	0.15
MSR Analogy (Mikolov et al., 2013b)	0.63	0.30	0.09	0.11	0.22
SemEval2012 (Jurgens et al., 2012)	1.2	0.21	0.32	0.13	0.26
BLESS (Baroni and Lenci, 2011)	2.77	0.22	0.19	0.12	0.11
Average	1.5	0.25	0.19	0.12	**0.16**

Table 2: Results (average euclidean distance) for intrinsic valuation of our JEs based on notable word analogy methods. Lower is better.

Dataset	SJE	CLIP	ALIGN	IMAGINATOR
Flickr 30K	0.8	0.4	0.2	0.06
MS COCO	0.9	1.3	0.7	0.2
Google CC	0.2	0.4	0.2	0.08
Visual Genome	1.1	1.4	0.9	0.1
Average	0.75	0.88	0.5	**0.11**

Table 3: Results (average pairwise euclidean distance) for instrinsic valuation of our JEs based on object similarity on objects from multiple datasets. Lower is better.

et al., 2021), ALIGN (Jia et al., 2021), and IMAGINATOR.

While SJE is competitive and ALIGN is better on word similarity compared to IMAGINATOR, but IMAGINATOR is significantly better on image object similarity/analogy, detailed in the next section.

6.2 Image Similarity

We use the Caltech 101 (Li et al., 2022) dataset objects for this evaluation. Figure 3 shows several examples of the relation between projected joint embeddings of these objects. Analogy-making on images is relatively challenging. Our hypothesis is vectors of the same/similar objects must be nearby in the IMAGINATOR vector space. We take out list of similar objects from the the Caltech 101 dataset, apply VGG19 and then orthogonally (Artetxe et al., 2018) project those objects to IMAGINATOR vector space. Then we calculate the pairwise-euclidean distance between such vectors and average them for the entire dataset. Table 4 offers a summary with the relative upside of IMAGINATOR compared to all the existing alternatives VGG19, SJE, CLIP, and ALIGN respectively listed next to each score. Furthermore, table 3 shows the performance on existing vision embedding techniques and IMAGINATOR on other similar datasets.

VGG19	SJE	CLIP	ALIGN	IMAGINATOR
5.6 (97% ↑)	1.9 (93% ↑)	1.5 (91% ↑)	0.92 (86% ↑)	**0.13**

Table 4: Results (average pairwise euclidean distance) for intrinsic valuation of our JEs based on object similarity on the Caltech 101 dataset. Lower is better.

7 IMAGINATOR for - Image Captioning and Image2tweet

The downstream vision-language (VL) tasks chosen to test our pre-trained JEs are: (i) image captioning, (ii) Image2Tweet, and (iii) text-based image retrieval.

7.1 Image Captioning

Image captioning is a common multimodal task which involves the generation of a textual description for an image. Describing the contents of an image requires visual understanding at an object level. We use JEs from IMAGINATOR to generate captions on datasets such as Flickr30k (Young et al., 2014) and COCO (Lin et al., 2014).

For an input image, we start by obtaining an image embedding using VGG19 (Simonyan and Zisserman, 2014), which is then orthogonally projected in IMAGINATOR embedding space. We use the JE of the image to find k nearest objects in the vector space. For our experiments we used $k = 10$, giving us 10 objects associated with the

(a)
IMAGINATOR: A kitchen with a sink, stove, oven, and beers.
Gold Caption: A commercial stainless kitchen with a pot of food cooking.
OSCAR: a kitchen with a lot of pots and pans in it.

(b)
IMAGINATOR: A vocalist, drummer, and a guitarist sings a tune.
Gold Caption:A musical band are by their instruments most likely playing a song .
OSCAR: A man on a stage with a guitar and a keyboard.

Figure 6: Examples of some image captioning outputs generated by IMAGINATOR along with the original caption and the caption generated by OSCAR (Li et al., 2020). For more examples please refer the supplementary file.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 7: Image retrieved for the query - *"several climbers climbing rock together"* - it is evident that ALBEF (Li et al., 2021) wrongly emphasized on *"rock together"*, whereas XVLM (Zeng et al., 2021) is unable to comprehend plurality in the query here, while BERT$_{IMAGINATOR}$ can do the job well.

input image. These objects are then passed to a sequence-to-sequence module, namely, the T5 transformer (Bhatia, 2021), which generates the final caption. We use a pre-trained T5 model, fine-tuned on Flickr30k and COCO. Figure 5 describes the captioning pipeline while Figure 6 shows some output examples. IMAGINATOR surpasses the current SoTA by 3.1 BLEU (Flickr30K) and 1.4 BLEU (COCO), as shown in Table 5. We also evaluated our captions using BERTScore and achieved 0.87 on Flickr30k and 0.88 on COCO, there is no BERTScore reported in UVLP (Zhou et al., 2020) and OSCAR (Li et al., 2020).

Method	Flickr30K	COCO
UVLP (Zhou et al., 2020)	30.1 (SoTA)	-
OSCAR (Li et al., 2020)	-	41.7 (SoTA)
SJE + k nearest objects + T5	**30.5**	**35.6**
CLIP + k nearest objects + T5	**31.3**	**36.3**
IMAGINATOR + k nearest objects + T5	**33.2**	**43.1**

Table 5: Comparison of different modelling approaches in image captioning.

7.2 Image2Tweet

Image2Tweet (Jha et al., 2021) is a task which is a notch above traditional image captioning in terms of complexity. Given an input image, the task involves generating a tweet like a human news reporter. Figure 24 shows some examples from the dataset.

The tweet is generated using a method similar to image captioning. The joint embedding of the input image is used to find the k nearest neighbouring embeddings in the projections space. These neighbours are then used to generate the tweet using a sequence-to-sequence model.

The results are based on the CIDEr metric (c.f. table 6). We found that using other datasets for training SoTA models fails miserably, indicating that Image2Tweet is a fairly complex problem. However, IMAGINATOR performs reasonably well on the task and surpasses comparable contemporary SoTA captioning methods UVLP (Zhou et al., 2020) and OSCAR (Li et al., 2020).

Method	CIDEr
Baseline of Image2tweet (Jha et al., 2021)	0.0003
UVLP (Zhou et al., 2020) [SoTA on Flickr]	0.003
OSCAR (Li et al., 2020) [SoTA on COCO]	0.004
CLIP (Radford et al., 2021)	0.006
Stanford joint embedding (Kolluru, 2019)	0.007
5 ensemble (Luo et al., 2018)	0.0090
IMAGINATOR + k nearest objects + T5	**0.0095**

Table 6: Performance of various multi-modal models in the Image2Tweet shared task.

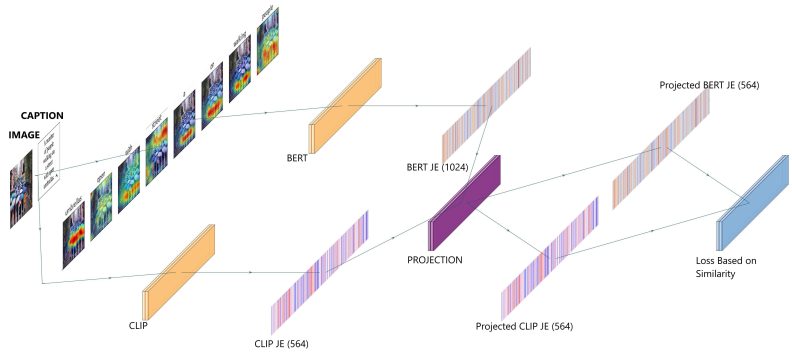


Figure 8: $BERT_{IMAGINATOR}$ - Training approach for Image Retrieval. The cosine similarity is used to train. Training happens in batches and similarity between corresponding image-sentence pair is maximised while for other pairs it is minimized.

7.3 Text-based Image retrieval

The fundamental question that we are seeking an answer to is whether using IMAGINATOR word-object level embeddings we can achieve compositionally and achieve a vector representation for sentence-image level. For example, by passing on word vectors in a sequence to a language model we can obtain a sentence-level vector representation. To verify the compositionality of joint modality embeddings, we tested our approach on the task of text-based image retrieval on the Flickr30K dataset (Young et al., 2014). The main challenge of this task is to find out the appropriate content in the visual space while the input is in the text space. Another reason for introducing compositionality is that each word is usually associated with multiple images. Hence, there is a need for us to learn a single image representation for a given text. Though we explore contrastive methods in section 8, to solve the above-mentioned challenges, we introduce an approach using BERT and evaluate it on text-based image retrieval.

Method	R@1	R@5	R@10
ALBEF (Li et al., 2021)	85.6	97.5	98.9
XVLM (Zeng et al., 2021)	86.1	97.3	98.7
BERT$_{IMAGINATOR}$	**89.48**	**98.1**	**99.2**

Table 7: Results for image retrieval: Recall@$\{1, 5, 10\}$ Score for Flickr30K (Young et al., 2014).

7.3.1 Compositionality of Joint Embeddings - $BERT_{IMAGINATOR}$

BERT is arguably the most successful modelling architectures in NLP. It accepts token embeddings as input and produces contextualized embeddings as output. In contrast, we propose $BERT_{IMAGINATOR}$, which is trained to take image+text as input and output a compositional vector representation for both modalities.

We utilize BERT (Devlin et al., 2018) and CLIP (Radford et al., 2021) as our backbones to generate JEs. Instead of feeding the BERT model tokenized words obtained via a tokenizer, we use IMAGINATOR (c.f. section 3.4) word-object embeddings as input to the model. We process necessary tokenization, position encoding, and segment embeddings accordingly, per the BERT architecture.

We utilize CLIP (Radford et al., 2021) for generating another JE using an image-sentence pair by obtaining the image and text embeddings from CLIP encoders and concatenating them. We refer to this as the *sentence JE*. Both these embeddings, viz., the *sentence JE* and *projected* $BERT_{IMAGINATOR}$, are projected to a common space using orthogonal projection (Artetxe et al., 2018), on which we compute our loss. Figure 8 visually depicts our training process while table 7 shows BERT$_{IMAGINATOR}$ outperforming SoTA information retrieval (IR) baselines, namely ALBEF (Li et al., 2021) and XLVM (Zeng et al., 2021) on Recall@$\{1, 5, 10\}$.

8 Conclusion and Takeaways

We proposed a new pre-trained joint embedding IMAGNIATOR. Our major contribution is on adopting count-based methods for joint modality, echoing the philosophy from Levy et al. (2015). IMAGINATOR outperformed SoTA on three tasks: *(i) image captioning, (ii) Image2Tweet, and (iii) text-based image retrieval*. We present an in-depth intrinsic evaluation along with a new architecture $BERT_{IMAGINATOR}$. In the future, we would like to explore other multimodal tasks such as VQA.

Discussion and Limitations

While IMAGINATOR pushes the boundaries of the state-of-the-art in tasks that involve language and vision joint modelling, there are some limitations.

Object Detection - Limited Number of Classes

IMAGINATOR utilizes the atomic units of multi-modal data – individual words for text representation and individual objects for image representation. Typically, the number of unique words (i.e., the vocabulary) is quite large in a given text relative to the number of objects in images. As such, IMAGINATOR being a joint learning technique is bottlenecked by the capabilities of existing object detection techniques since they only typically deal with a limited repertoire of objects (c.f. table 1). To enhance the richness and expressivity of JEs, object detection models that can identify the wide gamut of objects in the world would be critical.

Contrastive Learning

Contrastive learning is a task-independent technique that focuses in learning the similarity and differences between samples in a dataset. The objective here is to learn an embedding space where similar inputs, say samples belonging to the same class (c.f. figure 9), are embedded as similar representations while samples from dissimilar classes are separated in the embedding space. Despite IMAGINATOR performing well in several tasks, our object representation can be improved significantly from simply being an average of image embeddings. Instead, contrastive learning can help learn better vectors that capture the relations between images and their objects.

Figure 9: Same object class (horse) with different visuals.

Vision Transformer and Positional Encoding

A Vision Transformer (ViT) is a transformer that is targeted at vision processing tasks, such as object recognition and is much more robust than CNNs. It divides an image into fixed-size patches, embeds each of them, and includes a positional embedding along with the patch embedding as an input to the transformer encoder. In our case, if we could draw meaningful cross-modal connections between sections of text and the corresponding parts of images, a significant performance uptick can be potentially reached. This can be implemented using the various positional encoding schemes in ViT.

References

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK. Association for Computational Linguistics.

Gagan Bhatia. 2021. keytotext.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 406–414. ACM.

John Rupert Firth. 1957. *"A synopsis of linguistic theory 1930-1955."*. Oxford University Press.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 1406–1414, New York, NY, USA. Association for Computing Machinery.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

Zellig Harris. 1954. Distributional Structure. *Word*, 10:146–162.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Stanislaw Jastrzebski, Damian Lesniak, and Wojciech Marian Czarnecki. 2017. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *CoRR*, abs/1702.02170.

Rishabh Jha, Varshith Kaki, Varuna Kolla, Shubham Bhagat, Parth Patwa, Amitava Das, and Santanu Pal. 2021. Image2tweet: Datasets in Hindi and English for generating tweets from images. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 670–676, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. 2012. SemEval-2012 task 2: Measuring degrees of relational similarity. 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 356–364, Montréal, Canada. Association for Computational Linguistics.

Sethu Hareesh Kolluru. 2019. A neural architecture to learn image-text joint embedding.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the association for computational linguistics*, 3:211–225.

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. 2022. Caltech 101.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Di Lu, Spencer Whitehead, Lifu Huang, Heng Ji, and Shih-Fu Chang. 2018. Entity-aware image caption generation. *arXiv preprint arXiv:1804.07889*.

Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. 2018. Discriminability objective for training descriptive captions. *arXiv preprint arXiv:1803.04376*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. 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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Mohammad Taher Pilehvar, Dimitri Kartsaklis, Victor Prokhorov, and Nigel Collier. 2018. Card-660: Cambridge rare word dataset - a reliable benchmark for infrequent word representation models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1391–1401, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Herbert Rubenstein, John B. Goodenough, and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Bruce Thompson. 2000. Canonical correlation analysis.

Jianfeng Wang et al. 2022. Recent advances in vision-and-language pre-training (tutorial).

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Yan Zeng, Xinsong Zhang, and Hang Li. 2021. Multi-grained vision language pre-training: Aligning texts with visual concepts.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. 2022. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*.

Frequently Asked Questions (FAQs)

1. Doesn't averaging individual object embeddings (or even word embeddings) result in a noisy object embedding?
   Ans. Yes, averaging individual embeddings is a limitation of this work and a future avenue of exploration. On the other hand, concatenation is computationally expensive. However, empirically, we found that averaging gave better results than concatenation. We would like to explore autoencoding and contrastive learning in the future as mitigation methods.

2. Why does orthogonal projection work better than CCA-based methods?
   Ans. Orthogonal projection is a discriminative method that attempts to find out the discriminative projection for two vector spaces aligned to a unified dimension. On the other hand, CCA tries to learn relations among two vector spaces. While orthogonal projection offers competitive performance with a limited number of classes, CCA is undoubtedly more powerful when the number of classes is higher. In our case, since we only have 21K objects, orthogonal projection yielded better results.

3. Instead of directly learning a caption generation model based on the learned joint embedding, this paper projects VGG-19 embeddings orthogonally in the learned joint embedding space, using it to find the k nearest objects in the vector space, and then passes these objects through T5 for caption generation. What is the motivation behind this approach?
   Ans. Image object detection is a separate task altogether, and we are not trying to solve that problem here. Given an image, we first get its VGG-19 embedding and then project it to IMAGINATOR space since VGG-19 and IMAGINATOR have disparate embedding spaces and need to be aligned. A by-product of this approach is that it also helps affirm that IMAGINATOR performs well, otherwise it might raise doubts that the performance gain is happening due to T5 efficiency rather than IMAGINATOR. Lastly, we would like to draw the attention of readers that the proposed captioning architecture is very simple and still outperforms SoTA.

4. Did you consider experimenting with ResNet or Fast-RCNN?
   Ans. We performed experiments using ResNet, but the results were poor. One plausible explanation is the fact that higher embedding dimensions lead to a performance drop.

5. Why was the Detic the baseline architecture of choice for IMAGINATOR?
   Ans. The presumption of this work is to leverage the legacy of the NLP-centric count-based vectorification methods for joint modality. Therefore, maximizing the number of objects will give us a denser matrix to calculate the so-called collocation. In the future, we plan to seek methods that can detect more than 21K objects, and strongly believe that will have a positive effect on the learned joint embedding space.

Appendix

What is the value-addition of this work given Joint Embeddings have been explored in various ways for the past few years?

Learning joint embeddings has been a topic that has received immense interest from the multimodal AI community over the past decade (Chen et al., 2020; Jia et al., 2021; Tan and Bansal, 2019). A concise survey on this topic has been presented in Wang et al. (2022), which offers an extensive treatment of both early (i.e., input-level), mid (i.e., feature-level), and late (i.e., decision-level) fusion methods, depicted visually in figure 10. Learning joint embeddings using early-fusion methods (like the one we adopted in our work) essentially enables identifying cross-correlations between various modalities (such as text, images, video, audio, spatial/point-cloud information, etc.) early on in the learning process. As such, the resultant vector representations typically lead to top-notch performance in most downstream tasks. On the other hand, a vast majority of work focuses on feature fusion where modalities are first individually processed and then projected to a common vector space to draw correlations using variety of projection methods like CCA (Thompson, 2000), Deep CCA (Andrew et al., 2013), etc.

As mentioned in Section 2, IMAGINATOR's novelty is associated with the word-level grounding of objects using traditional count-based approaches, an NLP tradition that was prevalent before the neural era. This is a significant detour from recent work in learning joint embeddings that uses deep learning-based techniques, which suffer from a lack of control or ease of interpretation owing to their inherent black-box nature. As such, this design decision has allowed us to learn rich features that are collocation-based representations of visual objects that are grounded in words which represent the object's moniker(s). The collocation-based contextual word vectorization is primarily influenced by the distributional hypothesis *"You shall know a word by the company it keeps"* (Firth, 1957). Intrigued by how such a collocation-based method can aid visual contextual learning, we sought to testify its utility in learning joint emebddings. However, the ability of count-based joint embedding techniques can be severely limited due to the insufficient number of objects detected, which led to us overcoming this issue by using statistical correlational methods inspired by NLP (Levy et al. (2015). We plan to further scale this technique by first enabling detection of additional (>20K) visual objects, hypothesizing that this learning paradigm can lead to even richer representations.

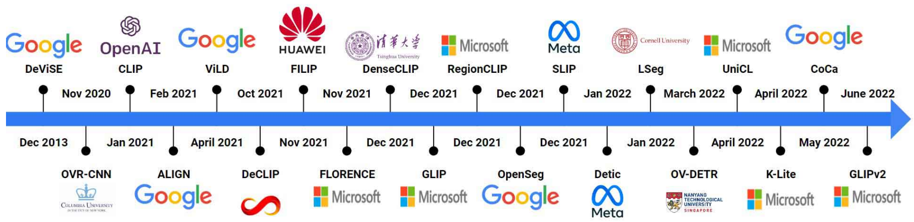


Figure 10: Notable recent work related to vision-language pre-training. Taken from Wang et al. (2022).

Rolodex of additional experiments carried out for the optimal generation of v_{oo}, v_{wo}, and v_{wor}

Table 8 expands on section 3.3 and 3.4 and shows a comparison of different embedding methods, dataset combinations, number of objects, loss functions, and projection methods with performance on the captioning task. We consider normalized count, PMI, PPMI, and Factorized PPMI for vector building and SVD and Eigenvalue factorization for dimensionality reduction. For dataset, we combine multiple datasets with an increase in the number of objects in each of the combined datasets. We consider Flickr30K, COCO, and Conceptual Captions (CC). For projection, we consider Orthogonal Projection, CCA, regularized CCA, and Deep CCA for our experiments. From Table 8, we can see that with an increase in the number of objects, the captioning score also correspondingly increases.

Embedding method	Dataset	No. of objects	Loss function	Projection method	Performance	
					Flickr30K	COCO
Normalized count	Flickr30K	1000	Triplet loss	Orthogonal	32.1	29.3
	Flickr30K + COCO	1080	Triplet loss	Orthogonal	32.4	33.7
PMI	Flickr30K + COCO	17000	Triplet Loss	-	33.4	34.8
PPMI	Flickr30K + COCO	17000	Triplet Loss	-	33.9	35.4
Factorized PPMI	Flickr30K + COCO	17000	Triplet Loss	-	34.1	37.9
Factorized PPMI + SVD	Flickr30K + COCO	17000	Triplet Loss	-	32.4	36.2
Factorized				CCA	30.2	32.2
PPMI	Flickr30K	1000	Triplet Loss	Regularized CCA	30.9	33.9
+				Deep CCA	30.5	33.2
Eigen	Flickr30K + COCO	1080	Triplet loss	Orthogonal	31.9	35.7
Value	Flickr30K + COCO	17000	Triplet loss	Regularized CCA	33.2	40.1
Factorization	Flickr30K + COCO	17000	Triplet loss	Deep CCA	33.8	38.1

Table 8: Results on image captioning datasets (Flickr30K, COCO, and Conceptual Captions) for different embedding methods, datasets, loss functions, and projection methods.

Intrinsic evaluation of IMAGINATOR

The goal behind intrinsic evaluation is to understand how well the embeddings adhere to the contextuality constraint. Building upon section 6, we consider standard relational terms - king, queen, boy, woman and performed an intrinsic evaluation on them to identify the relationships between these terms. We project the joint embeddings of the image and check the Euclidean distance among them; the lower the distance between similar terms, the better the contextuality. Figure 11 shows additional intrinsic evaluation examples.

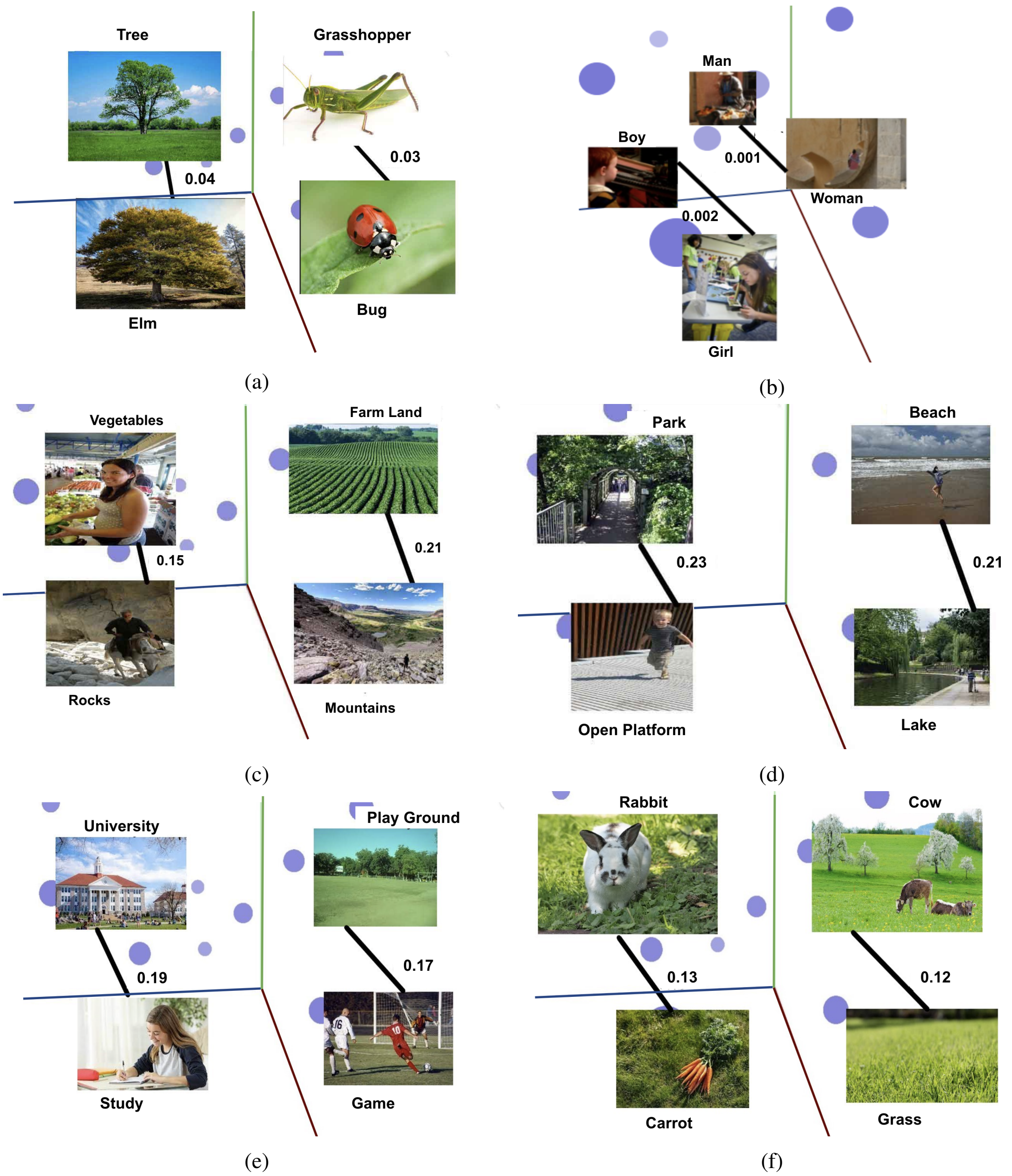


Figure 11: Examples for intrinsic evaluation of our JEs based on word analogies and image similarity.

Examples of image captioning - IMAGINATOR vs. Gold Caption vs. OSCAR (SoTA)

(a) **IMAGINATOR**: A bedroom with bedspreads, pillows, and a nightstand.
Gold Caption: the - bedroom stone cottage can sleep people.
OSCAR: A bedroom with a bed, dresser, and nightstand.

(b) **IMAGINATOR**: A group of people are singing and clapping while a group of musicians are performing.
Gold Caption: A band is playing in front of an audience and the singer is wearing an orange shirt.
OSCAR: A man holding a baseball bat in front of a crowd.

(c) **IMAGINATOR**: Photograph of a tall tower with steeples.
Gold Caption: sandcastle beach on bright sky.
OSCAR: A castle made of sand with a clock tower in the background.

(d) **IMAGINATOR**: A photo of a console.
Gold Caption: The player staring intently at a computer screen.
OSCAR: A man sitting in front of a flat screen TV.

(e) **IMAGINATOR**: A group of people playing ping-pong together.
Gold Caption: Young girls line up across each other and a ping-pong table in a gymnasium while a few boys plan on a table further back.
OSCAR: A group of children playing a game of ping pong.

(f) **IMAGINATOR**: "girls" and "boys" at a venue.
Gold Caption: party in the park under cherry blossoms.
OSCAR: A group of people sitting around a park with pink flowers.

(g) **IMAGINATOR**: A young woman and man rowing a boat.
Gold Caption: A man and woman are on a gray and white rowboat.
OSCAR: Group of people on a small boat in the water.

(h) **IMAGINATOR**: A kitchen with cabinets, cabinets, and a dishwasher.
Gold Caption: A kitchen with wooden cabinets and black appliances
OSCAR: A kitchen with a sink, dishwasher, stove and refrigerator.

(i) **IMAGINATOR**: Chefs cooking with a stover and other cookware in a laboratory.
Gold Caption: Two chefs in a restaurant kitchen preparing food .
OSCAR: Two men in a commercial kitchen preparing food.

Figure 12: Examples of some image captioning outputs generated by IMAGINATOR along with the original caption and the caption generated by OSCAR (Li et al., 2020) for each respective image

Examples of image retrieval - IMAGINATOR vs. SoTA: (i) ALBEF (Li et al., 2021), and (ii) XVLM (Zeng et al., 2021)

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 13: Image retrieved for the query: *"Two little children, one boy and one girl laughing"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 14: Image retrieved for the query: *"A dog is running in the sand"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 15: Image retrieved for the query: *"Bride and groom walking side by side"*.

(a) rank 1 image ALBEF (Li et al., 2021)

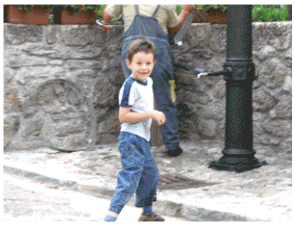
(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 16: Image retrieved for the query: *"Redhead woman in pig- tails and glasses sewing on a sewing machine"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)



(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 17: Image retrieved for the query: *"Smiling boy in white shirt and blue jeans in front of rock wall with man in overalls behind him"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 18: Image retrieved for the query: *"Two Asian or Spanish people, a woman and a man, sitting together in front of a glass window as cars pass"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 19: Image retrieved for the query: *"A little boy plays with a Nintendo GameCube controller inside a McDonald's"*.

(a) rank 1 image ALBEF (Li et al., 2021)

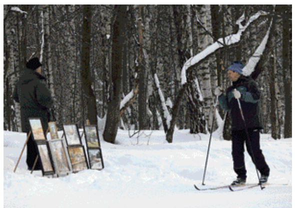
(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 20: Image retrieved for the query: *"A blonde woman wearing glasses and a gray sweatshirt is cutting something with scissors"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

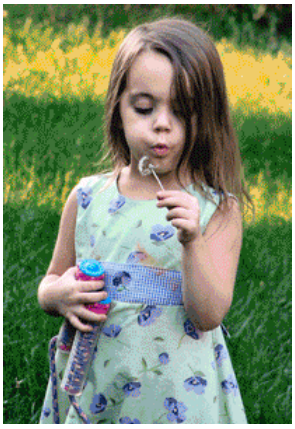


(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 21: Image retrieved for the query: *"A person wearing skis looking at framed pictures set up in the snow"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)



(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 22: Image retrieved for the query: *"A very young girl playing with a bubble-blowing wand , holding a bottle of bubble solution and walking through a park or field"*.

(a) rank 1 image ALBEF (Li et al., 2021)

(b) rank 1 image XVLM (Zeng et al., 2021)

(c) rank 1 image BERT$_{IMAGINATOR}$

Figure 23: Image retrieved for the query: *"A woman in an outdoor marketplace, wearing a large cone-shaped hat, standing behind two large baskets containing loaves of bread"*.

Image2Tweet examples - Gold vs. 5 ensemble SoTA (Lu et al., 2018) vs.$BERT_{IMAGINATOR}$

Image2tweet is a particularly hard problem to solve. It can involve social engineering, web information scraping, face recognition, etc. The results in table 6 show the current status of the problem and it needs substantial research work to develop a solution. Figure 24 shows additional Image2Tweet examples which indicate the quality of text generated given the image. Compared with the novel information present in the image, shows the amount of complexity associated with the task.

(a) **Gold Caption**: Should you wear a mask to protect yourself from coronavirus? #Coronavirus #COVID19
5 ensemble (Luo et al., 2018): a group of surgeons prepare for surgery.
IMAGINATOR: people wearing masks during the pandemic.

(b) **Gold Caption**: Donald Trump's India visit will be beneficial for both the countries.
5 ensemble (Luo et al., 2018): politician shakes hands with politician during a bilateral meeting.
IMAGINATOR: Two men are handshaking with an Indian flag in the background.

(c) **Gold Caption**: I am here to play cricket not gimmick - @PrithviShaw to press.
5 ensemble (Luo et al., 2018): cricket player during a press conference.
IMAGINATOR: A man in a press conference.

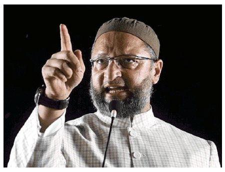
(d) **Gold Caption**: JEE (Main) begins today - students are following protocols - queue, social distancing, masks.
5 ensemble (Luo et al., 2018): students wearing face masks during a protest.
IMAGINATOR: young girls wearing masks in a queue.

(e) **Gold Caption**: Country needs so many doctors than politicians - pandemic realization.
5 ensemble (Luo et al., 2018): person, left, and person, right, are both members of the team.
IMAGINATOR: Two doctors with face shields.

(f) **Gold Caption**: 5G tech is picking up pace and expectations are high, but rollout is still years away in India.
5 ensemble (Luo et al., 2018): the logo on a background of a blue sky with clouds.
IMAGINATOR: 5G logo.

(g) **Gold Caption**: SC refuses to entertain plea against Madras HC order on Patanjali's use of 'Coronil'.
5 ensemble (Luo et al., 2018): a gothic buildiing.
IMAGINATOR: supreme court of India building.

(h) **Gold Caption**: No rugby for world champion as South Africa maintains ban.
5 ensemble (Luo et al., 2018): rugby player looks dejected after defeat
IMAGINATOR: A scene of a rugby match with three players visible.



(i) **Gold Caption**: I love India, but Indians don't like me.
5 ensemble (Luo et al., 2018): politician addresses a crowd of supporters.
IMAGINATOR: An angry politician delivering a speech.

(j) **Gold Caption**: Indian prime minister addressing to the nation in his own man ki baat.
5 ensemble (Luo et al., 2018): politician making a speech at a function.
IMAGINATOR: Modi is delivering a speech on camera.

(k) **Gold Caption**: Kamala Harris bringing energy, dollars and more to Joe Biden's campaign.
5 ensemble (Luo et al., 2018): politician gives a speech during the second day.
IMAGINATOR: Harris making promises.

(l) **Gold Caption**: US Presidential election: Hillary-Tulsi spat scorches Democratic Party.
5 ensemble (Luo et al., 2018): Two politicians are debating.
IMAGINATOR: Hillary Clinton and another woman in white dress.

Figure 24: Additional examples of Image2Tweet task - gold vs. 5 ensemble SoTA (Li et al., 2020)