Advances in Artificial Intelligence and Machine Learning; Research 3 (3) 1369–1388 Received 05-06-2023; Accepted 23-08-2023; Published 31-08-2023

Faster, Stronger, and More Interpretable: Massive Transformer Architectures for Vision-Language Tasks

Tong Chen[†] tonchn@google.com
Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA, USA 94043

Sicong Liu[†] sicongli@alumni.cmu.edu
Amazon Inc, 410 Terry Ave N, Seattle 98109, WA, USA

Zhiran Chen joann.zhiran.chen@gmail.com
Otter.ai, Inc., 800 W El Camino Real, Suite 170, Mountain View, CA 94040

Wenyan Hu huwy.vio@gmail.com
Meta Platforms, 1 Hacker Way Melon Park, CA, USA 94025

Dachi Chen chendachimail@gmail.com
Meta Platforms, 1 Hacker Way Melon Park, CA, USA 94025

Yuanxin Wang yuanxinw@andrew.cmu.edu
Carnegie Mellon University Pennsylvania, USA.

Qi Lyu lyuqi1@msu.edu
Michigan State University, 426 Auditorium Road, East Lansing, MI 48824

Cindy X. Le xl2738@columbia.edu
Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA, USA 94043

Wenping Wang wenpingw@alumni.cmu.edu
Carnegie Mellon University

Corresponding Author: Wenping Wang

Abstract

[†]These authors contributed equally to this work.
[‡]Current address: Otter.ai, Inc., 800 W El Camino Real, Suite 170, Mountain View, CA 94040.

Citation: Tong Chen, et al. Faster, Stronger, and More Interpretable: Massive Transformer Architectures for Vision-Language Tasks. Advances in Artificial Intelligence and Machine Learning. 2023;3(3):81.

Multi-layered transformer architectures have lately dominated the domain of vision-language tasks. However, massive transformer architectures can often be inaccessible to many researchers due to their sheer model sizes, and they are often treated as black boxes with poor interpretability. In this paper, we examine the weaknesses of such architectures and propose our own solutions. In particular, we select one of the state-of-the-art models called Oscar and apply distilling techniques and attention visualization to address the aforementioned issues. Moreover, we attempt to improve the overall effectiveness of the Oscar model by making its inferred object tags more useful. We show with detailed experimentation that we can both improve the performance of vision-language tasks and make them more transparent and accessible to all researchers. We discuss the findings with detailed analysis, including the effects of tags and confidence, the training behavior of distillation, and point out future directions in the end.

Keywords: Vision-language tasks, Multimodal machine learning, Multi-layered transformer.

1. INTRODUCTION

Vision-Language (V+L) tasks have long been a category of tasks that can easily apply to serve for social good and has gathered research attention; exemplary tasks include visual question answering, image-text retrieval, image captioning, etc. Learning a cross-modal representation between images and words is crucial to all these tasks, since classic encoder-decoder end-to-end methods have all been relying on such a basis. Typically, to learn such a set of embedding for both the text and image domains, proper pre-training is conducted, and then the model is fine-tuned to a specific task defined by the input and expected output of the given dataset.

Recent studies [1, 2], on vision-language pre-training have significantly improved the performances of most Vision-Language tasks through learning from massive sets of image-text pairs, and many of them [2–7], are based on multi-layer transformers. It has been shown that Visual-Language tasks can benefit from sharing embedding spaces when aligning inter-modal correspondences between images and texts. Nevertheless, pre-trained V+L models are usually memory intensive & computationally expensive given the nature of transformers recording multiple sets of weight and attention parameters, which makes back-propagation a memory-demanding job. Even the model-to-go for language generation task, BERT [8], has several hundred million trainable parameters and takes days to converge using an enormous cluster of top-tier graphics cards that most individuals and small companies cannot afford, let alone environmental concerns due to exponentially increasing use of power. Also, it has been not easy enough for humans to understand the way a particular model uses to extract, focus and make decisions in a multi-modal setting; it is hard to gauge what neural networks learn about vision and language space.

In this paper, based on the strong generalization power of the transformers and the discoveries we have made in one of the state-of-the-art models, which uses extrinsic tag information as a supplement, we want to make the model better utilizes information from the image side, especially through more proper use of related labeled information. On this path we test two ideas, one being reordering, removing and reorganizing input tags, and including confidence scores; the other being

the copy mechanism. This helps us understand how such models work and make decisions in the multimodal domain, and also find ways to make the model lighter.

Under these considerations, our main contributions to the work include the following points:

- Make the SoTA Vision-Language model **faster** by reducing model size via knowledge distillation while maintaining accuracy performance.
- Make the performance of the SoTA Vision-Language model **stronger** by re-engineering the input features using two methods: object tag confidence incorporation and object tag copy mechanism.
- Make the SoTA Vision-Language model **more interpretable** via visualizing its attention mechanisms through different design decisions on extrinsic information.

2. RELATED WORK

Transformers. Transformers, as proposed in [9], is a relatively new network structure that has the capacity to replace the long-standing rockstars of Convolutional and Recurrent neural networks. It can capture long-term dependencies between input tokens, through a self-attention mechanism.

The calculation of the attention function depends on three parts of input: Queries, Keys, and Values. Query and Key values conduct matrix multiplication together, the result goes through a softmax layer, and the output is multiplied to the value matrix.

Multi-head attention is defined by concatenating the attention heads from different representation subspaces:

$$\texttt{MultiHead}(Q, K, V) = \texttt{Concat}(head_1, ..., head_h)W,$$

where $head_i$ denotes the i-th attention head as in all h heads, calculated from each of the corresponding attention functions; the matrix W acts as a linear transformation, thus being called a "Transformer".

Pre-training transformer-based language models, then fine-tuning for downstream tasks has become a new norm for Natural Language Processing, considering this kind of model's powerful performance and its enormous cost of training from scratch. Pre-trained language models such as BERT [8], XLNet [10], RoBERTa [11], etc., have proven themselves to be successful in the language-only domain, but the many layers of transformers and up to hundreds of millions of parameters are a burden even for inference and fine-tuning.

Vision-Language Pre-Training. Guided by the trend from the release of BERT [8], there is a growing research trend in building pre-training large, generic models to solve a variety of Vision-Language problems, such as Visual Question Answering (VQA), image-text retrieval, image captioning, etc. Many existing methods, for example, VLP [12], bottom-up attention [13], LXMERT [5] and VL-BERT [3], employ BERT-like objectives to learn cross-modal embeddings/representations from visual region features and language token embeddings concatenated together (e.g. VL-BERT), or cross-referred in parallel (e.g. LXMERT). These models heavily rely on the self-attention mecha-

nism of Transformers [9], across and within modalities; the goal and effect is to learn joint representations that are appropriately aligned between the modalities. Typically researchers would use Faster RCNN to train and extract objects in the picture to be served, and when these items are recognized, a tag will also be guessed about the object. Compared to the prior, we make model changes that better utilize this precious enriched data source, and use it as a guide to make transformers focus more on specific on the saliently recognized objects with high confidence.

Knowledge Distillation. Knowledge distillation [14], is a model compression technique in which a more compact model - the student - is trained to reproduce the behavior of a larger model - the teacher - or an ensemble of models, so that comparable performance is retained while parameter size is greatly reduced. In practice, the teacher feeds the student with distribution losses at all levels.

A series of Knowledge Distillation papers around BERT, namely DistilBERT [15], TinyBERT [16], and Simplified TinyBERT [17], follow the same path of shrinking the original BERT [8], model through using smaller parameters, fewer layers and dropping sub-layers. These models use the soft output from Embedding, Attention, Hidden, and finally, Fully-connected Prediction layers to generate soft losses for the student model. In some cases, the hard label is also used as another loss term.

There are also knowledge distilled models that move beyond retaining the original structure. In [18], knowledge distillation is done through contrastive loss, in which the marginal representation between the teacher and student models is maximized in mutual information when a joint representation is provided. In [19], many other cross-modality methods are introduced, which include ways of learning from many teachers as in [20]. Compared to prior work, our work is among the first trials to introduce knowledge distilling in any form for a multimodal transformer model, built upon training concatenated field of representation for images and language.

3. PROPOSED APPROACH AND DESIGN

3.1 Problem Statement

Although our main focus of the problem is image captioning, the more essential task we are dealing with is training a multimodal model across image and language. We define the multimodal dataset as $[\mathbf{w}, \mathbf{v}]$, in which $\mathbf{w}$ denotes the caption tokens, and $\mathbf{v}$ denotes the extracted features of the image. The two modalities of data pair together, often in a many-to-one form, since very often we have many descriptions provided for a given image in the dataset.

To pretrain a multimodal model for the united language-image domain, a typical approach is to randomly mask part of the input tokens, and maximize the possibility that the model guesses out the right missing words, or $f([\mathbf{w'}, \mathbf{v}]) \rightarrow [\mathbf{w}]$. On the other hand, the task of image captioning would require a separate objective; training uses the same masking method, while during inference, only the image and a starting token are supplied; the model needs to generate text output corresponding to the input image, or $f([[CLS], \mathbf{v}]) \rightarrow [\mathbf{w}]$. Typically language generation is done through beam search to maximize the joint probability of the tokens.

3.2 Baseline

Compared to other aforementioned Visual-Language Pre-Training methods, the most outstanding difference of the model as in Oscar [4] is the use of auto-recognized object tags that connect elements in the two modalities. The pre-training objective for Oscar is defined into two parts according to the focus of loss with respect to the tags: *Masked Token Loss* in the Dictionary view, in which some input tokens are masked and the model is asked to train to predict them using the remaining input, minimizing the negative log-likelihood on word tokens; *Contrastive Loss* in the Modality view, in which tags are randomly replaced as a whole and the model is asked to predict whether the input contains polluted tags or not, minimizing the negative log-likelihood on the binary prediction. Application of loss functions like these are widely seen in other tasks as well [21]. A figurative view of this structure is shown in FIGURE 1.

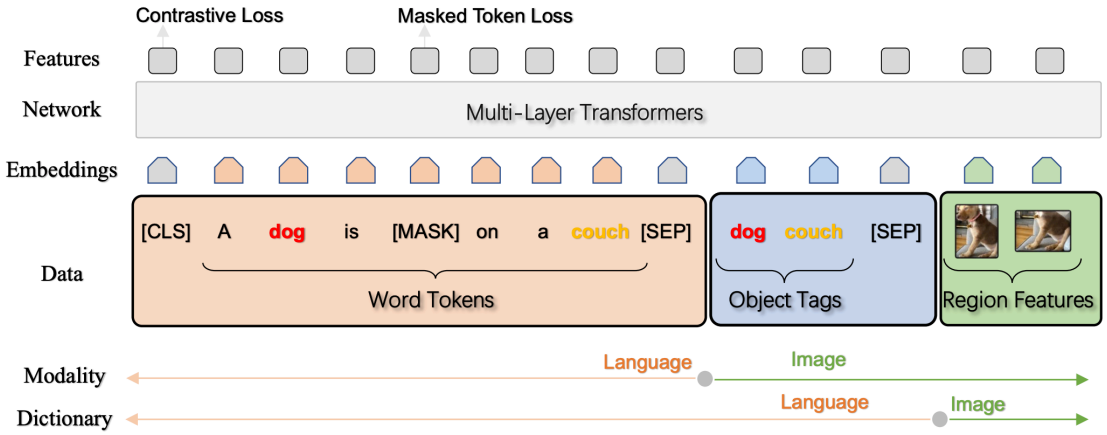

Figure 1: Structure of the Oscar model. The three input components across two modalities are concatenated and sent into the layers of transformers.

3.3 Proposed idea 1: Investigation of feature contribution and re-engineer feature

With the goal of understanding what factors contribute to cross-modal transformer's success and failure cases and re-engineering the features to make the model better accordingly, in this idea, we first designed ablation studies to study the effect of image features and object tags (along with object tags' order) on model performances on image captioning tasks. We used pre-trained the Oscar model *Bert-base* that is trained with both image region features and object tags. The image region features are extracted by the Faster R-CNN with ResNet-101, using object and attribute annotations from the Visual Genome dataset. The object tags are from the same Visual Genome model.

Then after the ablation study and error analysis, we found that object tags are crucial to the cross-modal transformer's success, however, the object tags information is not fully utilized during the fine-tuning stage because identified salient object tags with high confidence are not present in the predicted caption. In particular, in the implementation of Oscar, the confidence of each object tag is

Table 1: **Performance of the baseline shown above.** We compare the three models on the accuracy of the VQA and COCO Image Captioning task.

Model	VQA test-dev	IC CIDEr	IC SPICE
VL-BERT	70.50	120.5	21.3
LXMERT	72.42	122.3	22.4
Oscar	73.61	123.7	23.1

not utilized in the current Oscar model. We thought that if we were to incorporate the tag confidence into the training process, it would be effective in improving the performance. We thus propose to fine-tune the model for image captioning by incorporating confidence information for each object tag into the feature. We experimented with this approach by two different configurations: (1) reorder object tags based on descending confidence. (2) concatenating the confidence value for each object tag directly into the embedding.

Specifically, to represent objects in images, we encode not only its object tags, and image region features (Oscar's approach), but also added its predicted confidence value from Fast R-CNN results to the input embedding. The input samples are processed to quadruplets (w, o, c, i) consisting of captions w, object tag tokens o, and object tags confidence values c, and image region features i in the same way and dimensions as that during the Oscar pre-training [4].

Each of these experiments (FIGURE 2) took approximately 50 hours on 4 NVIDIA Tesla V100 GPUs.

3.4 Proposed Idea 2: Adding Copy Mechanisms Over Object Tags

Image captions tend to describe the most salient objects in the images and these salient objects are also the most important components in our captions. Oscar extracted object tags from Faster RCNN and used them to help align the visual modality and text modality. According to our dataset analysis, we found that more than 20% tokens are within these object tags and the tokens used in tags tend to be consistent with the ones used in the ground truth. We found that when Oscar tried to generate a token, it only attended previously generated tokens and visual features while ignoring the object tags. This leads to two error examples. First, when an object can be described with multiple names, Oscar may try to select tokens from the vocabulary, but the tokens used in the ground truth tend to be included in our object tags. Second, Oscar may miss some salient objects in its captions. Therefore, in order to improve the caption quality, it can be intuitive to directly copy tokens from object tags instead of predicting tokens from the large fixed vocabulary.

Specifically, the approach for idea 2 is to apply a weighted probability average over object tags. At each time step $\mathbf{t}$, $P_t = (1 - \alpha) * P(vocab) + \alpha * P(tag)$ where α is the probability of whether the generated token at the current step $\mathbf{t}$ is from object tags. The simplest way to estimate α is using the ratio of tags in our ground truth in the training set which is about 20%. But this naive approach will make the probability of object tags much higher than others. Therefore, we need to estimate a context-dependent α. One option we used is to check the top k predictions at the current

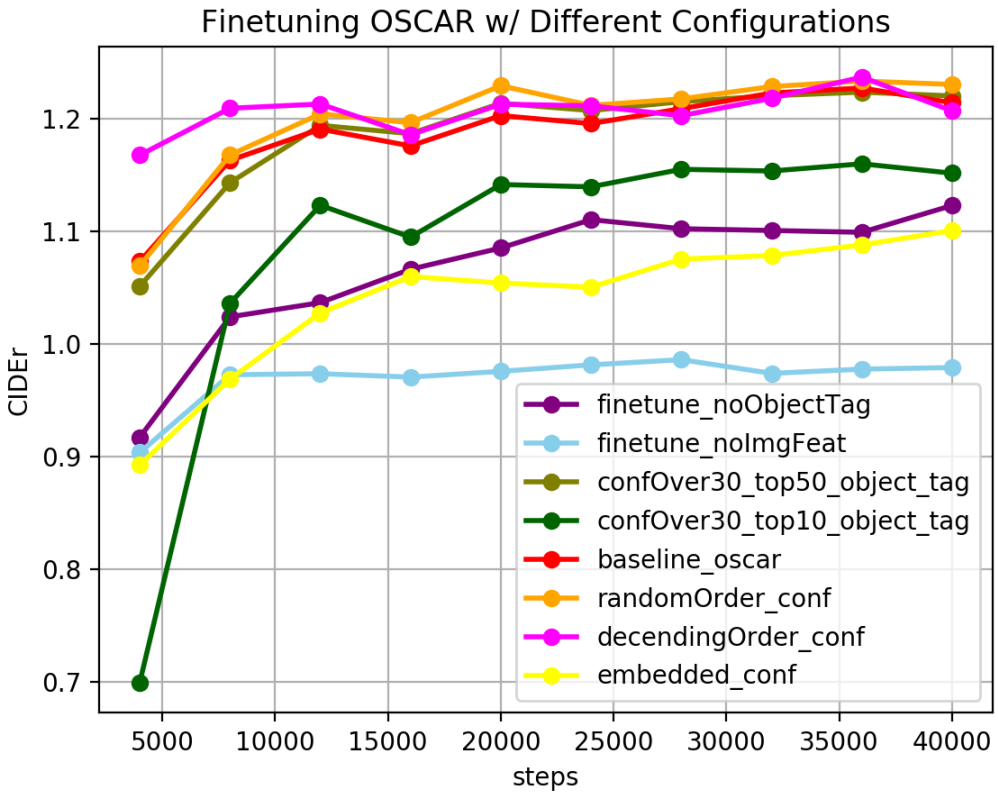

Figure 2: Fine-tuning Oscar with different configurations

step. If the top k predictions overlap with our object tags, we will add the probability distribution of object tags to the original distribution. Otherwise, we will set α to 0 since it is not reasonable to increase the probability of object tags when we want to generate some stop tokens. A more general solution for approximating α is applying a linear layer over the final representation of the model and using the normalized output of this linear layer as our α. This α is more context-dependent and can be trained with our model. The probability distribution of the object tags can be approximated by $P(vocab) = z^t_{text} * (z_{tags})^T$ where z^t_{text} is the representation of the decoder output at the current timestep t and z_{tags} is the representation of our object tags.

3.5 Proposed Idea 3: Distill the Network

We follow suit from the design of [15], which has proven itself to be a valid distillation of the original BERT model. The distilled Oscar model uses the same structure and input configurations as the original Oscar, except that the token-type embeddings and pooling layers are not kept, and only six layers are kept compared to the original design of 12 layers.

To properly initialize the student model, we use the pre-trained Oscar model checkpoint and selected the $1, 3, 5, 8, 10, 12$th layers' weights as the initial weights of the six transformer layers in the distilled student. Our consideration is that using a proper and similar initialization weight should greatly accelerate the progress of fitting. Considering the fact that we do not have enough computation power to fit from scratch, we fit the distilled model directly on the image captioning task, using the trained image captioning checkpoint as the teacher, and initialize weights from the same checkpoint.

The image captioning task uses masked word prediction as the training measure. We use four losses to force the student to produce similar results to the teacher:

- Cross Entropy Loss of the student's generated output compared to ground truth target. This aims to train the model to properly interact with designated input, as being actually used in Oscar's fine-tuning on image captioning task;
  $L_{Orig} = \sum_i \hat{t}_i * log(s_i)$
- Mean Squared Error Loss between the student's full logits and the teacher's generated logits;
  $L_{MSE} = \sum_{i,j}(t_{i,j} - s_{i,j})^2$
- Cosine Embedding Loss for the last hidden states output compared to the teacher's hidden states;
  $L_{COS} = \sum_i 1 - cos(hidden_{t,-1,i}, hidden_{s,-1,i})$
- Distillation Loss between the student's and the teacher's soft target loss.
  $L_{CE} = \sum_i t_i * log(s_i)$

The resulting loss is a linear combination of the four, and for the latter three distillation losses an alpha $\alpha_{mse}, \alpha_{ce}, \alpha_{cos}$ is assigned to each to adjust its ratio in the total loss.

4. EXPERIMENT SETUP

Dataset. The dataset we are going to use is the MS-COCO dataset [22]. MS-COCO, or the Microsoft Common Objects in COntext dataset, is a large-scale object detection, segmentation, and captioning dataset. Depending on the tasks it's aiming for, MS-COCO has six variants, namely Detection and Stuff (for object detection), DensePose and KeyPoints (for human pose detection), Panoptic (for scene segmentation), and last, Captioning, which is the only multi-modal task in all, which cross the modalities of language and vision.

The training part of the MS-COCO Captioning dataset has 112K images and 560K text descriptions. All these sets, including other datasets of the same structure, are used in the pre-training of the Oscar paper [4], so we also want to make sure whether a limited training set in pre-training would hamper model performance to what extent. In the meantime, in the subtasks we would like to explore, the datasets are used separately to make the pre-trained model familiar with the problem and answer settings.

Following [4], we adopt the widely used Karpathy split on the COCO caption dataset to conduct our experiments. Specifically, the dataset consists of $113,287$ images for training, $5,000$ images

for validation, and $5,000$ images for testing. Each image is associated with 5 human-generated captions.

Baseline models. We have seen a variety of language-vision pretrained transformer models, however, Oscar [4], includes the use of auto-recognized object tags that connect elements in the two modalities, and supply extrinsic information. This has inspired us on potential room for improvement, while we find it only to a limited degree, and the main challenge is how to organize the tags for maximum effect. Other similar models we have chosen as a comparison are VL-BERT [3], and LXMERT [5], both utilizing BERT-like layers to take both sides of information.

Methodology. Image Captioning is essentially a language generation task, and the generated text is compared against ground truth tokens for precision and recall. The metrics include BLEU@4 [23], CIDEr [24], METEOR [25], and SPICE [26]. These are all automatic metrics that try to mimic human judgments towards generated text; BLEU and METEOR are n-gram comparison metrics to gold standards that are more text-derived, whereas CIDEr compares the text with several reference texts using `tf-idf` and SPICE matches synonyms in the auto-detected scene-graphs [27]. In the following sections, we mainly report the CIDEr score, since it focuses more on the appearance of certain critical words, compared to other metrics that focus more on coherence.

In training, we follow the training parameters outlined in the original Oscar model: We use `adamW` as our optimizer with an initial learning rate of 0.00003, weight decay of 0.05, and an epsilon value of `1e-8`. We trained on 4 NVIDIA Tesla V100 GPUs with a batch size of 64 for 40k steps for most of our experiments.

5. RESULTS AND DISCUSSION

5.1 General Performance

Since VL-BERT and LXMERT are both pre-trained language models but are mainly tested on the VQA task instead of image captioning, to properly compare the models' performance we decide to compare both the VQA scores (accuracy% of the right answers) and also the score for the two models when they are adapted and fine-tuned to accept image captioning tasks. Performance metrics can be seen in Table 1.

Since all three models are transformer-inspired, it is expected that they have similar results, while Oscar outperforms through adding additional image-tag information.

5.2 Idea 1: Investigation of Feature Contribution and Re-Engineer Feature

Ablation: Image region features contribution To study the contribution of image region features to the model's performance, we removed image features at the fine-tuning stage and compared their performance with the Oscar baseline model. We found this results in significant deterioration in image-captioning quality as shown in the blue line *finetune_noImgFeat* in FIGURE 2, which signifies image region features are indeed crucial to the model's success.

Ablation: Object tags contribution To study the contribution of object tags to the model's performance, we removed object tags at the fine-tuning stage and compared their performance with the baseline model. We found this results in consistent deterioration in image-captioning quality at all fine-tuning steps, which signifies object tags are useful to the model's success. As should be in the purple line *finetune_noObjectTag* in FIGURE 2, in this case, the performance deterioration is not as significant as that of removing the image region features, which is expected.

Shuffling object tags We are also interested in, as an input to the transformer models, whether switching the order of object tags would influence its performance. We experimented with this idea by switching the order of the object tags randomly at inference time, and see how much does performance change. As the orange line *randomOrder_conf* in FIGURE 2 demonstrates, randomly order object tags actually achieves similar performance as the baseline model, meaning the model does not derive significant information from object tags' order at inference time.

Removing object tags by confidence level Then, since object tags generated by Faster R-CNN include confidence information. We are interested in if the model's performance could be influenced by the confidence of object tags that are passed in. We experimented with this idea by removing some of the tags at inference time, to keep only the object tags with more than 30% confidence, then ordering these object tags by confidence and taking the top k% (experimented with k=50 and k=30). As the olive line *confOver30_top50_object_tag* and dark green line *confOver30_top10_object_tag* in FIGURE 2 demonstrates when we keep the top 50% object tags that have at least 30% confidence, it achieves the similar performance as the baseline where all object tags are passed in, whereas, if we only keep the top 10%, the performance is much worse. This suggests that object tags with lower confidence passed in at inference time do not contribute much to improving performance.

Confidence as a feature In our experiments, we found that a significant amount of salient objects identified by the object tags are not present in the predicted caption. This tells us that Oscar fails to fully utilize the information provided by the object tags. We thought that if we were to incorporate the tag confidence into the training process, it would be effective in improving the performance. We thus fine-tuned the Oscar model for image captioning by incorporating confidence information for each object tag in the input. We explored this by mainly two ways: (1) to reorder object tags based on descending confidence. From the magenta line *descendingOrder_conf* in FIGURE 2, we see that this allowed the training to converge faster, but ultimately was not able to beat the Oscar baseline (2) the second way involves injecting the confidence value for each object tags directly into the word embedding during the fine-tuning stage. The yellow line *embedded_conf* in FIGURE 2 shows that this approach did not work well for us, but we suspect that this is due to insufficient training time.

Error Analysis For our *Confidence as a feature* approach, we compiled a list of generated captions that demonstrates the most significant differences in terms of CIDEr score. TABLE 3 compares the baseline model against the model we trained that uses confidence-based ordering. TABLE 4 compares the baseline model against the model we trained that injected the confidence value for each object tag. We see that the errors are mostly caused by the model focusing on an irrelevant section of the image, even if the generated caption is "technically" correct.

Table 2: **Performance of different copy mechanisms.** We compare CIDEr scores among different copy mechanisms.

Model	α	CIDEr
Oscar baseline		123.7
Weighted average	0.2	34.9
Weighted average	0.2 if top k overlap	119.6
Weighted average	linear layer	124.1

Table 3: **Error Analysis: Baseline vs. Conf-order** (Incorrect captions highlighted)

Baseline	**Confidence-ordered Model**	**Ground Truth**
a man standing on a baseball field holding a bat.	a man standing in the middle of a parking lot	a fire hydrant in the middle of the parking lot.
a group of horses that are standing in the sand.	a herd of horses standing on top of a sandy beach.	a herd of horses are roaming free on the shore of a beach.
a man holding a cell phone up to his ear.	a man talking on a cell phone in a restaurant.	a man talking on a cell phone in a building.
a truck is parked on the side of the road.	a truck parked next to a stone wall with a kite	a truck is parked in a recreation area.
a brown and white dog wearing a tie.	a dog that is wearing a tie in a room.	a brown and white dog wearing a neck tie.
a polar bear playing with a ball in the water.	a couple of bears are swimming in the water	a polar bear playing with a ball in a small pond area.

5.3 Idea 2: Copy Mechanism Over Object Tags

We tried different copy mechanisms over the object tags and the result is shown in TABLE 2. We found simply averaging the probability distribution between object tags and vocabulary was harmful to our model. In this method, the model will generate object tokens at some unexpected time steps. After the model generated some unexpected tokens and conditioned on this context, the model failed to generate the rest of the sentence. For example, it would generate some terrible repeated patterns since it hasn't seen such unexpected context in the training time and didn't know what to predict next. The perplexity of generated captions in this method is relatively low. The performance of the Top-K overlap copy mechanism is similar to the baseline model. Here we choose 10 as our K. We found that this approach mitigated the issue in the first approach. In this method, our model will skip many time steps where we need to predict some stop words. In other words, Top-K overlaps

Table 4: **Error Analysis: Baseline vs. Conf-embedded** (Incorrect captions highlighted)

Baseline	**Confidence-embedded Model**	**Ground Truth**
a close up of a person holding a hair dryer	a close up of a woman brushing her teeth	a girl with blonde curly hair brushing her teeth.
a woman walking out of the ocean with a surfboard.	a woman carrying a surfboard on a beach.	a surfer runs along the beach with a white board.
a red fire hydrant sitting on the side of a road.	a fire hydrant on the side of the road.	a fire hydrant on the side of a street.
a purple and white bus driving down a street.	a bus that is parked on the side of the road.	a purple and white bus driving down a street.
a man in a black shirt and orange bow tie.	a close up of a person wearing a suit and tie	a close up photo of a man with an orange bow tie.
a woman blow drying her hair in a bathroom.	a woman taking a selfie in a bathroom mirror.	a woman stands in a bathroom blow drying her hair

copy mechanism will attend to the object tags according to the given context. Only when we need to generate an object at the current step, our model will attend object tags. The third method is to use a linear layer over the final representation of our model to approximate the α. This can better utilize the context information and thus generate better captions. More importantly, this α is learned instead of simply estimated by our dataset.

5.4 Idea 3: Training Curves and Losses for the Distilled Model

The original training setting for the parts of losses are $\alpha_{mse} = 0.2, \alpha_{ce} = 0.5, \alpha_{cos} = 0.2$. We have trained the distilled model to a much lower loss, however, the model still cannot properly generate sentences given the tokens. After training for 10 epochs, the resulting model can still only generate consecutive random words. Thus we would want to analyze how loss has been reduced through the epochs.

In Figure 3, each step represents a 64-size batch; thus each epoch represents about 8200 steps. We have tested different batch sizes under the acceptable memory constraints of V100 graphics cards, and discovered that larger batch sizes than 64 won't save time but generalizes worse; smaller batch sizes help reduce loss quicker but takes too long to train.

As shown in Figure 3, when the alpha of MSE loss and CE loss are set into the same multitude, the loss is mainly guided by MSE loss. Other parts of the losses are in the lower hand and can not

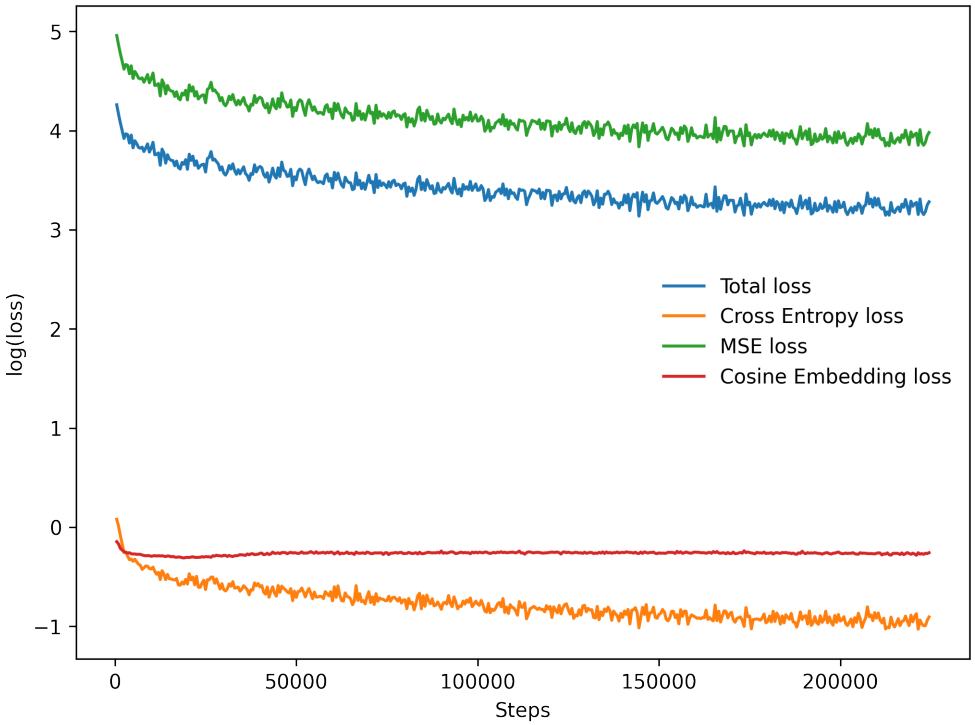

Figure 3: **Parts of losses in distillation.** Since the loss is very high on the MSE loss side, we use the logarithm scale for comparison.

contribute enough to training. Based on the current low-loss model, we suppressed the weights of MSE loss to 1e-4 and trained for another epoch; The results bounced back to regular levels. After 40k steps, we reached a **CIDEr score of 80.4**. This has proven that our student model has learned some features from its teacher model.

This has given us the following insights:

- Balancing the losses is a crucial part of combining the losses for model distillation; we have to properly suppress the losses that are too high.
- Token-type embeddings, compared to the implementation of BERT, introduce more information in Oscar, so removing this would hinder some extra performance.
- Even despite the suppression from other loss components, the other losses still get trained and can recover from the suppression.

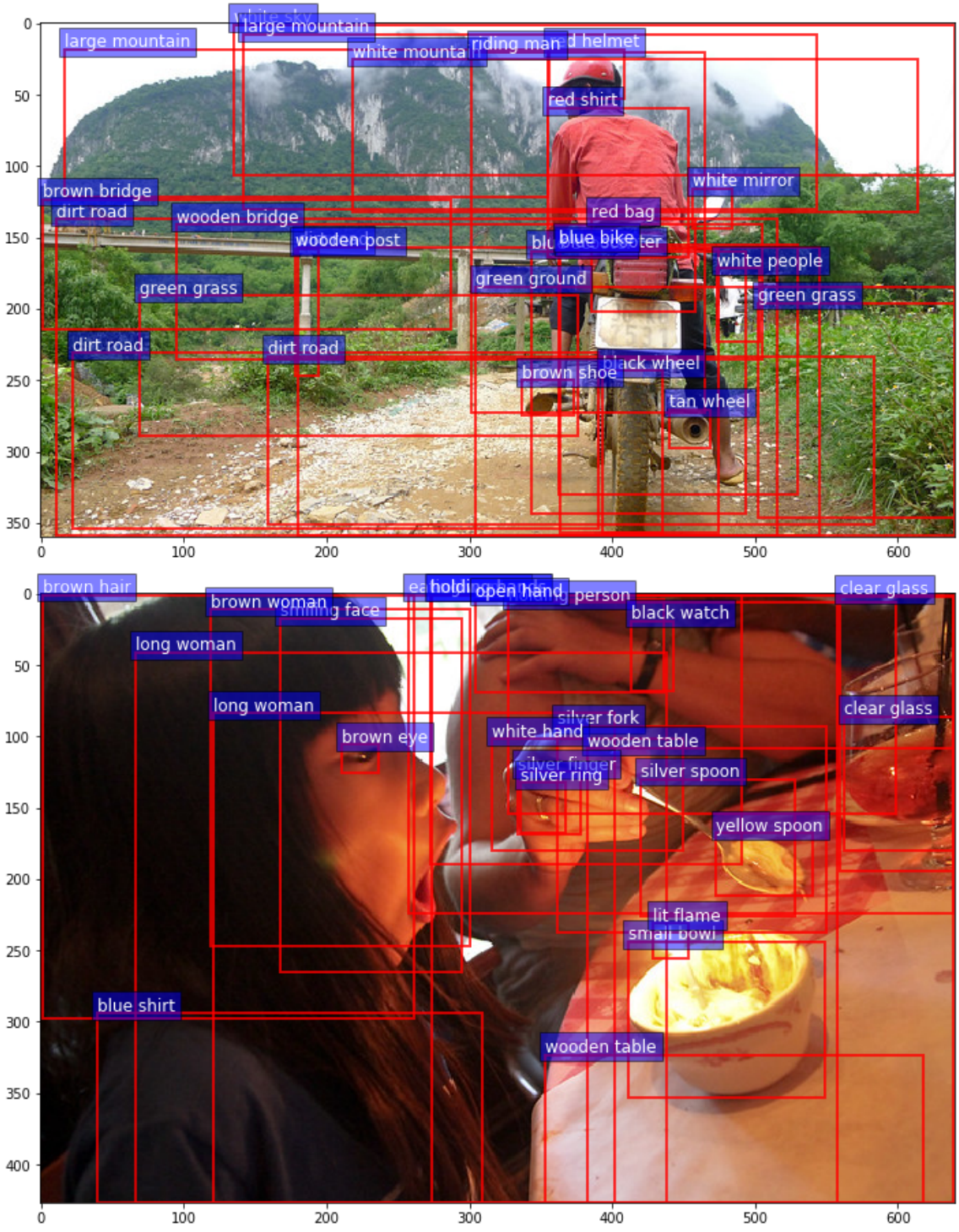

Figure 4: Image features visualization for error analysis case study

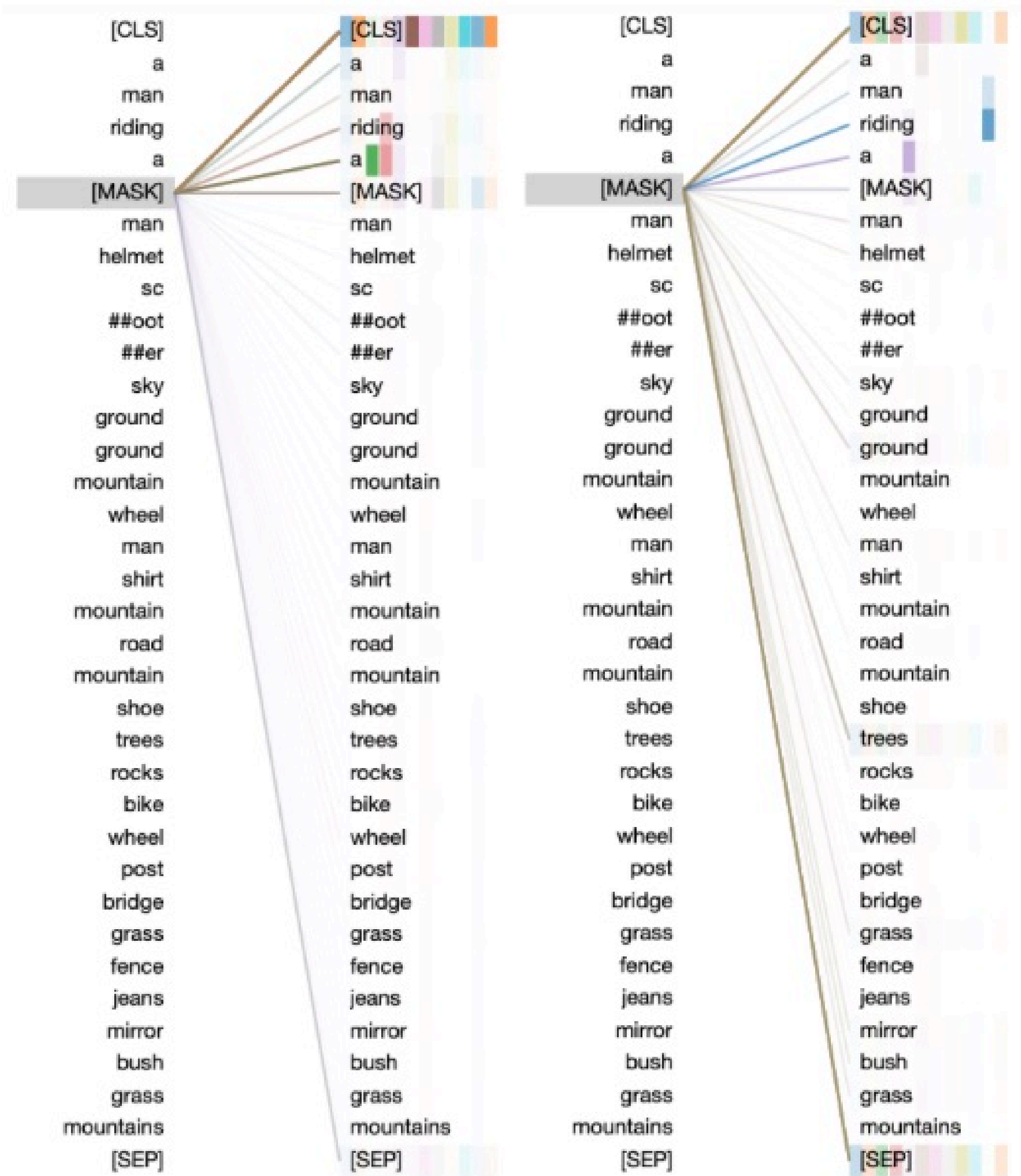

Figure 5: Attention weights when Oscar generated motorcycle for the first image

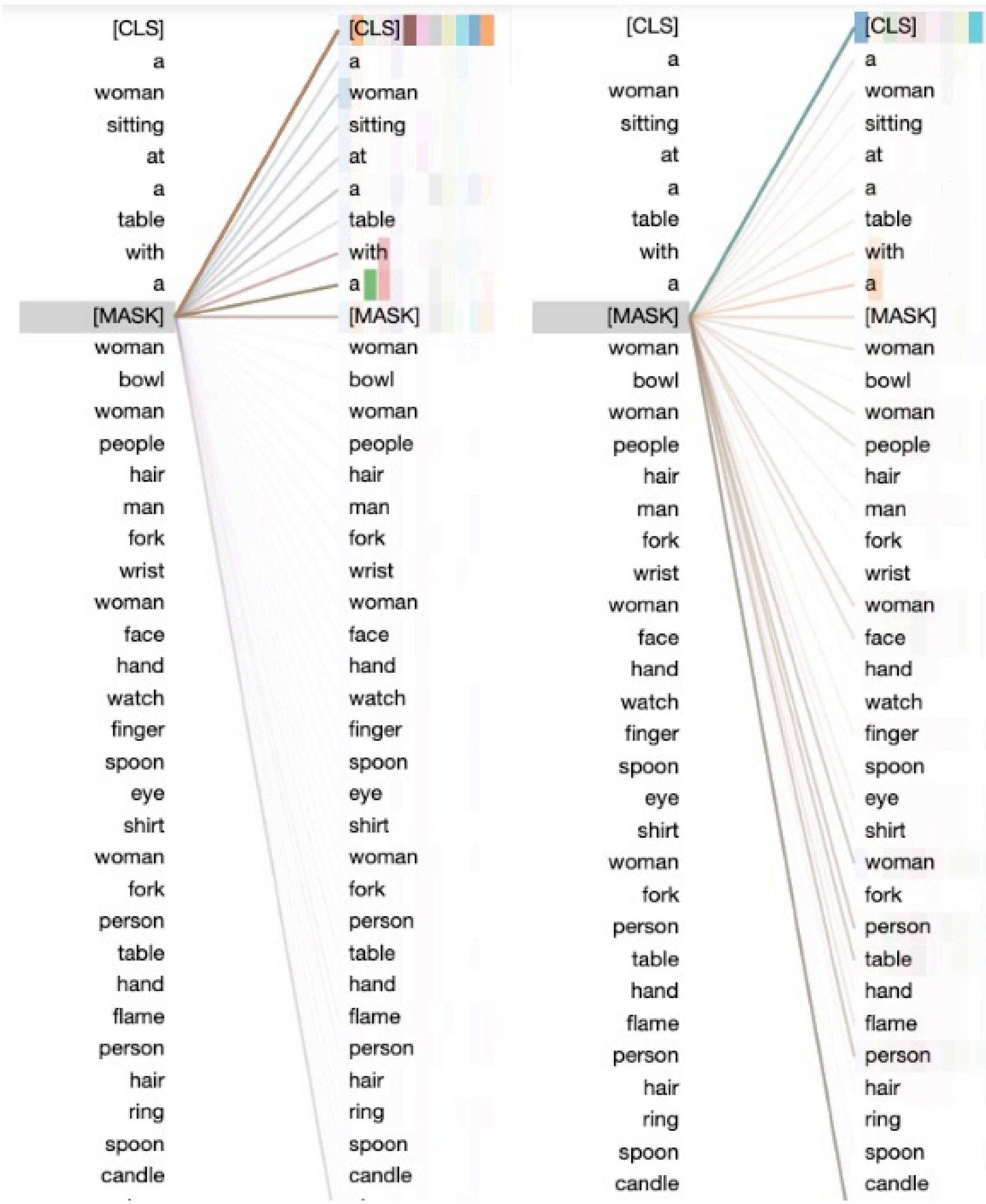

Figure 6: Attention weights when Oscar generated table for the second image

5.5 Investigation of Interpretability via Attention

Besides the ablation study, we also analyze the failure cases. Two examples are shown in Figure 4. For the first example, the Oscar model missed 'helmet' in its prediction although 'helmet' explicitly exists in the object tags with high confidence. And 'candle' is also missed in the prediction for the second example. Also, Oscar used 'motorcycle' in its prediction since the frequency of 'motorcycle' is quite high in our training set. However, 'motorcycle' is not in the object tags. These two observations in our error analysis indicate a potential problem that Oscar didn't make full use of object tags.

So we continue to analyze the attention weights during the generation process, as shown in FIGURE 5. Here Oscar tries to predict [MASK] and will generate 'motorcycle' in this position. All tokens above [MASK] are previously generated tokens and tokens below [MASK] are object tags. Ideally, we hope to observe that [MASK] attends to 'bike' in the object tags. However, in most layers, especially for the deeper layers, [MASK] only attends previously generation captions. Only in the second and the third layers, [MASK] attends our object tags, as shown in FIGURE 5 right. But these attended tags such as trees, ground, and grass seem to be irrelevant or not directly relevant to the predicted tokens. In fact, we found [MASK] has no interaction with the 'bike' tag. FIGURE 6 shows a similar result, only a few tags are attended in the limited layers. This observation means when Oscar predicts the next token, it mainly focuses on its context and selects a token from the whole vocabulary instead of from the set of object tags which can be more consistent with the ground truth.

According to our ablation study and failure case analysis, we found object tags are actually used in Oscar and can help improve the performance of image captioning tasks. However, Oscar did not explicitly make full use of object tags for the generation process. So one of the potential directions is to better utilize the object tags. Also, how to obtain better object tags can be another promising direction.

6. CONCLUSION AND FUTURE DIRECTIONS

This paper focuses on improving image captioning task on the baseline model Oscar comprehensively from three different aspects: (1) using distillation to make the model **faster** (2) re-engineering input features involving object tag confidence incorporation and adding copy mechanisms over object tags to make the model **stronger** (3) providing a standardized approach to visualize the attention weights, which is the foundation of all transformer-based models, to make the model **more interpretable**.

The contribution of object tag confidence incorporation is the finding that object tag confidence might be a good feature according to our analysis that salient objects with high confidence are sometimes not present in the predicted caption. But it is difficult to leverage the tag confidence effectively. One promising future direction of this approach is in the line of finding ways to leverage the object tag information more effectively: to directly embed coordinates of identified object tags to help better align the image features and object tags; better alignment of these two modalities could lead to promising results.

One promising future direction for the copy mechanism approach is applying a dynamic pointer network. Since the object tags are dynamic among different data points, we cannot simply add the vocabulary size. Instead, we should use a dynamic network to predict the copy score for each object tag with bilinear interaction between the decoder output and object tags' output representation which is similar to the way we estimate the tag probability distribution in the first approach. The key difference is that in our current approach for idea 2, we only calculate the tag distribution at the inference time while we could modify the loss function and train the model with a new objective in this approach for future work. Specifically, $y_{t,n}^{tag} = (W^{tag} z_n^{tag} + b^{tag})^T (W^{text} z_t^{text} + b^{text})$ where $y_{t,n}^{tag}$ is the probability of tag n at time step t. Then we concatenate y_t^{tag} with y_t^{vocab} and take the argmax from the whole token sets. Since this approach will turn the objective into a multi-label classification task, we need to tune the hyperparameters to fit this new setting. Meanwhile these mechanisms can be further implemented with hardware accelerators [28], to achieve better performance.

On the distillation side, this time we mostly borrow the methods used in the distilled version of BERT. Considering the nature of BERT-like transformer-based pretrained models that consider input from other modalities as just another type of embedding input, it might at first look not particularly "multimodal", but developing distilling methods and training goals that work distinctly for such vision-language models is a valid consideration of future work, and can also serve research in related fields that equip pretrained models.

We found the attention weights visualization method very helpful for both successful and failed case analyses. In this work, we just used the simplest raw attention weights for such visualization. In the future, more advanced approaches such as [29], and [30], can be incorporated. We hope our contribution can be helpful to other tasks and important applications in general like [31–33], etc.

References

[1] Xu K, Ba J, Kiros R, Cho K, Courville A,et al. Attend and Tell: Neural Image Caption Generation With Visual Attention. In: Proceedings of the international conference on machine learning; 2015:2048-2057.

[2] Wang Y, Joty S, Lyu MR, King I, Xiong C, et al. Vd-Bert: A Unified Vision and Dialog Transformer With Bert. 2020. Arxiv preprint: https://arxiv.org/pdf/2004.13278.pdf

[3] Su W, Zhu X, Cao Y, Li B, Lu L,et al. Vl-Bert: Pretraining of Generic Visual-Linguistic Representations. 2019. ArXiv preprint: https://arxiv.org/pdf/1908.08530.pdf

[4] Li X, Yin X, Li C, Zhang P, Hu X, et al. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In: Proceedings of the European conference on computer vision. Springer. 2020:121-137.

[5] Tan H, Bansal M. Lxmert: Learning Cross-Modality Encoder Representations From Transformers. 2019. Arxiv Preprint: https://arxiv.org/abs/1908.07490

[6] Chen T, Wang X, Yue T, Bai X, Le CX, Wang W. Enhancing Abstractive Summarization with Extracted Knowledge Graphs and Multi-Source Transformers. Appl Sci. 2023;13:7753.

[7] Yang X. Linguistically-Inspired Neural Coreference Resolution. Adv Artif Intell Mach Learn. 2023;3:1122-1134.

[8] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pretraining of Deep Bidirectional Transformers for Language Understanding. 2018.Arxiv preprint: https://arxiv.org/abs/1810.04805

[9] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN et al. Attention Is All You Need. In proc Adv Neural Inf Process Syst. 2017:5998-6008.

[10] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV. Xlnet: Generalized Autoregressive Pretraining for Language Understanding. Adv Neural Inf Process Syst. 2019;32:5753-5763.

[11] Liu Y, Ott M, Goyal N, Du J, Joshi M, et al. A Robustly Optimized BERT Pretraining Approach. 2019. Arxiv Preprint: https://arxiv.org/abs/1907.11692

[12] Luowei Z, Palangi H, Zhang L ,H Hu, Corso J, et al. Unified Vision-Language Pre-training for Image Captioning and Vqa. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence. 2020; 34:13041–13049.

[13] Anderson P, He X, Buehler C, Teney D, Johnson M,et al. Bottom-up and Top-Down Attention for Image Captioning and Visual Question Answering. In: Proceedings of the CVPR. 2018:6077-6086.

[14] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network. 2015. Arxiv Preprint: arxiv.org/pdf/1503.02531.pdf

[15] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2019. ArXiv preprint: https://arxiv.org/pdf/1910.01108.pdf

[16] Jiao X, Yin Y, Shang L, Jiang X, Chen X, et al. Tinybert: Distilling Bert for Natural Language Understanding. 2019. Arxiv Preprint: https://arxiv.org/abs/1909.10351

[17] Chen X, He B, Hui K, Sun L, Sun Y. Simplified Tinybert: Knowledge Distillation for Document Retrieval. 2020. Arxiv preprint: https://arxiv.org/abs/2009.07531

[18] Tian Y, Krishnan D, Isola P. Contrastive Representation Distillation. 2019. Arxiv Preprint: https://arxiv.org/pdf/1910.10699.pdf

[19] Wang L, Yoon KJ. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. 2021;44:3048-3068.

[20] Do T, Tran H, Do T, Tjiputra E, Tran Q. Compact Trilinear Interaction for Visual Question Answering. In: Proceedings of the proceedings of the IEEE International Conference on Computer Vision. 2019:392-401.

[21] Yu K, Wang Y, Zeng S, Liang C, Bai X, et al. InkGAN: Generative Adversarial Networks for Ink-And-Wash Style Transfer of Photographs. 2023;3:1220-1233.

[22] Lin TY, Maire M, Belongie S, Hays J, Perona P, et al. Microsoft Coco: Common Objects in Context. In: Proceedings of the European conference on computer vision. Springer. 2014:740-755.

[23] Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002:311-318.

[24] Vedantam R, Zitnick CL, Parikh D. Cider: Consensus-Based Image Description Evaluation. In: Proceedings of the proceedings of the IEEE conference on computer vision and pattern recognition. 2015:4566-4575.

[25] Banerjee S, Lavie A. Meteor: An Automatic Metric for MT Evaluation With Improved Correlation With Human Judgments. In: Proceedings of the proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. 2005:65-72.

[26] Anderson P, Fernando B, Johnson M, Gould S. Spice: Semantic Propositional Image Caption Evaluation. In: Proceedings of the European conference on computer vision. Springer. 2016;382-398.

[27] Kilickaya M, Erdem A, Ikizler-Cinbis N, Erdem E. Re-Evaluating Automatic Metrics for Image Captioning. 2016. Arxiv Preprint: https://arxiv.org/pdf/1612.07600.pdf

[28] Zhou Y, Gupta U, Dai S, Zhao R, Srivastava N, Jin H, Featherston J, Lai YH, Liu G, Velasquez GA, Wang W. Rosetta: A realistic high-level synthesis benchmark suite for software programmable FPGAs. In Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2018:269-278.

[29] Vig J. A Multiscale Visualization of Attention in the Transformer Model. In: Proceedings of the proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Florence, Italy: Association for Computational Linguistics. 2019:37-42.

[30] Chefer H, Gur S, Wolf L. Transformer Interpretability Beyond Attention Visualization. In: Proceedings of the proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).2021:782-791.

[31] Wang W, Guo Y, Shen C, Ding S, Liao G, Fu H, Prabhakar PK. Integrity and Junkiness Failure Handling for Embedding-based Retrieval: A Case Study in Social Network Search. 2023. ArXiv preprint: https://arxiv.org/pdf/2304.09287.pdf

[32] Yue T, Wang H. Deep learning for genomics: A concise overview. 2018. ArXiv preprint arXiv: https://arxiv.org/pdf/1802.00810.pdf

[33] Ye W, Liu X, Yue T, Wang W. A Sparse Graph-Structured Lasso Mixed Model for Genetic Association with Confounding Correction. 2017. ArXiv preprint arXiv: https://arxiv.org/pdf/1711.04162.pdf