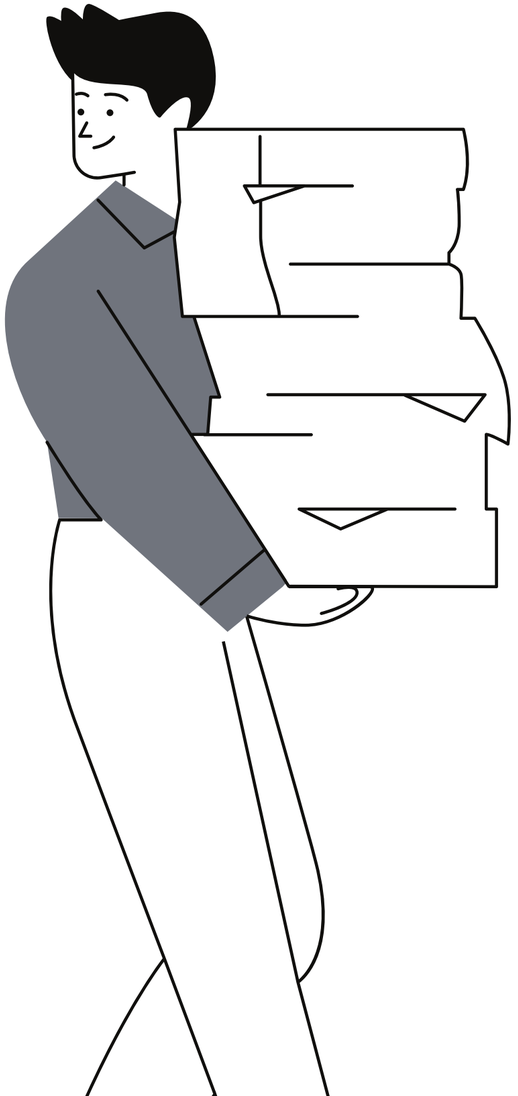
Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 11, 2021

Graph Structured Network for Image -Text Matching

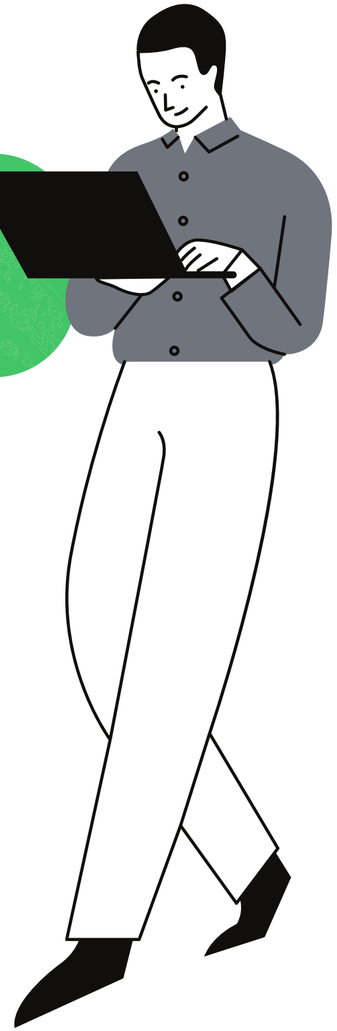
Digital Image Processing Presentation

made by
Madalina-Alina Racovita



1. Introduction
2. Related work
3. Method
4. Experiments
5. Conclusion

1 Introduction

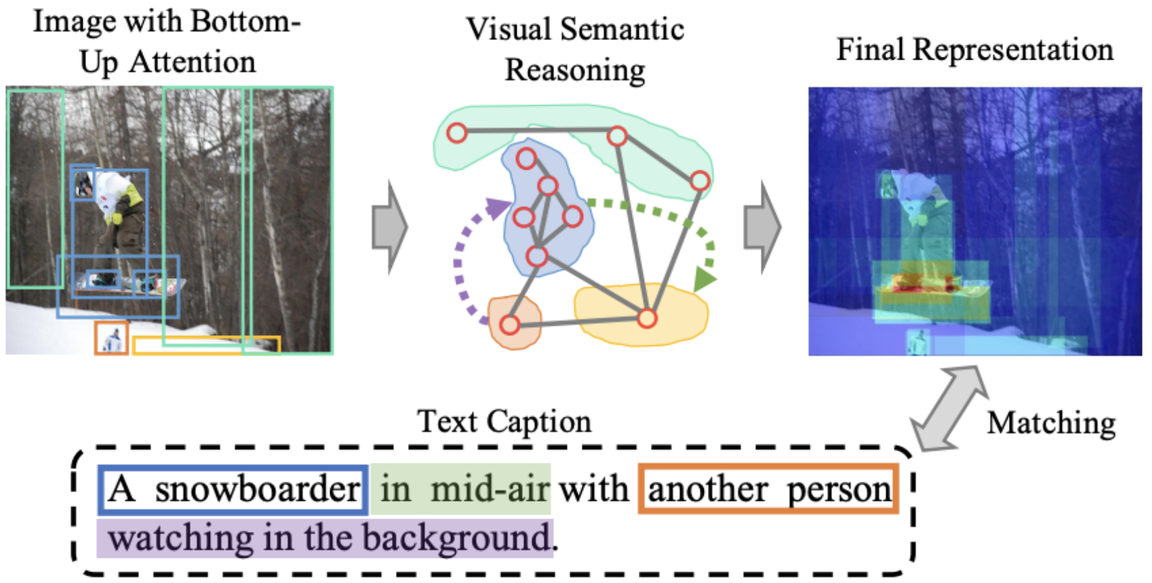


Introduction

Image-text matching has received growing interest since it bridges vision and language. The key challenge lies in

how to learn correspondence between image and text,

such that can reflect similarity of image-text pairs accurately.



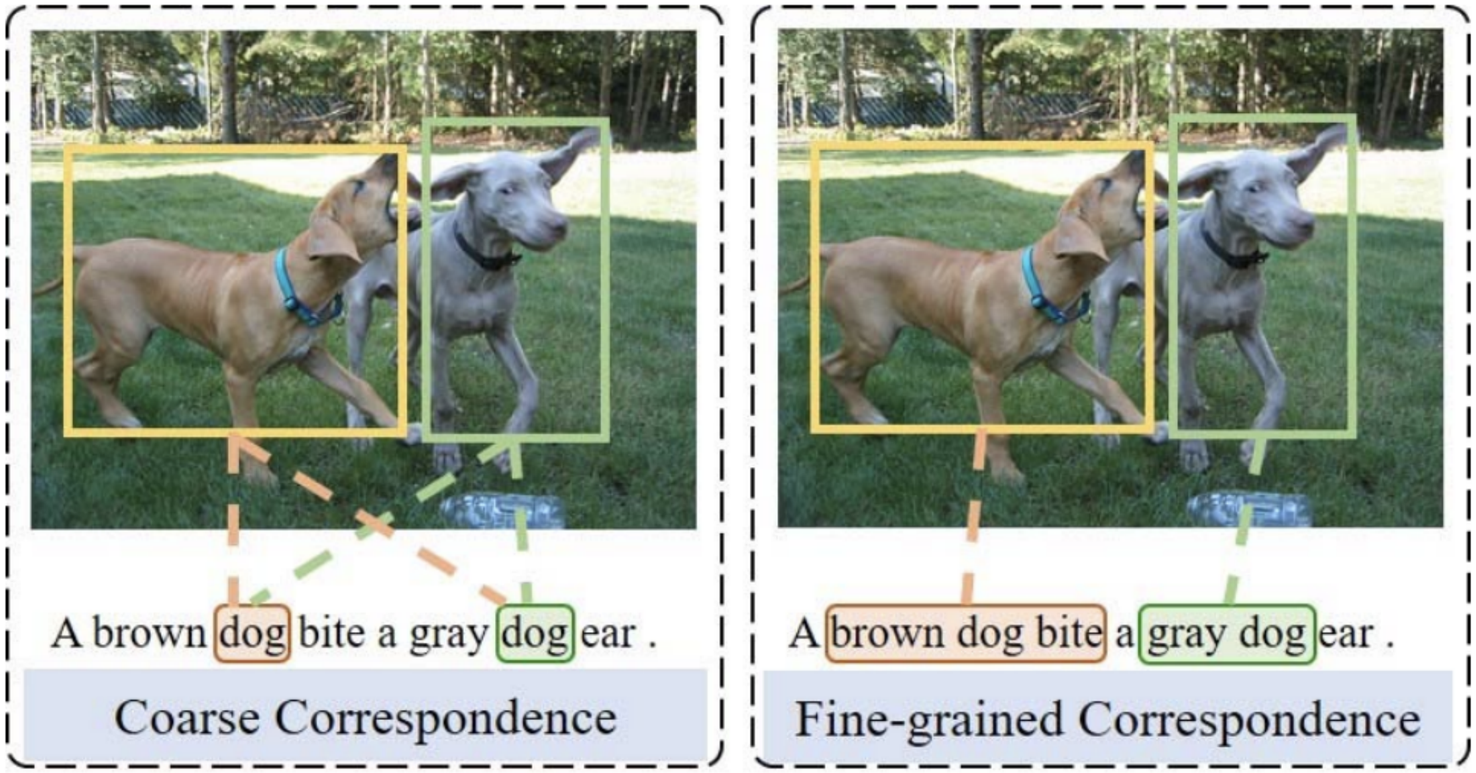
Why do we need this image-text matching and which are the challenges it brings in?

1. It bridges the **vision and language** areas which has potential to **improve the performance of** other **multimodal applications.**

2. The **current representation of image usually lacks global semantic concepts** as in its corresponding text caption.

3. **Existing works learn coarse correspondence** based on object co-occurrence statistics, while **failing to learn fine-grained phrase correspondence**.

Coarse vs. **Fine-Grained** Correspondence



In the left figure, the two dogs are coarsely correlated with the word “dog”, while neglecting their relation and attribute (**bite or being bitten? gray or brown?**).

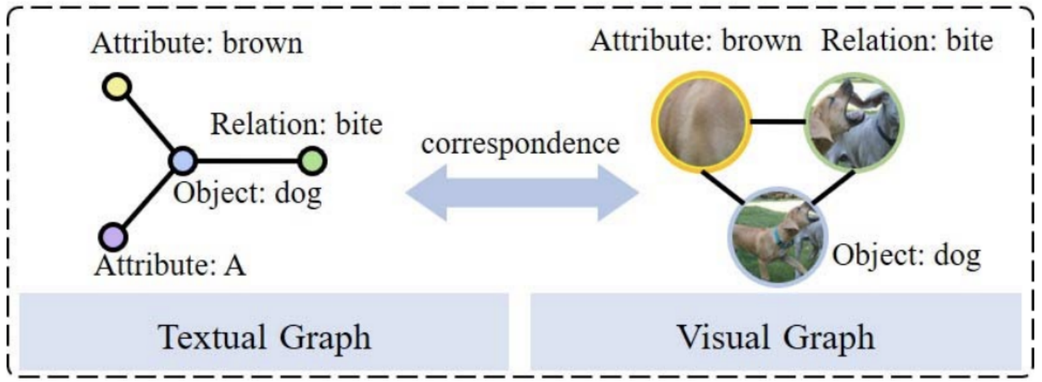
In the right figure, the **gray and brown dogs are fine-grained correlated** with finer textual details, which is **achieved by learning phrase correspondence using a graph-based method**.

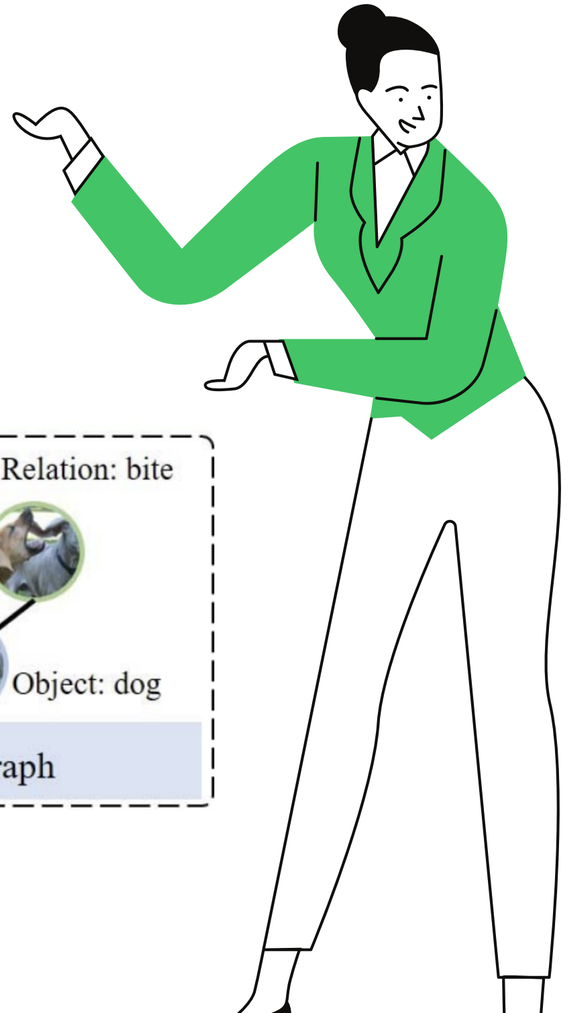
How fine-grained correspondence can be achieved?

✓ Using Graph Structured Matching Network(GSMN).

What are GSMN exactly doing?

explicitly models **object, relation and attribute** as a structured phrase





What are GSMN exactly doing?

Not only allows to learn **correspondence of object, relation and attribute separately**, but also benefits to **learn fine-grained correspondence of structured phrase**. This is achieved by **node-level matching** **and** **structure-level matching**.

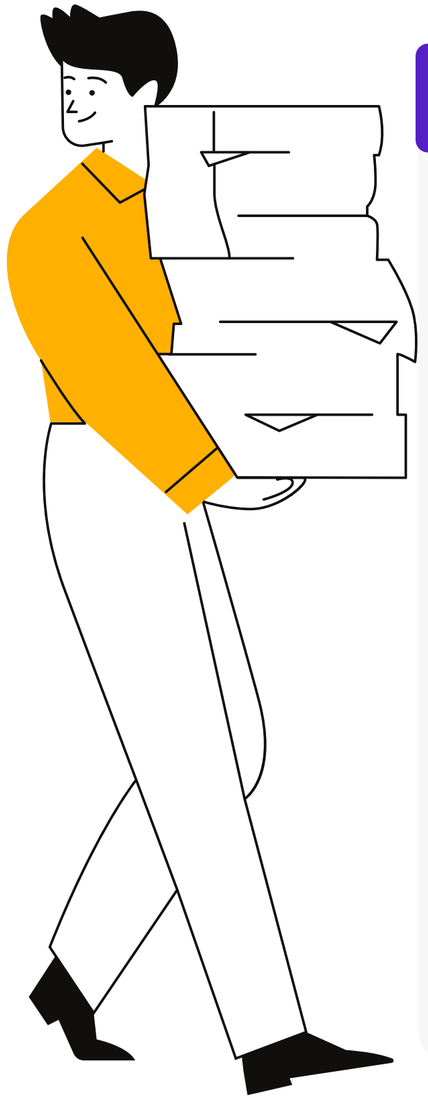
The **node-level matching** **associates each node with its relevant nodes from another modality,** where the node can be object, relation or attribute. The associated nodes then **jointly infer fine-grained correspondence by fusing neighborhood associations at structure-level matching**

2 Related work

Related work

Existing works learn correspondence of image and text based on object co-occurence, which is roughly categorized into **two types:**

1. **global correspondence,** which learns the ***correspondence between the whole image and sentence***
2. **local correspondence,** learns the one ***between local region and word.***

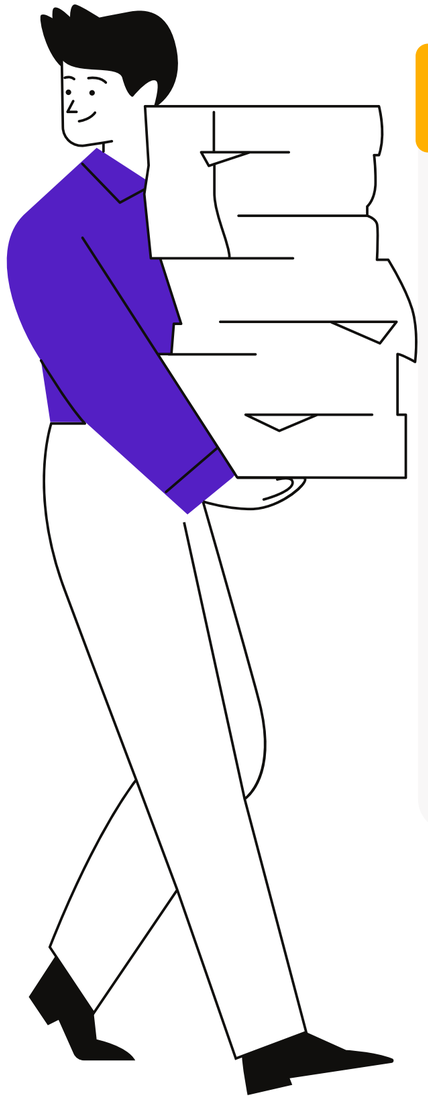


Global correspondence: HOW?

IMAGE-SENTENCE

A main line of research on this field is to first **represent image and text** as **feature vectors**, and then **project them into a common space optimized by a ranking loss**.

1. **Liu et al.** propose to **densely correlate image and text exploiting residual blocks.**
2. **Gu et al.** imagine what the matched instance should look like, and **improve the correspondence of target instance to this imagined instance**.
3. **Wang et al.** point out that **the correspondence within the same modality should also be preserved while learning correspondence in different modalities.** Based on this observation, Wu et al. preserve **graph structure among neighborhood images or texts.** Such global correspondence learning methods cannot learn correspondence of image and text accurately, because primary objects play the dominant role in the global representation of image-text pairs while secondary objects are mostly ignored.



Local correspondence: HOW?

REGION-WORD

The local correspondence is based on learning methods that can obtain region-word correspondence. Some works focus on l**earning correspondence of salient objects.**

1. **Ji et al. exploit saliency model to localize salient regions,** and hence the region-word can be correlated more accurately. A lightweight **saliency model** is employed using an **external saliency dataset as a supervision**.

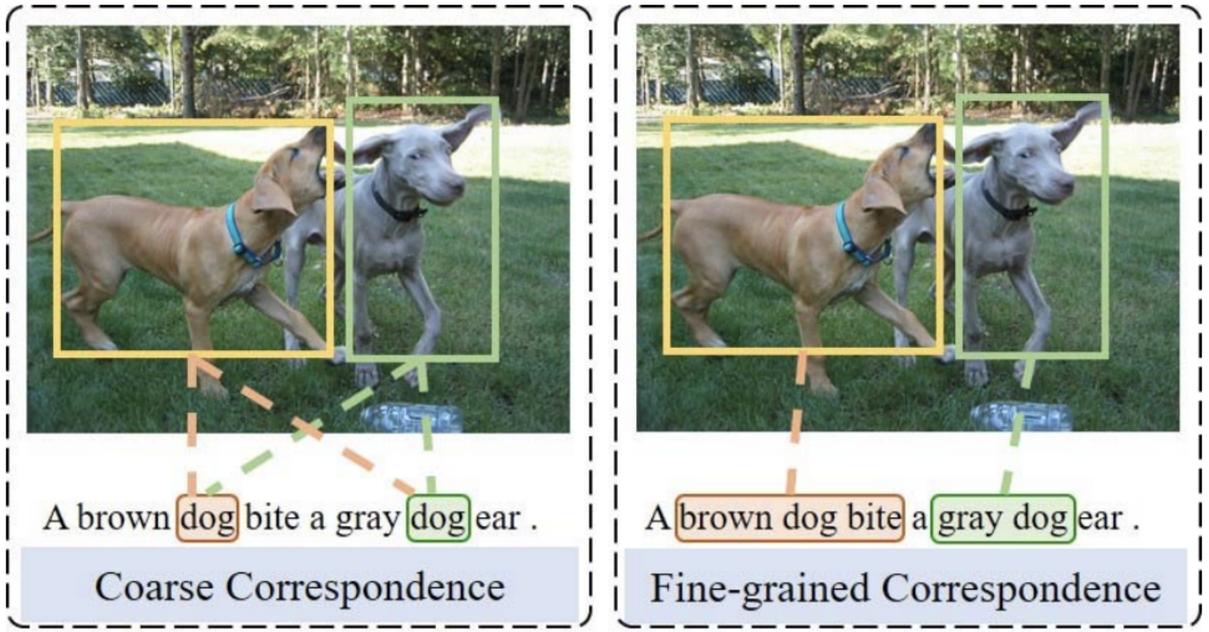
2. A recent approach **SCAN** greatly improves the matching performance. They learn **region-word correspondence using attention mechanism,** where each region corresponds to multiple words and vice versa.

Limitations of the existent work

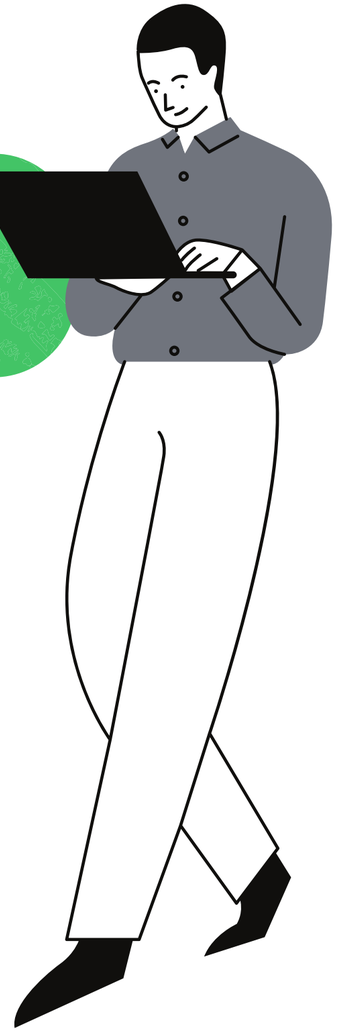
Existing works only learn coarse correspondence based on object co-occurrence statistics, while ***failing to learn fine-grained correspondence of structured object, relation and attribute.***

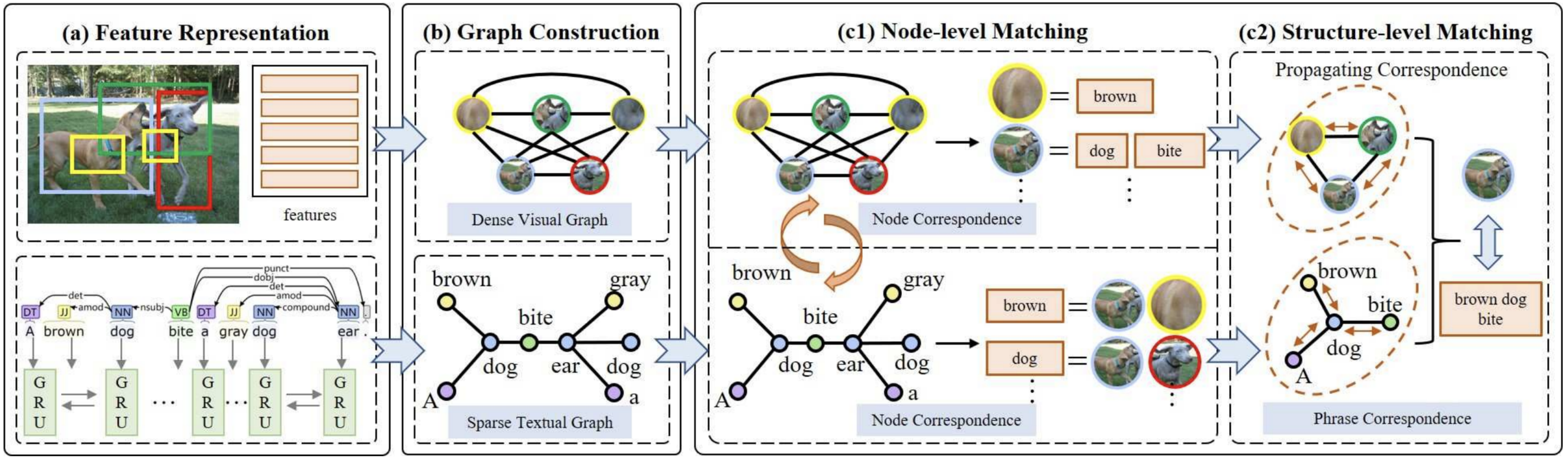
As a result, they suffer from **two limitations**:
(1) it is **hard to learn correspondences of the relation and attribute** as they are **overwhelmed by object correspondence.**
(2) **objects are prone to correspond to wrong categories** **without the guidance of descriptive relation and attribute**.

As shown in picture, the coarse correspondence will incorrectly ***correlate the word "dog" with all the dogs in the image, while neglecting dogs are with finer details***, i.e. brown or gray.



3 Method





1

Faster-RCNN
Stanford CoreNLP
to detect salient regions, and parse the semantic dependency, respectively

2

Nodes: object, relation or attribute
Edge exists if 2 nodes are **semantically dependent** (i..e the nodes interact with each other). The node of an object will connect with the node of its relations or attributes.

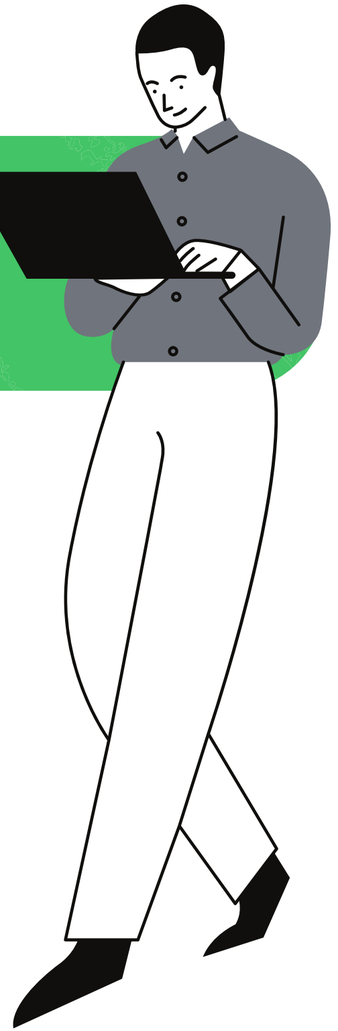
3

Node-level Matching: learn correspondence of object, relation and attribute separately

4

Structure-level Matching: Propagating the learned correspondence to neighbors to jointly infer fine-grained phrase correspondence

3.1 Graph construction



Graph Structured Network for Image -Text Matching

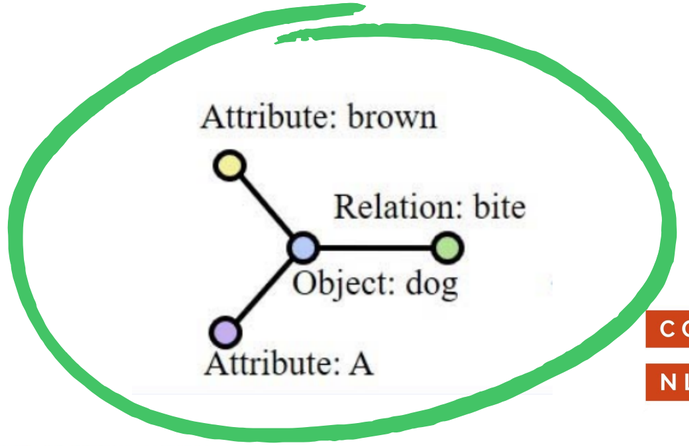
January 11, 2021

Textual graph

The idea is to construct an **undirected sparse graph G1 = (V1, E1) for each text.** It is used the matrix A to represent the **adjacent matrix of each node**, and add self-loops. The **edge weight** is denoted as a matrix W_e, which shows the semantic dependency of nodes. To achieve this, the **similarity matrix S of word representation u** is computed this way:

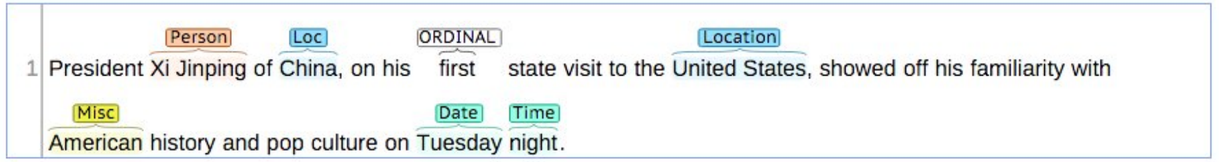
$$s_{ij} = \frac{\exp(\lambda u_i^T u_j)}{\sum_{j=0}^{m} \exp(\lambda u_i^T u_j)}.$$

$$W_e = \|S \circ A\|_2 .$$

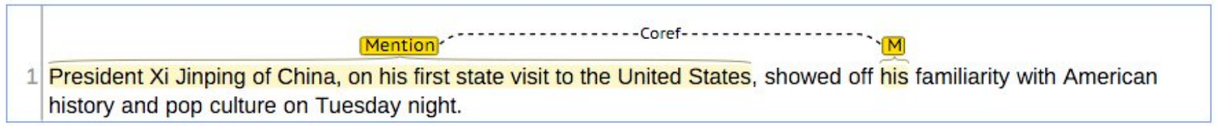


CORE NLP

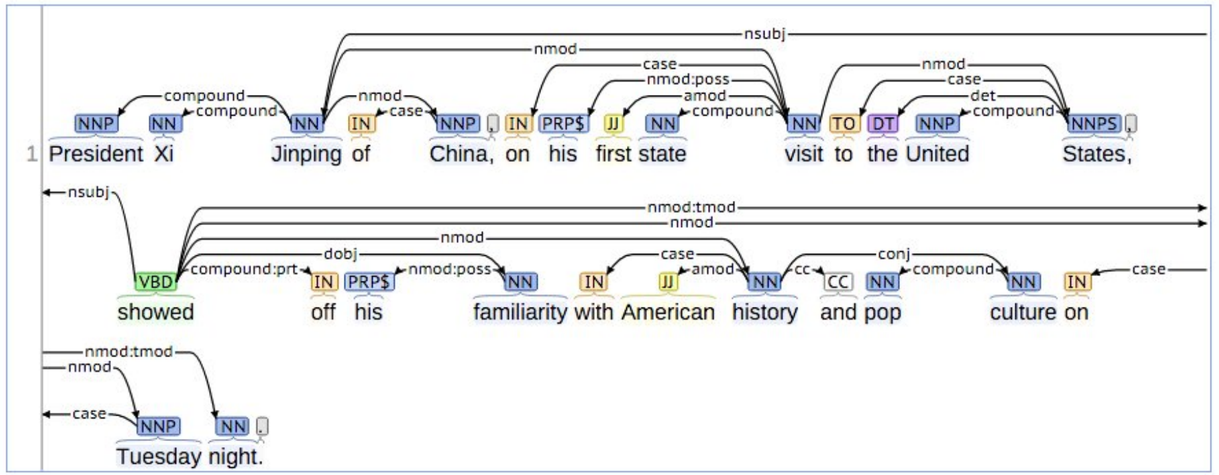
Named Entity Recognition:



Coreference:



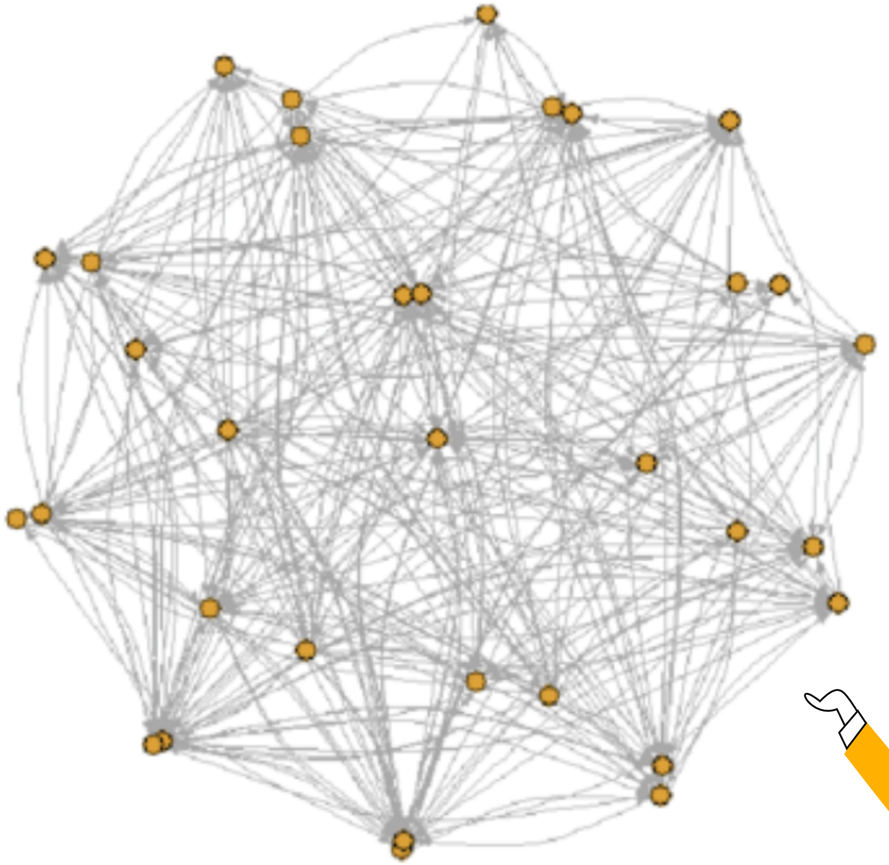
Basic Dependencies:

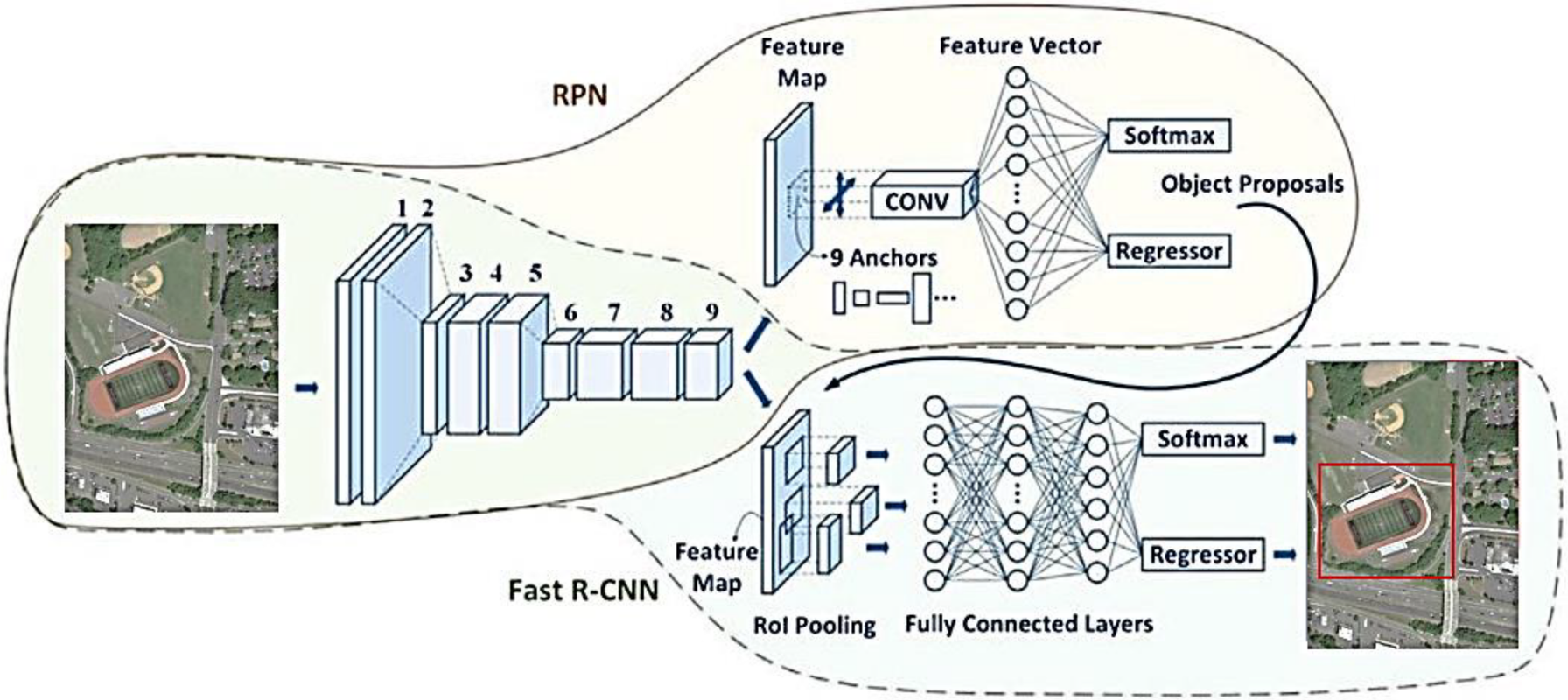


Visual graph

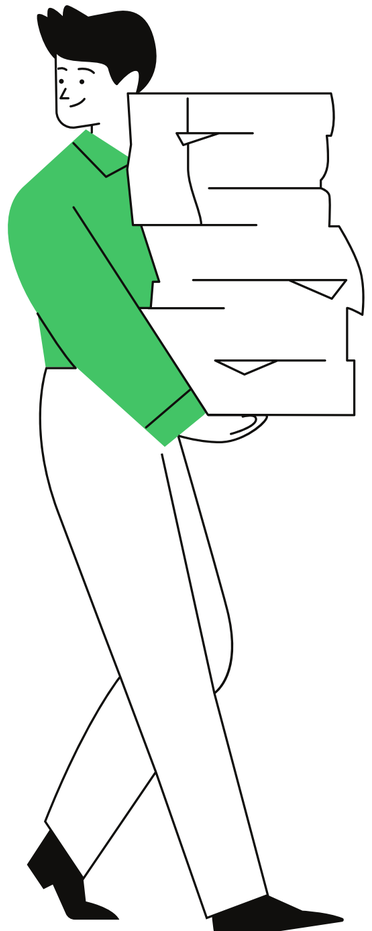
To construct the **visual graph G2 = (V2, E2)**, they represent each image as an **undirected fully-connected graph,** where the **node is set as salient regions detected by Faster-RCNN**, and each node is associated with all the other nodes.

They also use **the polar coordinates to model the spatial relation of each image,** which disentangles the orientation and distance of pair-wise regions.

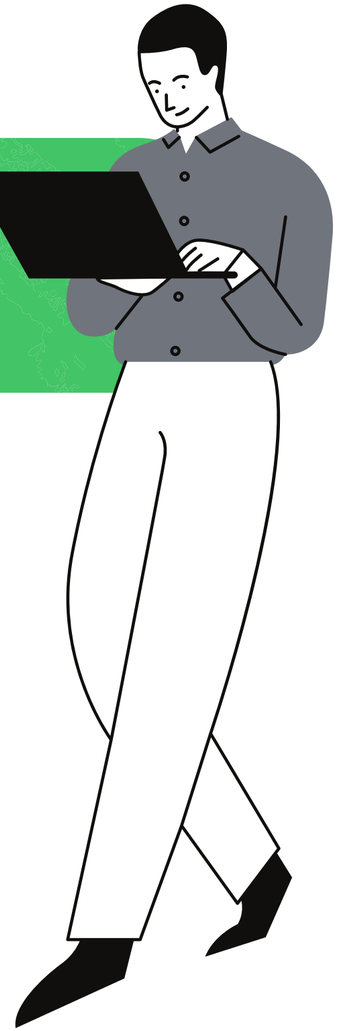




Faster R-CNN architecture



3.2 Multimodal graph matching

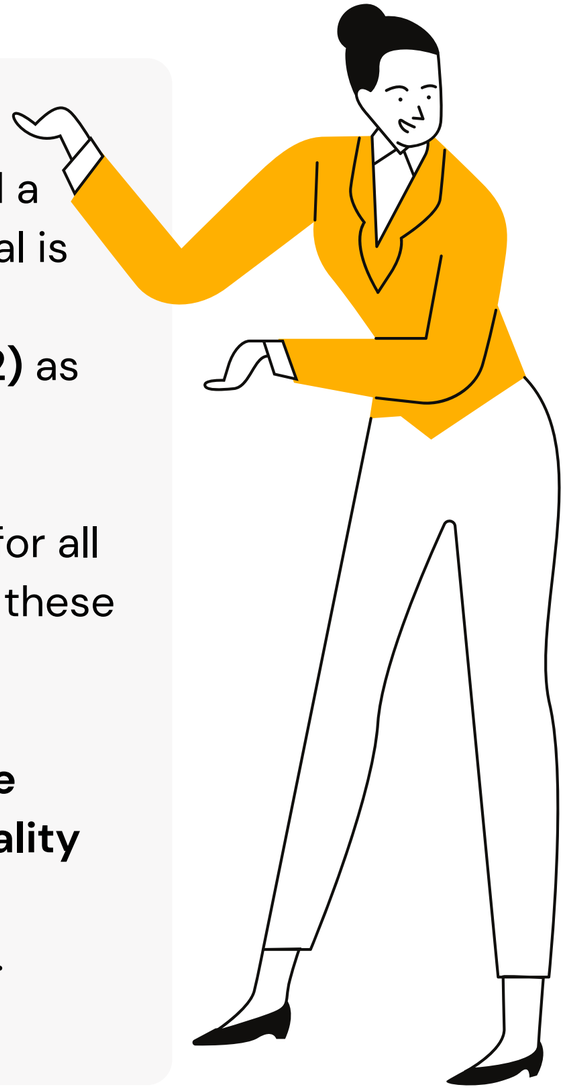


Multimodal graph matching

Given a textual graph **G1 = (V1, E1)** of a text, and a visual graph **G2 = (V2, E2)** of an image, their goal is to match two graphs to learn fine-grained correspondence, producing **similarity g(G1, G2)** as **global similarity of an image-text pair**.

Firstly they compute all nodes representation for all images and texts. To compute the similarity of these heterogeneous graphs:

1. perform **node-level matching to associate each node with nodes from another modality graph**, i.e. ***learning node correspondence***
2. then perform **structure-level matching** i.e. ***learning phrase correspondence***



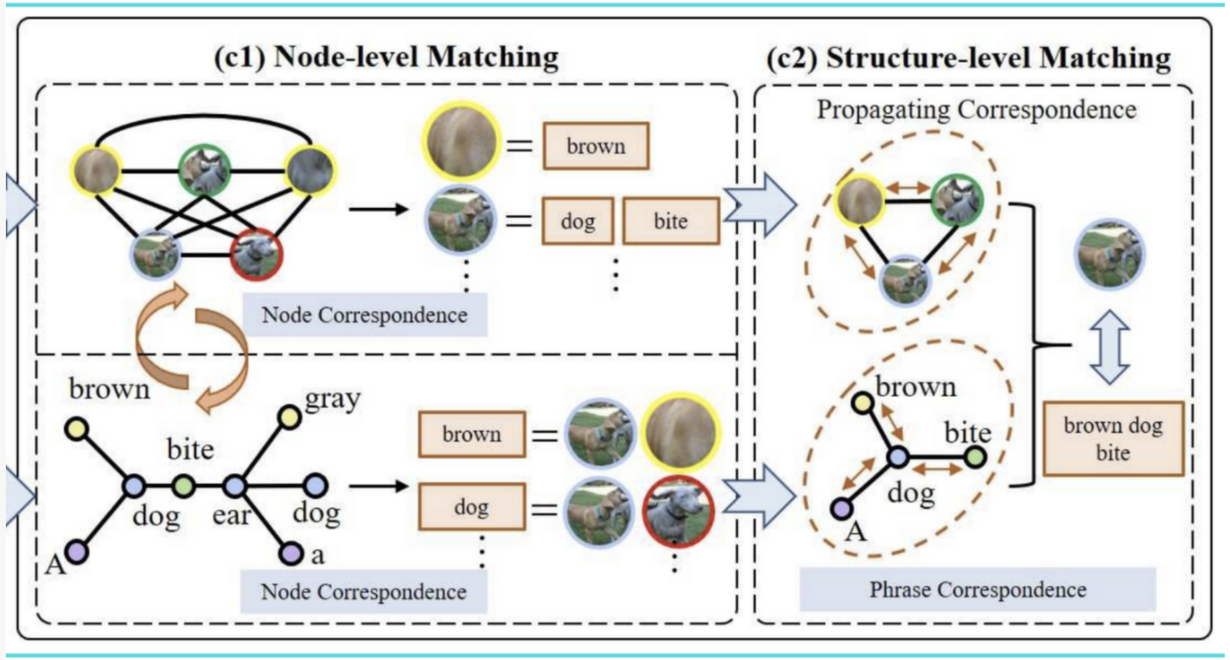
Node-level matching

For this **first compute similarities between visual and textual nodes, followed by a softmax function along the visual axis**. The similarity value measures ***how the visual node corresponds to each textual*** node:

$$C_{t \to i} = \text{softmax}_\beta(\lambda U_\alpha V_\beta^T) V_\beta.$$

Similarly for ***how the textual node corresponds to each visual node:***

$$C_{i \to t} = \text{softmax}_\alpha(\lambda V_\beta U_\alpha^T) U_\alpha$$

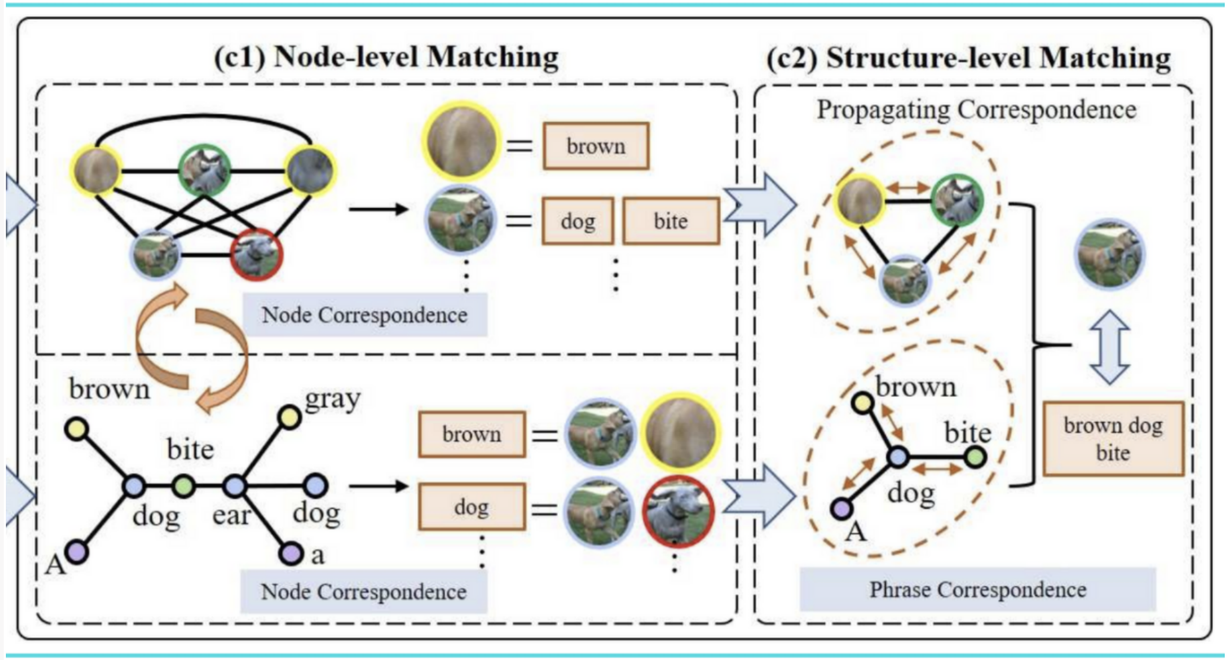


Structure-level matching

The structure-level matching takes the node-level matching vectors as input, and propagates these vectors to neighbors along with the graph edge. Such a design benefits to learn fine-grained phrase correspondence as neighboring nodes guide that.

The matching vector of each node is updated by integrating neighborhood matching vectors using GCN. GCN layer will apply **K kernels** that learn how to integrate neighborhood matching vectors, formulated as:

$$\hat{x}_i = ||_{k=1}^{K} \sigma \left(\sum_{j \in N_i} W_e W_k x_j + b \right)$$



Structure-level matching

They feed the **convolved vectors into a multi-layer perceptron (MLP)** to jointly consider the learned correspondence of all the phrases, and infer the global matching score. This represents **how much one structured graph matches another structured graph.** This process is formulated as:

$$s_{t\to i} = \frac{1}{n}\sum_{i} W_s^u(\sigma(W_h^u \hat{x}_i + b_h^u)) + b_s^u,$$

$$s_{i\to t} = \frac{1}{m}\sum_{j} W_s^v(\sigma(W_h^v \hat{x}_j + b_h^v)) + b_s^v.$$

The **overall matching score** of an image-text pair is computed as the sum of matching score at two directions:

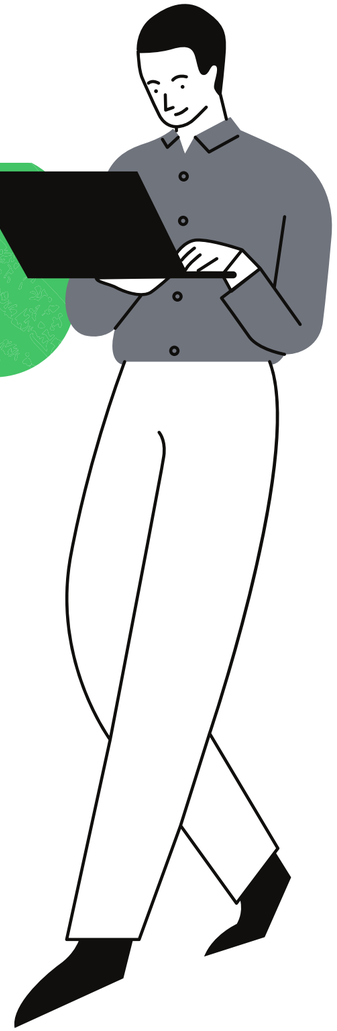
$$g(G_1, G_2) = s_{t\to i} + s_{i\to t}.$$

Objective functions

$$L = \sum_{(I,T)} [\gamma - g(I,T) + g(I,T^{'})]_{+} + [\gamma - g(I,T) + g(I^{'},T)]_{+}.$$

The **similarity in positive pairs should be higher than that in negative pairs** by a margin gamma. I' and T' are hard negatives.

4 Experiments

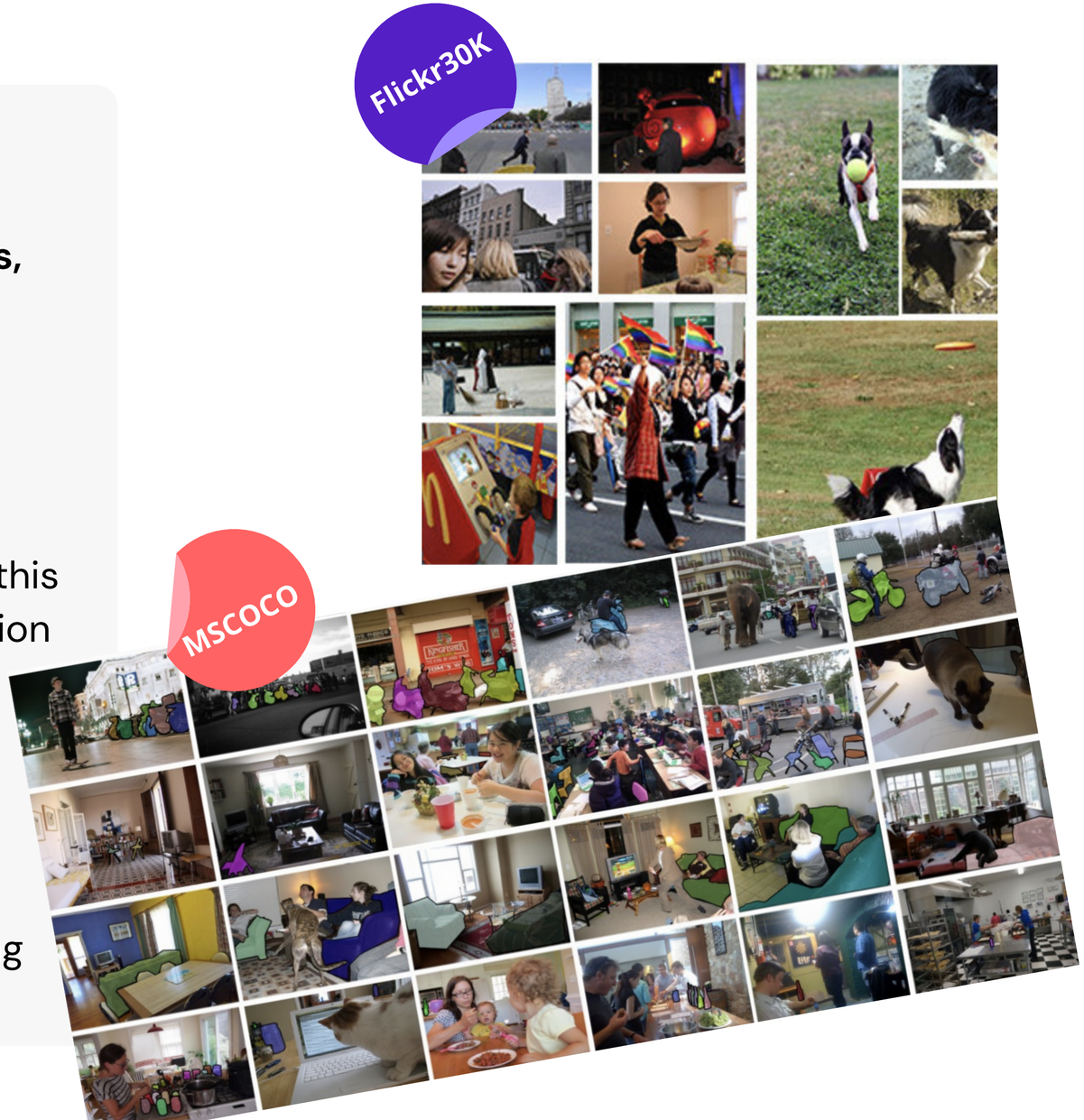


Experiments

To validate the effectiveness of the proposed method, the evaluation was made on **two most widely used benchmarks, Flickr30K and MSCOCO**. Each benchmark contains multiple image-text pairs, where each image is described by five corresponding sentences.

Flickr30K collects **31,000 images** and 31,000 × 5 = **155,000 sentences** in total. Following the settings in previous works, this benchmark is split into 29,000 training images, 1,000 validation images, and 1,000
testing images.

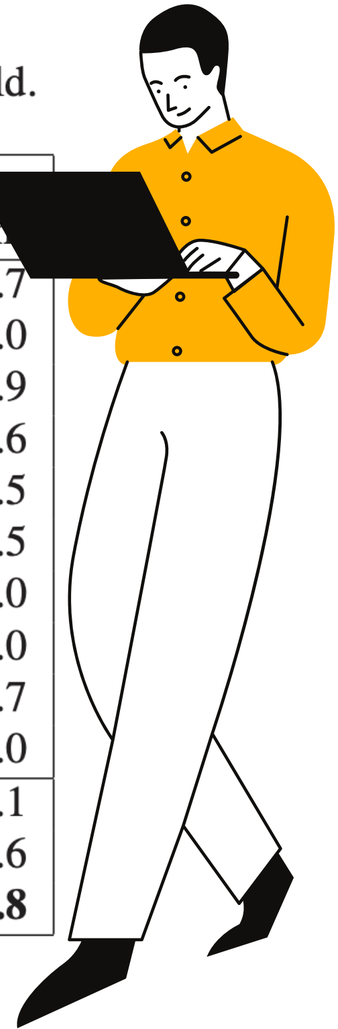
A large-scale benchmark MSCOCO contains **123,287 images** and 123,287 × 5 = **616,435 sentences**, from which they use 113,287 images for training, and both the validation and testing sets contain 5,000 instances.



Results Flickr30K

Table 1: Image-text matching results on Flickr30K, $'ft'$ and $'fixed'$ are fine-tuning and no fine-tuning. The bests are in bold.

Method	Image Backbone	Text Backbone	Image-to-Text			Text-to-Image			rSum
			R@1	R@5	R@10	R@1	R@5	R@10	
m-CNN [20]	fixed VGG-19	ft CNN	33.6	64.1	74.9	26.2	56.3	69.6	324.7
DSPE [29]	fixed VGG-19	w2v+HGLMM	40.3	68.9	79.9	29.7	60.1	72.1	351.0
VSE++ [2]	ft ResNet-152	ft GRU	52.9	79.1	87.2	39.6	69.6	79.5	407.9
TIMAM [27]	fixed ResNet-152	Bert	53.1	78.8	87.6	42.6	71.6	81.9	415.6
DANs [23]	ft ResNet-152	ft LSTM	55.0	81.8	89.0	39.4	69.2	79.1	413.5
SCO [9]	fixed ResNet-152	ft LSTM	55.5	82.0	89.3	41.1	70.5	80.1	418.5
GXN [4]	ft ResNet-152	ft GRU	56.8	-	89.6	41.5	-	80.1	268.0
SCAN [14]	Faster R-CNN	ft Bi-GRU	67.4	90.3	95.8	48.6	77.7	85.2	465.0
BFAN [18]	Faster R-CNN	ft Bi-GRU	68.1	91.4	-	50.8	78.4	-	288.7
PFAN [30]	Faster R-CNN	ft Bi-GRU	70.0	91.8	95.0	50.4	78.7	86.1	472.0
GSMN (sparse)	Faster R-CNN	ft Bi-GRU	71.4	92.0	96.1	53.9	79.7	87.1	480.1
GSMN (dense)	Faster R-CNN	ft Bi-GRU	72.6	93.5	96.8	53.7	80.0	87.0	483.6
GSMN (sparse+dense)	Faster R-CNN	ft Bi-GRU	**76.4**	**94.3**	**97.3**	**57.4**	**82.3**	**89.0**	**496.8**



Results Flickr30K: Top 5 matchings image-text. Mismatches are reported in red.

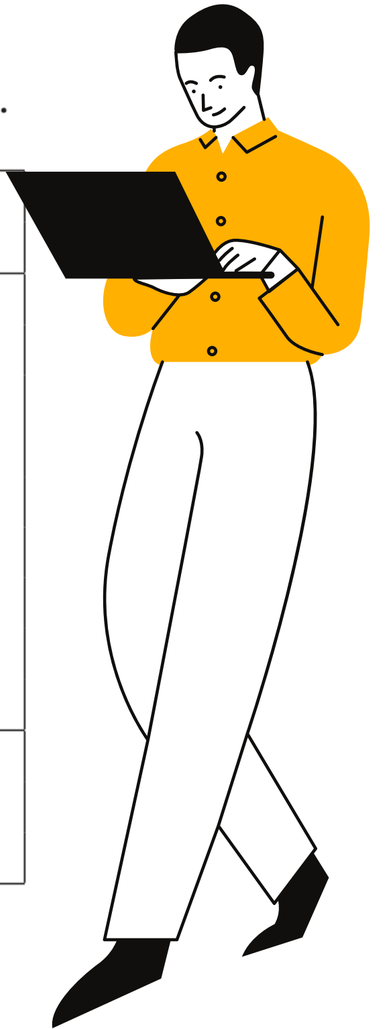
1.People are fixing the roof of a house .
2.Three men , one on a ladder , work on a roof .
3.Group gathered to go snowmobiling .
4.Two men on a rooftop while another man stands atop a ladder watching them
5.Two men sitting on the roof of a house while another one stands on a ladder .

1.An employee is handing a woman a bag while she is browsing through fish on ice at a street market .
2. A lady in a striped shirt is shopping at a fish market .
3. People are shopping for fish at a market .
4. A woman wearing a short-sleeved striped shirt and carrying a green shopping bag examines fish in an open air market while a vendor in a white apron over a gray t-shirt hands her an empty plastic bag .
5. The woman is at marketplace buying fish from an Arabic person .

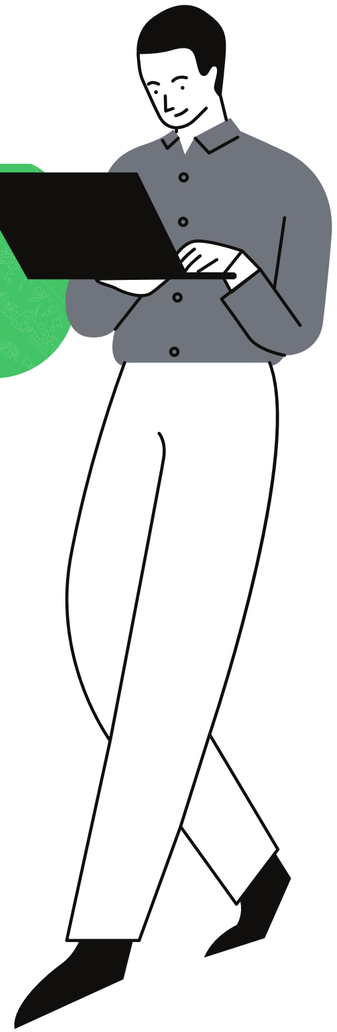
Results MSCOCO

Table 2: Image-text matching results on MSCOCO, $'ft'$ and $'fixed'$ are fine-tuning and no fine-tuning. The bests are in bold.

Method	Image Backbone	Text Backbone	Image-to-Text			Text-to-Image			rSum
			R@1	R@5	R@10	R@1	R@5	R@10	
m-CNN [20]	fixed VGG-19	ft CNN	42.8	73.1	84.1	32.6	68.6	82.8	384.0
DSPE [29]	fixed VGG-19	w2v+HGLMM	50.1	79.7	89.2	39.6	75.2	86.9	420.7
VSE++ [2]	ft ResNet-152	ft GRU	64.7	-	95.9	52.0	-	92.0	304.6
DPC [35]	ft ResNet-152	ft ResNet-152	65.5	89.8	95.5	47.1	79.9	90.0	467.8
GXN [4]	ft ResNet-152	ft GRU	68.5	-	97.9	56.6	-	94.5	317.5
SCO [9]	fixed ResNet-152	ft LSTM	69.9	92.9	97.5	56.7	87.5	94.8	499.3
SCAN [14]	Faster R-CNN	ft Bi-GRU	72.7	94.8	98.4	58.8	88.4	94.8	507.9
BFAN [18]	Faster R-CNN	ft Bi-GRU	74.9	95.2	-	59.4	88.4	-	317.9
PFAN [30]	Faster R-CNN	ft Bi-GRU	76.5	96.3	**99.0**	61.6	89.6	95.2	518.2
GSMN (sparse)	Faster R-CNN	ft Bi-GRU	76.1	95.6	98.3	60.4	88.7	95.0	514.0
GSMN (dense)	Faster R-CNN	ft Bi-GRU	74.7	95.3	98.2	60.3	88.5	94.6	511.6
GSMN (sparse+dense)	Faster R-CNN	ft Bi-GRU	**78.4**	**96.4**	98.6	**63.3**	**90.1**	**95.7**	**522.5**



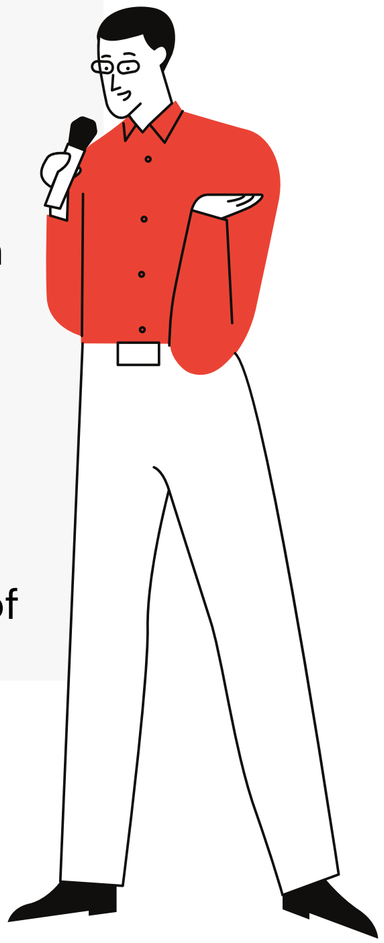
5 Conclusion



Remarks on what we discovered today

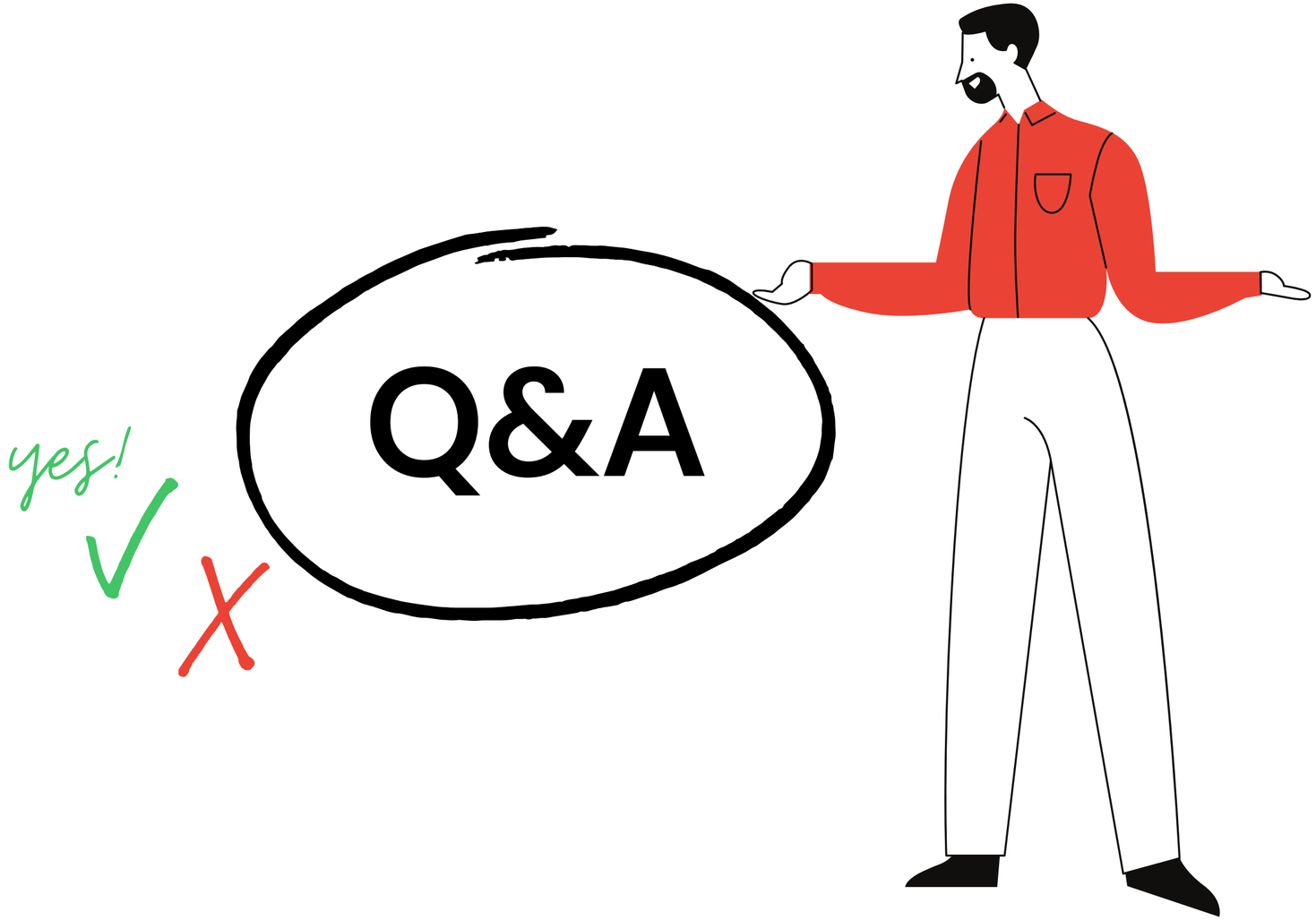
In this paper, there was proposed a **graph structured matching network for image-text matching**, which performs matching on heterogeneous visual and textual graphs. This was achieved by ***node-level matching and structure-level matching*** that infer *fine-grained correspondence* by propagating node correspondence along the graph edge.

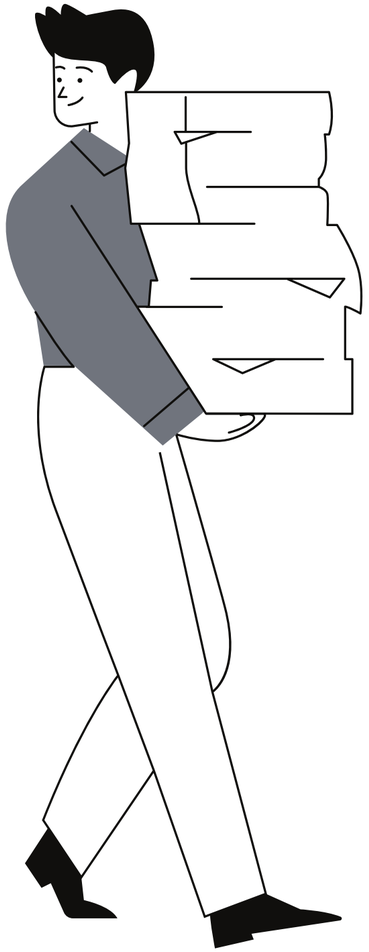
Moreover, **such a design can learn correspondence of relation and attribute, which are mostly ignored by previous works**. With the guidance of relation and attribute, the object correspondence can be greatly improved. Extensive experiments demonstrate the superiority of this network.



Graph Structured Network for Image -Text Matching

January 11, 2021





References

1. **Github repository**: https://github.com/CrossmodalGroup/GSMN

2. **Article**: https://profs.info.uaic.ro/~ancai/DIP/articole/alocate/Racovita%20-%20Liu_Graph_Structured_Network_for_Image-Text_Matching_CVPR_2020_paper.pdf

Thank you!