VisualSparta: An Embarrassingly Simple Approach to Large-scale Text-to-Image Search with Weighted Bag-of-words

Xiaopeng Lu[*]
Language Technologies Institute
Carnegie Mellon University
xiaopen2@andrew.cmu.edu

Tiancheng Zhao, Kyusong Lee
SOCO Inc
Pittsburgh, USA
{tianchez,kyusongl}@soco.ai

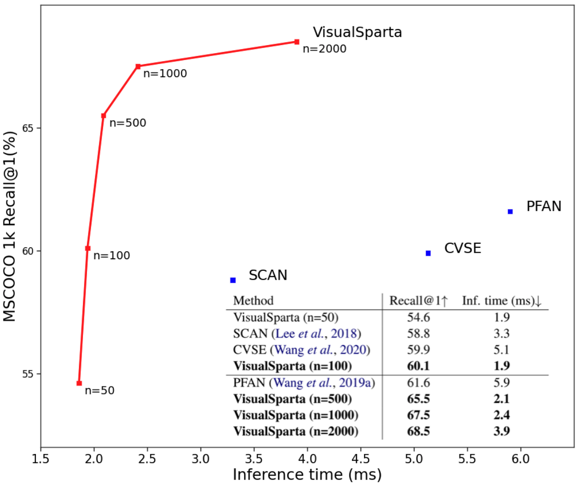
Abstract

Text-to-image retrieval is an essential task in cross-modal information retrieval, i.e., retrieving relevant images from a large and unlabelled dataset given textual queries. In this paper, we propose VisualSparta, a novel (**Visual**-text **Spar**se **T**ransformer M**a**tching) model that shows significant improvement in terms of both accuracy and efficiency. VisualSparta is capable of outperforming previous state-of-the-art scalable methods in MSCOCO and Flickr30K. We also show that it achieves substantial retrieving speed advantages, i.e., for a 1 million image index, VisualSparta using CPU gets ~391X speedup compared to CPU vector search and ~5.4X speedup compared to vector search with GPU acceleration. Experiments show that this speed advantage even gets bigger for larger datasets because VisualSparta can be efficiently implemented as an inverted index. To the best of our knowledge, VisualSparta is the first transformer-based text-to-image retrieval model that can achieve real-time searching for large-scale datasets, with significant accuracy improvement compared to previous state-of-the-art methods.

1 Introduction

Text-to-image retrieval is the task of retrieving a list of relevant images from a corpus given text queries. This task is challenging because in order to find the most relevant images given text query, the model needs to not only have good representations for both textual and visual modalities, but also capture the fine-grained interaction between them.

Existing text-to-image retrieval models can be broadly divided into two categories: query-agnostic and query-dependent models. The dual-encoder architecture is a common query-agnostic model, which uses two encoders to encode the query and images separately and then compute the relevancy via inner product (Faghri *et al.*, 2017; Lee *et al.*, 2018; Wang *et al.*, 2019a). The transformer architecture is a well-known query-dependent model (Devlin *et al.*, 2018; Yang *et al.*, 2019). In this case, each pair of text and image is encoded by concatenating and passing into one single network, instead of being encoded by two separate encoders (Lu *et al.*, 2020; Li *et al.*, 2020b). This method borrows the knowledge from large pretrained transformer models and shows much better accuracy compared to dual-encoder methods (Li *et al.*, 2020b).



Method	Recall@1↑	Inf. time (ms)↓
VisualSparta (n=50)	54.6	1.9
SCAN (Lee *et al.*, 2018)	58.8	3.3
CVSE (Wang *et al.*, 2020)	59.9	5.1
VisualSparta (n=100)	**60.1**	**1.9**
PFAN (Wang *et al.*, 2019a)	61.6	5.9
VisualSparta (n=500)	**65.5**	**2.1**
VisualSparta (n=1000)	**67.5**	**2.4**
VisualSparta (n=2000)	**68.5**	**3.9**

Figure 1: **Inference Time vs. Model Accuracy.** Each dot represents Recall@1 for different models on MSCOCO 1K split. By setting top *n*-terms to 500, our model significantly outperforms the previous best query-agnostic retrieval models, with ~2.8X speedup. See section 5.1 for details.

Besides improving the accuracy, retrieval speed has also been a long-existing subject of study in the information retrieval (IR) community (Manning *et al.*, 2008). Query-dependent models are prohibitively slow to apply to the entire image corpus because it needs to recompute for every dif-

[*] This work was partially done during an internship at SOCO

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 5020–5029
August 1–6, 2021. ©2021 Association for Computational Linguistics

ferent query. On the other hand, query-agnostic model is able to scale by pre-computing an image data index. For dual-encoder systems, further speed improvement can be obtained via Approximate Nearest Neighbors (ANN) Search and GPU acceleration (Johnson *et al.*, 2019).

In this work, we propose VisualSparta, a simple yet effective text-to-image retrieval model that outperforms all existing query-agnostic retrieval models in both accuracy and speed. By modeling fine-grained interaction between visual regions with query text tokens, our model is able to harness the power of large pre-trained visual-text models and scale to very large datasets with real-time response. To our best knowledge, this is the first model that integrates the power of transformer models with real-time searching, showing that large pre-trained models can be used in a way with significantly less amount of memory and computing time. Lastly, our method is embarrassingly simple because its image representation is essentially a weighted bag-of-words, and can be indexed in a standard Inverted Index for fast retrieval. Comparing to other sophisticated models with distributed vector representations, our method does not depend on ANN or GPU acceleration to scale up to very large datasets.

Contributions of this paper can be concluded as the following: **(1)** A novel retrieval model that achieves new state-of-the-art results on two benchmark datasets, i.e., MSCOCO and Flickr 30K. **(2)** Weighted bag-of-words is shown to be an effective representation for cross-modal retrieval that can be efficiently indexed in an Inverted Index for fast retrieval. **(3)** Detailed analysis and ablation study that show advantages of the proposed method and interesting properties that shine light for future research directions.

2 Related Work

Large amounts of work have been done on learning a joint representation between texts and images (Karpathy and Fei-Fei, 2015; Huang *et al.*, 2018; Lee *et al.*, 2018; Wehrmann *et al.*, 2019; Li *et al.*, 2020b; Lu *et al.*, 2020). In this section, we revisit dual-encoder based retrieval model and transformer-based retrieval model.

2.1 Dual-encoder Matching Network

Most of the work in text-to-image retrieval task choose to use the dual-encoder network to encode information from text and image modalities. In Karpathy and Fei-Fei (2015), the author used a Bi-directional Recurrent Neural Network (BRNN) to encode the textual information and used a Region Convolutional Neural Network (RCNN) to encode the image information, and the final similarity score is computed via the interaction of features from two encoders. Lee *et al.* (2018) proposed stacked cross-attention network, where the text features are passed through two attention layers to learn interactions with the image region. Wang *et al.* (2019a) encoded the location information as yet another feature and used both deep RCNN features (Ren *et al.*, 2016) and the fine-grained location features for the Region of Interest (ROI) as image representation. In Wang *et al.* (2020), the author utilized the information from Wikipedia as an external corpus to construct a Graph Neural Network (GNN) to help model the relationships across objects.

2.2 Pre-trained Language Models (PLM)

Large pre-trained language models (PLM) show great success over multiple tasks in NLP areas in recent years (Devlin *et al.*, 2018; Yang *et al.*, 2019; Dai *et al.*, 2019). After that, research has also been done on cross-modal transformer-based models and proves that the self-attention mechanism also helps jointly capture visual-text relationships (Li *et al.*, 2019; Lu *et al.*, 2020; Qi *et al.*, 2020; Li *et al.*, 2020b). By first pretraining model under large-scale visual-text dataset, these transformer-based models capture rich semantic information from both texts and images. Models are then fine-tuned for the text-to-image retrieval task and show improvements by a large margin. However, the problem of using transformer-based models is that it is prohibitively slow in the retrieval context: the model needs to compute pair-wise similarity scores between all queries and answers, making it almost impossible to use the model in any real-world scenarios. Our proposed method borrows the power of large pre-trained models while reducing the inference time by orders of magnitude.

PLM has shown promising results in Information Retrieval (IR), despite its slow speed due to the complex model structure. The IR community recently started working on empowering the classical full-text retrieval methods with contextualized information from PLMs (Dai and Callan, 2019; MacAvaney *et al.*, 2020; Zhao *et al.*, 2020). Dai and Callan (2019) proposed DeepCT, a model that learns to generate the query importance score from the contextualized representation of large transformer-based models. Zhao *et al.* (2020) proposed sparse transformer matching model (SPARTA), where the model learns term-level interaction between query and text answers and generates weighted term representations for answers during index time. Our work is motivated by works in this direction and extends the scope to the cross-modal understanding and retrieval.

3 VisualSparta Retriever

In this section, we present VisualSparta retriever, a fragment-level transformer-based model for efficient text-image matching. The focus of our proposed model is two-fold:

- Recall performance: fine-grained relationship between queries and image regions are learned to enrich the cross-modal understanding.
- Speed performance: query embeddings are non-contextualized, which allows the model to put most of the computation offline.

3.1 Model Architecture

3.1.1 Query representation

As query processing is an online operation during retrieval, the efficiency of encoding query needs to be well considered. Previous methods pass the query sentence into a bi-RNN to give token representation provided surrounding tokens (Lee *et al.*, 2018; Wang *et al.*, 2019a, 2020).

Instead of encoding the query in a sequential manner, we drop the order information of the query and only use the pretrained token embeddings to represent each token. In other words, we do not encode the local contextual information for the query and purely rely on independent word embedding E_{tok} of each token. Let a query be $q = [w_1, ..., w_m]$ after tokenization, we have:

$$\hat{w}_i = E_{tok}\left(w_i\right) \tag{1}$$

where w_i is the i-th token of the query. Therefore, a query is represented as $\hat{w} = \{\hat{w}_1, ..., \hat{w}_m\}, \hat{w}_i \in \mathbb{R}^{d_H}$. In this way, each token is represented independently and agnostic to its local context. This is essential for the efficient indexing and inference, as described next in section 3.3.

3.1.2 Visual Representation

Compared with query information which needs to be processed in real-time, answer processing can be rich and complex, as answer corpus can be indexed offline before the query comes. Therefore, we follow the recent works in Vision-Language Transformers (Li *et al.*, 2019, 2020b) and use the contextualized representation for the answer corpus.

Specifically, for an image, we represent it using information from three sources: regional visual features, regional location features, and label features with attributes, as shown in Figure 2.

Regional visual features and location features Given an image v, we pass it through Faster-RCNN (Ren *et al.*, 2016) to get n regional visual features v_i and their corresponding location features l_i:

$$v_1, ..., v_n = \mathrm{RCNN}(v), v_i \in \mathbb{R}^{d_{rcnn}} \tag{2}$$

and the location features are the normalized top left and bottom right positions of the region proposed from Faster-RCNN, together with the region width and height:

$$l_i = [l_{xmin}, l_{xmax}, l_{ymin}, l_{ymax}, l_{width}, l_{height}] \tag{3}$$

Therefore, we represent one region by the concatenation of two features:

$$E_i = [v_i; l_i] \tag{4}$$

$$E_{image} = [E_1, ..., E_n], E_i \in \mathbb{R}^{d_{rcnn}+d_{loc}} \tag{5}$$

where E_{image} is the representation for a single image.

Label features with attributes Additional to the deep representations from the proposed image region, previous work by Li *et al.* (2020b) shows that the object label information is also useful as an additional representation for the image. We also encode the predicted objects and corresponding attributes obtained from Faster-RCNN model with pretrained word embeddings:

$$\hat{o}_i = E_{tok}(o_i) + E_{pos}(o_i) + E_{seg}(o_i) \tag{6}$$

$$E_{label} = [\hat{o_1}, ..., \hat{o_k}], \hat{o}_i \in \mathbb{R}^{d_H} \tag{7}$$

where k represents the number of tokens after the tokenization of attributes and object labels for n

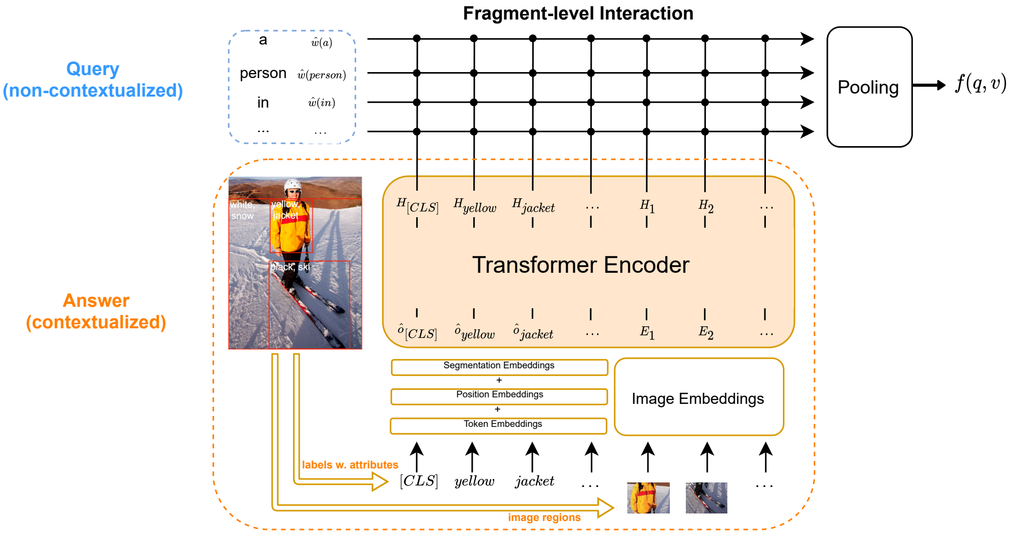


Figure 2: VisualSparta Model. It first computes contextualized image region representation and non-contextualized query token representation. Then it computes a matching score between every query token and image region that can be stored in an inverted index for efficient searching.

image regions. E_{tok}, E_{pos}, and E_{seg} represent token embeddings, position embeddings, and segmentation embeddings respectively, similar to the embedding structure in Devlin *et al.* (2018).

Therefore, one image can be represented by the linear transformed image features concatenated with label features:

$$a = [(E_{image}W + b); E_{label}] \tag{8}$$

where $W \in \mathbb{R}^{(d_{rcnn}+d_{loc})\times d_H}$ and $b \in \mathbb{R}^{d_H}$ are the trainable linear combination weights and bias. The concatenated embeddings a are then passed into a Transformer encoder T_{image}, and the final image feature is the hidden output of it:

$$H_{image} = T_{image}(a) \tag{9}$$

where $H_{image} \in \mathbb{R}^{(n+k)\times d_H}$ is the final contextualized representation for one image.

3.1.3 Scoring Function

Given the visual and query representations, the matching score can now be computed between a query and an image. Different from other dual-encoder based interaction model, we adopt the fine-grained interaction model proposed by Zhao *et al.* (2020) to compute the relevance score by:

$$y_i = \max_{j\in[1,n+k]}(\hat{w}_i^T H_j) \tag{10}$$

$$\phi(y_i) = \text{ReLU}(y_i + b) \tag{11}$$

$$f(q,v) = \sum_{i=1}^{m} \log(\phi(y_i) + 1) \tag{12}$$

where Eq.10 captures the fragment-level interaction between every image region and every query word token; Eq.11 produces sparse embedding outputs via a combination of ReLU and trainable bias, and Eq.12 sums up the score and prevents an overly large score using log operation.

3.2 Retriever training

Following the training method presented in Zhao *et al.* (2020), we use cross entropy loss to train VisualSparta. Concretely, we maximize the objective in Eq. 13, which tries to decide between the ground truth image v^+ and irrelevant/random images V^- for each text query q. The parameters to learn include both the query encoder E_{tok} and the image transformer encoder T_{image}. Parameters are optimized using Adam (Kingma and Ba, 2014).

$$J = f(q, v^+) - \log \sum_{k\in V^-} e^{f(q,k))} \tag{13}$$

In order to achieve efficient training, we use other image samples from the same batch as negative examples for each training data, an effective technique that is widely used in response selection (Zhang *et al.*, 2018; Henderson *et al.*, 2019). Preliminary experiments found that as long as the batch size is large enough (we choose to use batch size of 160), this simple approach performs equally well compared to other more sophisticated methods, for example, sample similar images that have nearby labels.

3.3 Efficient Indexing and Inference

VisualSparta model structure is suitable for real-time inference. As discussed in section 3.1.1, since query embeddings are non-contextualized, we are able to compute the relationship between each query term w_i and every image v offline.

Concretely, during offline indexing, for each image v, we first compute fragment-level interaction between its regions and every query term in the vocabulary, same as in Eq. 10. Then, we cache the computed ranking score:

$$\text{CACHE}(w, v) = \text{Eq. } 11 \tag{14}$$

During test time, given a query $q = [w_1, ..., w_m]$, the ranking score between q and an image v is:

$$f(q, v) = \sum_{i=1}^{m} \log(\text{CACHE}(w_i, v) + 1) \tag{15}$$

As shown in Eq. 15, the final ranking score during inference time is an O(1) look-up operation followed by summation. Also, the query-time computation can be fit into an Inverted Index architecture (Manning *et al.*, 2008), which enables us to use VisualSparta index with off-the-shelf search engines, for example, Elasticsearch (Gheorghe *et al.*, 2015).

4 Experiments

4.1 Datasets

In this paper, we use MSCOCO (Lin *et al.*, 2014)[1] and Flickr30K (Plummer *et al.*, 2015)[2] datasets for the training and evaluation of text-to-image retrieval tasks. MSCOCO is a large-scale multi-task dataset including object detection, semantic segmentation, and image captioning data. In this experiment, we follow the previous work and use the image captioning data split for text-to-image model training and evaluation. Following the experimental settings from Karpathy and Fei-Fei (2015), we split the data into 113,287 images for training, 5,000 images for validation, and 5,000 images for testing. Each image is paired with 5 different captions. The performance of 1,000 (1K) and 5,000 (5K) test splits are reported and compared with previous results.

Flickr30K (Plummer *et al.*, 2015) is another publicly available image captioning dataset, which contains 31,783 images in total. Following the split from Karpathy and Fei-Fei (2015), 29,783 images are used for training, and 1,000 images are used for validation. Scores are reported based on results from 1,000 test images.

For speed experiments, in addition to MSCOCO 1K and 5K splits, we create 113K split and 1M split, two new data splits to test the performance in the large-scale retrieval setting. Since these splits are only used for speed experiments, we directly reuse the training data from the existing dataset without the concern of data leaking between training and testing phases. Specifically, the 113K split refers to the MSCOCO training set, which contains 113,287 images, ∼23 times larger than the MSCOCO 5K test set. The 1M split consists of one million images randomly sampled from the MSCOCO training set. Speed experiments are done on these four splits to give comprehensive comparisons under different sizes of image index.

4.2 Evaluation Metrics

Following previous works, we use recall rate as our accuracy evaluation metrics. In both MSCOCO and Flikr30K datasets, we report Recall@*t*, *t*=[1, 5, 10] and compare with previous works.

For speed performance evaluation, we choose query per second and latency(ms) as the evaluation metric to test how each model performs in terms of speed under different sizes of image index.

4.3 Implementation Details

All experiments are done using the PyTorch library. During training, one NVIDIA Titan X GPU is used. During speed performance evaluation, one NVIDIA Titan X GPU is used for models that need GPU acceleration. One 10-core Intel 9820X CPU is used for models that needs CPU acceleration. For the image encoder, we initialize the model weights from Oscar-base model (Li *et al.*, 2020b) with 12 layers, 768 hidden dimensions, and 110M parameters. For the query embedding, we initialize it from the Oscar-base token embedding. The Adam optimizer (Kingma and Ba, 2014) is used with the learning rate set to 5e-5. The number of training epochs is set to 20. The input sequence length is set to 120, with 70 for labels with attributes features and 50 for deep visual features. We search on batch sizes (96, 128, 160) with Recall@1 validation accuracy, and set the batch size to 160.

[1] https://cocodataset.org

[2] http://bryanplummer.com/Flickr30kEntities

		MSCOCO-1K			MSCOCO-5K			Flickr 30K		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Query-dependent	Unicoder-VL (Li *et al.*, 2020a)	69.7	93.5	97.2	46.7	76.0	85.3	71.5	90.9	94.9
	Oscar (Li *et al.*, 2020b)	75.7	95.2	98.3	54.0	80.8	88.5	-	-	-
Query-agnostic	SM-LSTM (Huang *et al.*, 2017)	40.7	75.8	87.4	-	-	-	30.2	60.4	72.3
	DAN (Nam *et al.*, 2017)	-	-	-	-	-	-	39.4	69.2	79.1
	VSE++ (Faghri *et al.*, 2017)	52.0	-	92.0	30.3	-	72.4	39.6	-	79.5
	CAMP (Wang *et al.*, 2019b)	58.5	87.9	95.0	39.0	68.9	80.2	51.5	77.1	85.3
	SCAN (Lee *et al.*, 2018)	58.8	88.4	94.8	38.6	69.3	80.4	48.6	77.7	85.2
	PFAN (Wang *et al.*, 2019a)	61.6	89.6	95.2	-	-	-	50.4	78.7	86.1
	CVSE (Wang *et al.*, 2020)	59.9	89.4	95.2	35.3	66.4	78.4	52.9	80.4	87.8
	VisualSparta (ours)	**68.7**	**91.2**	**96.2**	**45.1**	**73.0**	**82.5**	**57.1**	**82.6**	**88.2**

Table 1: Detailed comparisons of text-to-image retrieval results in MSCOCO (1K/5K) and Flickr30K datasets

4.4 Experimental Results

We compare both recall and speed performance with the current state-of-the-art retrieval model in text-to-image search. Query-dependent model refers to models in which image information cannot be encoded offline, because each image encoding is dependent on the query information. These models usually achieve promising performance in recall but suffer from prohibitively slow inference speed. Query-agnostic model refers to models in which image information can be encoded offline and is independent of query information. In section 4.4.1 and 4.4.2, we evaluate accuracy and speed performance respectively for both lines of methods.

4.4.1 Recall Performance

As shown in Table 1, the results reveal that our model is competitive compared with previous methods. Among query-agnostic methods, our model is significantly superior to the state-of-the-art results in all evaluation metrics over both MSCOCO and Flickr30K datasets and outperforms previous methods by a large margin. Specifically, in MSCOCO 1K test set, our model outperforms the previously best query-agnostic method (Wang *et al.*, 2019a) by 7.1%, 1.6%, 1.0% for Recall@1, 5, 10 respectively. In Flickr30K dataset, VisualSparta also shows strong improvement compared with the previous best method: in Recall@1,5,10, our model gets 4.2%, 2.2%, 0.4% improvement respectively.

We also observe that VisualSparta reduces the gap by a large margin between query-agnostic and query-dependent methods. In MSCOCO-1K split, the performance of VisualSparta is only 1.0%, 2.3%, 1.0% lower than Unicoder-VL method (Li *et al.*, 2020a) for Recall@1,5,10 respectively. Compared to Oscar (Li *et al.*, 2020b), the current state-of-the-art query-dependent model, our model is 7% lower than the Oscar model in MSCOCO-1K Recall@1. This shows that there is still room for improvement in terms of accuracy for query-agnostic model.

4.4.2 Speed Performance

Index Size vs. Query/s	GPU		CPU	
	Oscar	CVSE	CVSE	Visual Sparta
1K	0.4	195.1	177.4	**451.4**
5K	0.06	191.0	162.0	**390.5**
113K	0.003	101.2	5.4	**275.5**
1M	0.0003	21.7	0.3	**117.3**

Table 2: **Model Speed vs. Index Size**: VisualSparta experiments are done under setting top-n term scores to 1000. Detailed settings are reported in section 4.4.2.

To show the efficiency of VisualSparta model in both small-scale and large-scale settings, we create 113K dataset and 1M dataset in addition to the original 1K and 5K test split, as discussed in section 4.2. Speed experiments are done using these four splits as testbeds.

To make a fair comparison, we benchmark each method with its preferred hardware and software for speed acceleration. Specifically, For CVSE model (Wang *et al.*, 2020), both CPU and GPU inference time are recorded. For CPU setting, the Maximum Inner Product Search (MIPS) is performed using their original code based on Numpy (Harris *et al.*, 2020). For GPU setting, we adopt the model and use FAISS (Johnson *et al.*, 2019), an optimized MIPS library, to test the speed performance. For Oscar model (Li *et al.*, 2020b), since the query-dependent method cannot be formulated as a MIPS problem, we run the original model using GPU acceleration and record the speed. For VisualSparta, we use the top-1000 term scores settings for the experiment. Since VisualSparta can be fit into an inverted-index architecture, GPU ac-

			MSCOCO-1k			MSCOCO-5k		
n	Inf. time (ms)↓	query/s↑	R@1↑	R@5↑	R@10↑	R@1↑	R@5↑	R@10↑
50	1.9	537.0	54.6	82.8	90.7	33.0	60.0	71.1
100	1.9	514.7	60.1	86.2	92.8	37.1	64.6	75.3
500	2.1	477.7	65.5	90.3	95.1	42.5	70.6	80.4
1000	2.4	414.5	67.5	90.9	95.8	43.7	71.7	81.5
2000	3.9	256.3	68.5	91.1	96.0	44.4	72.5	82.1
all	6.9	144.1	68.7	91.2	96.2	45.1	73.0	82.5

Table 3: Effect of top-n term scores in terms of speed and accuracy tested in MSCOCO dataset; ↑ means higher the better, and ↓ means lower the better.

celeration is not required. For all experiments, we use 5000 queries from MSCOCO-1K split as query input to test the speed performance.

As we can see from Table 2, in all four data splits (1K, 5K, 113K, 1M), VisualSparta significantly outperforms both the best query-agnostic model (CVSE (Wang *et al.*, 2020)) and the best query-dependent model (Oscar (Li *et al.*, 2020b)). Under CPU comparison, the speed of VisualSparta is 2.5, 2.4, 51, and 391 times faster than that of the CVSE model in 1K, 5K, 113K, and 1M splits respectively.

This speed advantage also holds even if previous models are accelerated with GPU acceleration. To apply the latest MIPS progress to the comparison, we adopt the CVSE model to use FAISS (Johnson *et al.*, 2019) for better speed acceleration. Results in the table reveal that the speed of VisualSparta can also beat that of CVSE by 2.5X in the 1K setting, and this speed advantage increases to 5.4X when the index size increases to 1M.

Our model holds an absolute advantage when comparing speed to query-dependent models such as Oscar (Li *et al.*, 2020b). Since the image encoding is dependent on the query information, no offline indexing can be done for the query-dependent model. As shown in Table 2, even with GPU acceleration, Oscar model is prohibitively slow: In the 1K setting, Oscar is ∼1128 times slower than VisualSparta. The number increases to 391,000 when index size increases to 1M.

5 Model Analysis

5.1 Speed-Accuracy Flexibility

As described in section 3.3, each image can be well represented by a list of weighted tokens independently. This feature makes VisualSparta flexible during indexing time: users can choose to index using top-n term scores based on their memory constraint or speed requirement.

Table 3 compares recall and speed in both MSCOCO 1K and 5K split under different choices of n. From the comparison between using all term scores and using top-2000 term scores, we found that VisualSparta can get ∼1.8X speedup with almost no performance drop. if higher speed is needed, n can always be set to a lower number with a sacrifice of accuracy, as shown in Table 3.

Figure 1 visualizes the trade-off between model accuracy and inference speed. The x-axis represents the average inference time of a single query in millisecond, and the y-axis denotes the Recall@1 on MSCOCO 1K test set. For VisualSparta, each dot represents the model performance under certain top-n term score settings. For other methods, each dot represents their speed and accuracy performance. The curve reveals that with larger n, the recall becomes higher and the speed gets slower. From the comparison between VisualSparta and other methods, we observe that by setting top-n term scores to 500, VisualSparta can already beat the accuracy performance of both PFAN (Wang *et al.*, 2019a) and CVSE (Wang *et al.*, 2020) with ∼2.8X speedup.

5.2 Ablation Study on Image Encoder

As shown in Figure 2, the image encoder takes a concatenation of object label features with attributes and deep visual features as input. In this section, we do an ablation study and analyze the contributions of each part of the image features to the final score.

In Table 4, different components are removed from the image encoder for performance comparison. From the table, we observe that removing either attributes features (row 1) or label features with attributes (row 2) only hurts the performance by a small margin. However, when dropping visual features and only using label with attributes features for image representation (row 3), it appears that the model performance drops by a large margin, where the Recall@1 score drops from 68.7% to 49.1%(-19.6%).

From this ablation study, we can conclude that

#		MSCOCO-1k R@1	MSCOCO-1k R@5	MSCOCO-1k R@10	MSCOCO-5k R@1	MSCOCO-5k R@5	MSCOCO-5k R@10
1	VisualSparta	**68.7**	91.2	96.2	**45.1**	**73.0**	**82.5**
2	– attributes features	68.2(-0.5)	**91.8**(+0.6)	**96.3**(+0.1)	44.4(-0.7)	72.8(-0.2)	82.4(-0.1)
3	– labels w. attributes features	66.7(-2.0)	91.2(+0.0)	95.9(-0.3)	43.4(-1.7)	71.6(-1.4)	81.6(-0.9)
4	– visual features	49.1(-19.6)	80.3(-10.9)	89.4(-6.8)	26.5(-18.6)	54.1(-18.9)	66.8(-15.7)

Table 4: Ablation study with using different features in the image answer encoding

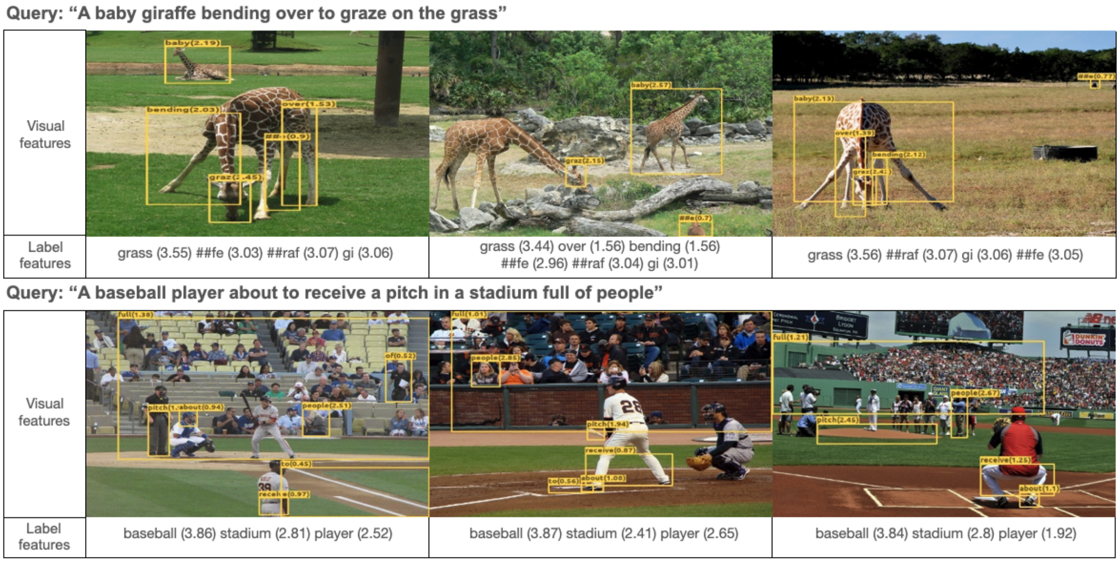

Query: "A baby giraffe bending over to graze on the grass"
Visual features
baby(2.19)
bending(2.03)
over(1.53)
##fe(0.9)
graz(2.45)
baby(2.57)
graz(2.15)
##fe(0.7)
##fe(0.77)
baby(2.13)
over(1.39)
bending(2.12)
graz(2.4)
Label features
grass (3.55) ##fe (3.03) ##raf (3.07) gi (3.06)
grass (3.44) over (1.56) bending (1.56) ##fe (2.96) ##raf (3.04) gi (3.01)
grass (3.56) ##raf (3.07) gi (3.06) ##fe (3.05)
Query: "A baseball player about to receive a pitch in a stadium full of people"
Visual features
full(1.38)
of(0.52)
pitch(1.
about(0.94)
people(2.51)
to(0.45)
receive(0.97)
full(1.01)
people(2.85)
pitch(1.94)
receive(0.87)
to(0.56)
about(1.08)
full(1.21)
people(2.67)
pitch(2.45)
receive(1.25)
about(1.1)
Label features
baseball (3.86) stadium (2.81) player (2.52)
baseball (3.87) stadium (2.41) player (2.65)
baseball (3.84) stadium (2.8) player (1.92)

Figure 3: Example retrieved images with features attended given query terms; term scores are in parentheses.

deep visual features make the most contribution to the VisualSparta model structure, which shows that deep visual features are significantly more expressive compared to textual features, i.e., label with attributes features. More importantly, it shows that VisualSparta is capable of learning cross-modal knowledge, and the biggest gain indeed comes from learning to match query term embeddings with deep visual representations.

5.3 Cross-domain Generalization

Models	R@1	R@5	R@10
VSE++(Faghri *et al.*, 2017)	28.4	55.4	66.6
LVSE(Engilberge *et al.*, 2018)	34.9	62.4	73.5
SCAN(Lee *et al.*, 2018)	38.4	65.0	74.4
CVSE(Wang *et al.*, 2020)	38.9	67.3	76.1
VisualSparta (ours)	**45.4**	**71.0**	**79.2**

Table 5: Cross-dataset performance; models are trained on MSCOCO dataset and tested on Flickr30K dataset.

Table 5 shows the cross-domain performance for different models. All models are trained on MSCOCO and tested on Flickr30K. We can see from the table that VisualSparta consistently outperforms other models in this setting. This indicates that the performance of VisualSparta is consistent across different data distributions, and the performance gain compared to other models is also consistent when testing in this cross-dataset settings.

5.4 Qualitative Examples

We query VisualSparta on the MSOCO 113K split and check the results. As shown in Figure 3, visual and label features together represent the max attended features for given query tokens. Interestingly, we observe that VisualSparta model is capable of grounding adjectives and verbs to the relevant image regions. For example, "graz" grounds to the head of giraffe in the first example. This further confirms the hypothesis that weighted bag-of-words is a valid and rich representation for images.

6 Conclusion

In conclusion, this paper presents VisualSparta, an accurate and efficient text-to-image retrieval model that shows the state-of-the-art scalable performance in both MSCOCO and Flickr30K. Its main novelty lies in the combination of powerful pre-trained image encoder with fragment-level scoring. Detailed analysis also demonstrates that our approach has substantial scalability advantages compared to previous best methods when indexing large image datasets for real-time searching, making it suitable for real-world deployment.

References

Zhuyun Dai and Jamie Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*, 2019.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

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

Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3984–3993, 2018.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

Radu Gheorghe, Matthew Lee Hinman, and Roy Russo. *Elasticsearch in action*. Manning, 2015.

Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. Convert: Efficient and accurate conversational representations from transformers. *arXiv preprint arXiv:1911.03688*, 2019.

Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2310–2318, 2017.

Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2018.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pages 11336–11344, 2020.

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

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

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Expansion via prediction of importance with contextualization. *arXiv preprint arXiv:2004.14245*, 2020.

Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge university press, 2008.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 299–307, 2017.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pretraining with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. Position focused attention network for image-text matching. *arXiv preprint arXiv:1907.09748*, 2019.

Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. Camp: Cross-modal adaptive message passing for text-image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5764–5773, 2019.

Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020.

Jonatas Wehrmann, Douglas M Souza, Mauricio A Lopes, and Rodrigo C Barros. Language-agnostic visual-semantic embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5804–5813, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. Sparta: Efficient open-domain question answering via sparse transformer matching retrieval. *arXiv preprint arXiv:2009.13013*, 2020.