MultiModal Machine Learning
Course Project

ReCLIP: A Strong Zero-Shot Baseline for Referring Expression Comprehension

Zhao Zejun
Hao Zihan

2023.01.15

Summary of the Paper

1. Background

In this section, we first describe the task at hand —Referring Expression Comprehension (ReC) and then introduce CLIP, the pre-trained model the authors primarily use.

1.1 Task description

In referring expression comprehension (ReC), the model is given an image and a textual referring expression describing an entity in the image. The goal of the task is to select the object (bounding box) that best matches the expression. Task accuracy is measured as the percentage of instances for which the model selects a proposal whose intersection-over-union (IoU) with the ground-truth box is at least 0.5. In this paper, the authors focus on the zero-shot setting in which they apply a pre-trained model to ReC without using any training data for the task.

1.2 Pre-trained model architecture

The zero-shot approaches that the authors consider are general in that the only requirement for the pre-trained model is that when given a query consisting of an image and text, it computes a score for the similarity between the image and text. In this paper, they primarily use CLIP. CLIP has an image-only encoder and a text-only transformer. They mainly use the RN50x16 and ViTB/32 versions of CLIP. The image encoder takes the raw image and produces an image representation $x \in \mathbb{R}^d$, and the text transformer takes the sequence of text tokens and produces a text representation $y \in \mathbb{R}^d$. The model's probability of matching image i with caption j is given by $exp(\beta x_i^T y_j)/\sum_{k=1}^{N} exp(\beta x_i^T y_k)$, where β is a hyperparameter.

2. ReCLIP

ReCLIP consists of two main components: (1) a region-scoring method and (2) a rule-based relation resolver. In this section, we first describe the region scoring method. However, using controlled experiments on a synthetic dataset, the authors find that CLIP has poor zero-shot spatial reasoning performance. Therefore, they propose a system that uses heuristics to resolve spatial relations.

2.1 Isolated Proposal Scoring (IPS)

The authors' proposed method, isolated proposal scoring, is based on the observation that ReC is similar to the contrastive learning task with which models like CLIP are pre-trained. Therefore, for each proposal, they create a new image in which that proposal is isolated. They consider two methods of isolation - cropping the image to contain only the proposal and blurring everything in the image except for the proposal region. The score for an isolated proposal is obtained by passing it and the expression through the CLIP. To use cropping and blurring in tandem, they obtain a score s_{crop} and s_{blur} for each proposal and use $s_{crop} + s_{blur}$ as the final score.

2.2 Spatial Relation Resolver

Using controlled experiments on a synthetic dataset, the authors find that CLIP has poor zero-shot spatial reasoning relations. So they propose to decompose complex expressions into simpler primitives. The basic primitive is a predicate applying to an object, which they use CLIP to answer. The second primitive is a spatial relation between objects, for which they use heuristic rules.

2.2.1 Predicates

A predicate is a textual property that the referent must satisfy. For example, "the cat" and "blue airplane" are predicates. We write $P(i)$ to say that object i satisfies the predicate P. We model P as a categorical distribution over objects, and estimate $p(i) = Pr[P(i)]$ with the pre-trained model using IPS mentioned above.

2.2.2 Relations

We consider seven spatial relations –left, right, above, below, bigger, smaller, and inside. We write $R(i, j)$ to mean that the relation R holds between objects i and j, and we use heuristics to determine the probability $r(i, j) = Pr[R(i, j)]$. Besides, we also consider superlative relations, which refer to a kind of relation that an object has some relation to all other objects satisfying the same predicate.

2.2.3 Semantic Trees

We first use spaCy to build a dependency parse for the given expression and then extract a semantic tree from the dependency parse, where each noun chunk is modeled as a node and the dependency paths between noun chunks become relations between nodes.

In the tree, each node N contains a predicate P_N and has a set of children; an edge (N, N') between N and its child N' corresponds to a relation $R_{N,N'}$. We define $\pi_N(i)$ as the probability that node N refers to object i, and compute it recursively. For each node N, we first set $\pi_N(i) = p_N(i)$ and then iterate through each child N' and update $\pi_N(i)$ as follows:

$$\begin{aligned}\pi'_N(i) &\propto \pi_N(i) \sum_j Pr[R_{N,N'}(i, j) \wedge P_{N'}(j)] \\ &\propto \pi_N(i) \sum_j r_{N,N'}(i, j) \pi_{N'}(j)\end{aligned}$$

To compute our final score, we ensemble the distribution π_{root} for the root node with the output of plain IPS (with the whole input expression) by multiplying the proposal probabilities elementwise.

3. Experiments

3.1 Datasets

We compare ReCLIP to other zero-shot methods on RefCOCOg, RefCOCO and RefCOCO+. RefCOCO and RefCOCO+ were created in a two-player game, and RefCOCO+ is designed to avoid spatial relations. RefCOCOg includes spatial relations and has longer expressions on average.

3.2 Results

	RefCOCOg		RefCOCO+			RefCOCO		
Model	**Val**	**Test**	**Val**	**TestA**	**TestB**	**Val**	**TestA**	**TestB**
Random	18.12	19.10	16.29	13.57	19.60	15.73	13.51	19.20
Supervised SOTA	83.35	81.64	81.13	85.52	72.96	87.51	90.40	82.67
CPT-Blk w/ VinVL (Yao et al., 2021)	32.1	32.3	25.4	25.0	27.0	26.9	27.5	27.4
CPT-Seg w/ VinVL (Yao et al., 2021)	36.7	36.5	31.9	35.2	28.8	32.2	36.1	30.3
CLIP								
CPT-adapted	22.32	23.65	23.85	21.55	25.92	23.16	21.44	26.95
GradCAM	50.86	49.70	47.83	**56.92**	37.70	42.85	**51.07**	35.21
ReCLIP w/o relations	57.70	57.19	47.43	50.02	43.85	41.97	43.42	39.02
ReCLIP	**59.33**	**59.01**	**47.87**	50.10	**45.10**	**45.78**	46.10	**47.07**
CLIP w/ Object Size Prior								
CPT-adapted	28.98	30.14	26.64	25.13	27.27	26.08	25.38	28.03
GradCAM	52.29	51.28	49.41	59.66	38.62	44.65	53.49	36.19
ReCLIP w/o relations	59.19	59.01	54.66	60.27	46.33	48.53	53.60	40.84
ReCLIP	60.85	61.05	55.07	60.47	47.41	54.04	58.60	49.54

Fig. 1. Accuracy on the RefCOCOg, RefCOCO+ and RefCOCO datasets.

Our Work

1. Re-Implementation

	RefCOCOg		RefCOCO+			RefCOCO		
	Val	Test	Val	TestA	TestB	Val	TestA	TestB
Authors'	59.33	59.01	47.87	50.10	45.10	45.78	46.10	47.07
Ours	59.31	58.93	47.83	50.11	45.10	45.71	46.14	47.14

Tab. 1. Accuracy on RefCOCO/g/+.

2. Analysis

As the paper says, although ReCLIP outperforms the baselines, there is still a considerable gap between it and supervised methods. So we carried out analysis on the results of our re-implementation to see why.

2.1 Frequency Analysis

We first analyse the frequency of words in the failure cases and get the results in Tab.2. From the result, we can see that spatial relation between objects is still the core of the task and the principal challenge in improving the system is making relation-handling more accurate and at the same time more flexible.

Word	Frequency
left	710
right	634
on	574
in	490
guy	465
man	456
person	391
the	308
woman	243
shirt	220

Tab. 2. Top 10 frequent words in failure cases of RefCOCO TestA.

2.2 Case Analysis

Beyond the frequency analysis, we also propose further case analysis to categorize the failure cases.

2.2.1 Cases that can be ambiguous

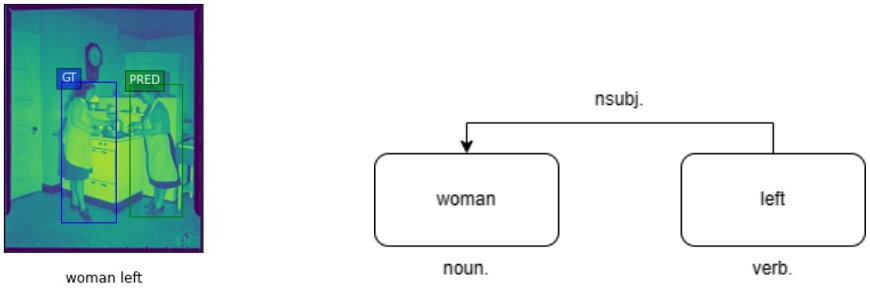
ReCLIP is likely to fail on cases shown in Fig.2 where there exist multiple valid objects. The reason why this happens is that isolated proposals generated by IPS lose the information whether themselves are the major objects in the origin image so some objects that are much less conspicuous in the background can be finally chosen. This is different from us human.

(a) woman (b) yellow shirt

Fig. 2. Examples of cases that can be ambiguous.

2.2.2 Cases where the dependency parses built are not correct

We notice that keywords like "left" and "right", which we specifically look at while resolving spatial relation, are ambiguous in English and this can be confusing to spaCy. For example, as shown in Fig.3, "left" is mistakenly thought to be a verb by spaCy. Therefore, our resolver doesn't realize the existence of the superlative relation.



(a) woman left (b) dependency parse

Fig. 3. Examples of cases where the dependency parses built are not correct.

2.2.3 Cases where the relation interpreter is not flexible enough

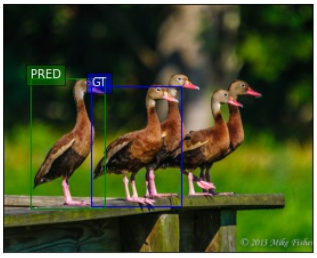
ReCLIP's relation interpreter is more like hard-coded and deterministic so it's not flexible enough on specific cases shown in Fig.4. In Fig.4(a), the vehicles are running into the paper so our heuristic rule of "behind" that object a is "behind" object b only when the center of object a is above that of object b, just doesn't work. Case in Fig.4(b) is similar.

(a) car behind blue bus (b) plant holder behind dog

Fig. 4. Examples of cases where the relation interpreter is not flexible enough.

2.2.4 Cases where the semantic trees extracted are not correct

This kind of error is directly related to the structure of semantic tree supported and the procedure of entity extraction. Due to the flexibility of languages, a rule-based extraction cannot handle all cases equally well. Cases where captions involve counting and predicative are the most representative of this kind of error, which can result in miss of correct root node of the semantic tree. Examples are in Fig.5.



second duck from left

(a) second duck from left

right woman serving cake

(b) right woman serving cake

Fig. 5. Examples of cases where the semantic trees extracted are not correct.

3. Improvements

In view of the different categories of failure cases, we propose different modifications upon the origin ReCLIP and carry out series of experiments.

3.1 On the dependency parse and related entity extraction

As we can see from Sec.2.2.2, ambiguity in language results in incorrect dependency parse from spaCy. Through our observation, in these cases, the target noun chunk is often not the head(root) chunk that we get from the spaCy, for example, the "woman" in Fig.3(b) is a child of "left" but not the root. So we propose a BFS search (depth=1 is enough for ReC task) from the head chunk until we find the target chunk. At the same time, we store the scanned node and check at the end if there exists a superlative relation. Through this method, our ReCLIP can correctly understand the "woman left" in Fig.3 as "leftmost woman".

About the dependency parse, we also check how spaCy deals with "on" and "in" since they appear in Tab.2. We find that spaCy of old version doesn't deal with them reasonably, as shown in Fig.6. So we upgrade the

spaCy used by ReCLIP to v3.4 to build more reasonable dependency parse.

(a) v3.0 (b) v3.4

Fig. 6. Dependency parse of "man on the left" from spaCys of different versions.

The improved ReCLIP outperforms the origin ReCLIP by about 2% on the validation set of RefCOCO and the performance is similar on the validation set of RefCOCOg/+, as shown in Tab.3. Some of the reason could be the different settings of datasets in Sec.3.1.

Model	RefCOCOg	RefCOCO+	RefCOCO
Origin ReCLIP	59.33	47.87	45.78
Improved ReCLIP	**59.44**	**47.88**	**47.98**

Tab. 3. Accuracy on the validation set of RefCOCO/g/+.

3.2 On the spatial relation interpreter

The current spatial relation interpreter of ReCLIP is still not accurate or flexible enough. For example, the origin ReCLIP believes object *a* is in the "left" of object *b* if the center of object *a* is in the left of that of object *b* in the image. However, with arms open, a person on the right is easily to have a center on the left, which breaks the rule. So we decide to modify and extend the current interpreter to better fit the task. Different ideas are proposed here to fix the problem.

3.2.1 Keep using deterministic heuristics

We still decide the spatial relation between two objects by computing a boolean value according to deterministic rules and mainly focus on modifying the heuristic rules to improve the system. We first try to make ReCLIP thinks object a is in the "left" of object b only if the center and the left/right border of object a are all in the left of those of object b in the image. Similar modification is also applied to other relations and we get the results in Tab.4. We can see that our more strict rules make ReCLIP perform even worse than the origin version. We also try other modification like comparing the center and the border of different boxes but the results are still worse than the origin.

Model	RefCOCOg	RefCOCO+	RefCOCO
Origin ReCLIP	**59.33**	**47.87**	**45.78**
Improved ReCLIP	58.35	47.65	43.85

Tab. 4. Accuracy on the validation set of RefCOCO/g/+.

3.2.2 Switch to random variables

We try to replace the boolean value using a random variable to represent the probability of existence of certain spatial relation. Our idea is to ensemble the origin boolean value proposed by the paper with our new rule-based boolean value by computing their weighted average. The results using different parameters are shown in the Tab.5. The performance is still not good. We analyse the source code and find out that the procedure of probability update involves multiplys of relation matrix and probability vector, which will cause problems if the elements of relation matrix are random variables between 0 and 1. So the definitions of the random variables of spatial relations should be more careful and reasonable.

Model	RefCOCO
Origin ReCLIP (weight=1.0)	**45.78**
Improved ReCLIP (weight=0.5)	44.06
Improved ReCLIP (weight=0.7)	44.07
Improved ReCLIP (weight=0.9)	44.07

Tab. 5. Accuracy on the validation set of RefCOCO.

3.3 On the extraction of semantic tree

The core part of the spatial relation resolver is how to extract a correct semantic tree from the dependency parse using a rule-based system. We usually take the tokens between two noun chunk to extract their relation. However, as the paper says, tokens like "with", which themselves have ambiguity, cannot be handled correctly by ReCLIP. The authors simply drop "with" and the following words. We modify this part to catenate the noun chunks on both sides of "with" as a new noun chunk.

We also find that some abbreviation will lead to cases with zero noun chunks, which even don't have root nodes for semantic trees. These cases is usually either a single noun chunk or a superlative object. So we just search for keywords of superlatives using all the tokens from the caption to fix this problem.

Results of this part are shown below in Tab.6.

Model	RefCOCOg	RefCOCO+	RefCOCO
Origin ReCLIP	**59.33**	47.87	45.78
Improved ReCLIP ("with" handled)	**59.33**	47.83	45.73
Improved ReCLIP (abbreviation handled)	58.93	**47.98**	**46.45**

Tab. 6. Accuracy on the validation set of RefCOCO/g/+.

3.4 Combined

We combine the useful methods (which outperform origin ReCLIP) and propose a improved ReCLIP. The final performance are shown below in Tab.7.

Model	RefCOCOg		RefCOCO+			RefCOCO		
	Val	Test	Val	TestA	TestB	Val	TestA	TestB
Origin ReCLIP	**59.33**	**59.01**	47.87	50.10	45.10	45.78	46.10	47.07
Improved ReCLIP	59.01	58.60	**48.06**	**50.16**	**45.39**	**48.63**	**48.06**	**50.82**

Tab. 7. Accuracy on RefCOCO/g/+.

Division of work

Part		Person in charge
1		Hao Zihan, Zhao Zejun
2.1		Hao Zihan
2.2		Zhao Zejun
3.1		Zhao Zejun
3.2	3.2.1	Hao Zihan, Zhao Zejun
	3.2.2	Zhao Zejun
3.3		Zhao Zejun
3.4		Zhao Zejun
Report Writing		Zhao Zejun

Tab. 8. Division of work.

Experiments

Repository

https://github.com/jellllly420/reclip

Notebook

https://colab.research.google.com/drive/1O3Khgw270fKZcPlbOeTIWPiwJVxNtjSv?usp=sha