BHARATH SRIVATSAN
COS 598B, Spring ‘18

OPEN WORLD ANNOTATION WITH SCENE GRAPHS

Roadmap

1. Challenge Overview

- Open World Annotation
- Cognitive Tasks

2. Visual Genome Dataset

- Overview
- Components & Methods
- Previous Work
- Evaluation Metrics
- Findings

3. Scene Graph Applications

- Image Retrieval:
  - Key Challenge
  - Dataset
  - Implementation
  - Experiments/Results
- Scene Graph Generation:
  - Key Challenge
  - Implementation
  - Experiments/Results

1. Open World Annotation
2. Cognitive Tasks

1.

CHALLENGE OVERVIEW

Open World Annotation

Real-world, natural, datasets include:

- Different objects embedded together in complex scenes
- Occlusion of interesting regions
- An open universe with classes not known beforehand

MS-COCO was a good example of this intuition; Visual Genome extends upon it

“Cognitive” Tasks

Involve higher-order questions on images; closer step to “understanding” images:

- Image description synthesis
- Visual Question Answering
- Intuitional leaps (why?, relationships, subjective attributes)

Scene graphs are one way of representing a higher-order “knowledge”

1 Overview

2 Components & Methods

3 Previous Work

4 Evaluation Metrics

5 Findings

2.

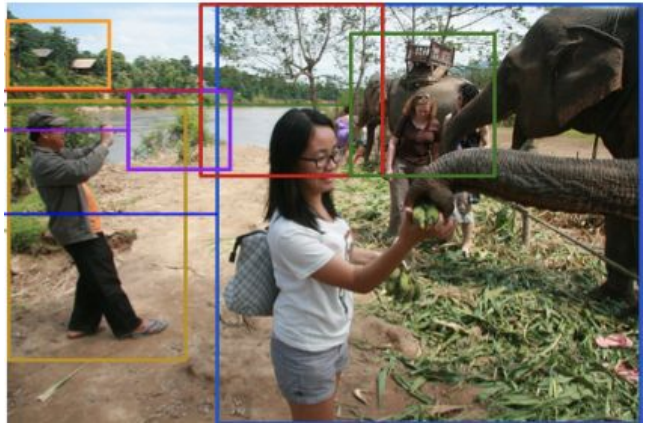
VISUAL GENOME DATASET

***Visual Genome**: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*

By: Krishna et al., 2016

Visual Genome: Overview

For the following slides: Images Credit: *Krishna et al.*



object detection | **object attributes** | **object classification** | **scene classification** | **fine-grained recognition** | **action recognition**

Q: How many people are wearing a lettered, zip-up red jacket?
A: Just one.

Q: What is the most valuable device in this room?
A: The television.

Q: What animal is the balloon modelled after?
A: Blue whale.

Q: Where was the picture taken?
A: At the beach.

Q: What kind of boat is the far left blue boat?
A: Sail boat.

Q: What is the snowboarder doing?
A: Jumping.

text detection | **spatial reasoning** | **event understanding** | **common sense** | **person identification** | **facial expressions**

Q: When was the bridge built?
A: 1932.

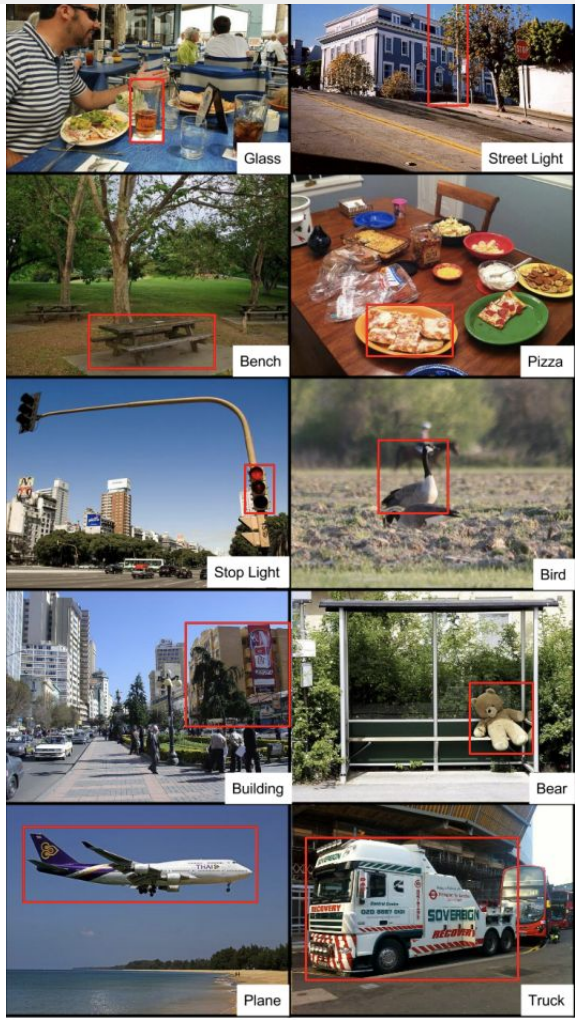
Q: Where is the American flag?
A: Behind president Reagan.

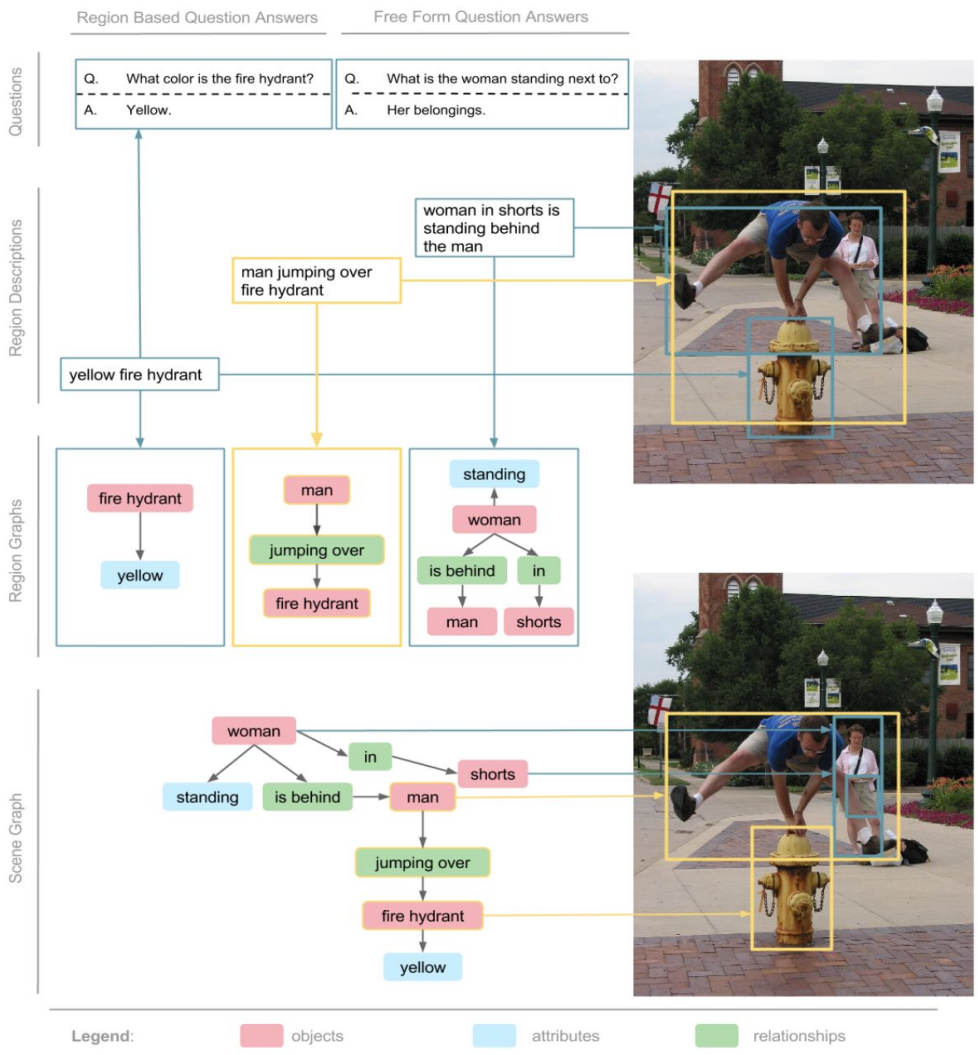
Q: What holiday is being celebrated?
A: Fourth of July.

Q: Why is the man's tie moving?
A: The wind is blowing.

Q: Who is this man?
A: Derek Jeter.

Q: What expression is on most people's faces?
A: They are smiling.





Components

1. Regions & Descriptions
2. Objects & Bounding Boxes
3. Attributes
4. Relationships
5. Region Graphs
6. Scene Graph
7. Regional + Freeform Question-Answer Pairs

1: Regional Descriptions

Intuition: creating region-based descriptions would reduce description of only major features

1. Worker picks three new bounding boxes, describes each region **uniquely** (shown most similar regions)

$$S_n(d_i, d_j) = b(d_i, d_j) \exp\left(\frac{1}{N} \sum_{n=1}^{N} \log p_n(d_i, d_j)\right), \quad b(d_i, d_j) = \begin{cases} 1 & \text{if } len(d_i) > len(d_j) \\ e^{1 - \frac{len(d_j)}{len(d_i)}} & \text{otherwise} \end{cases}$$

2. Algorithm enforces < 0.7 similarity to image-specific & global descriptions
3. Worker draws region boxes judged on coverage

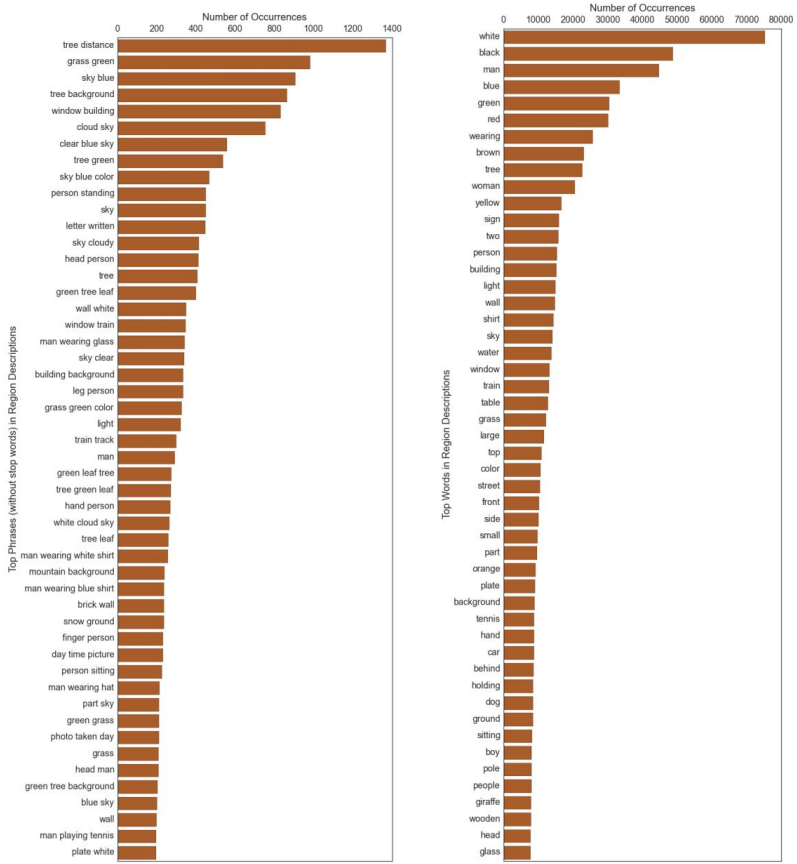
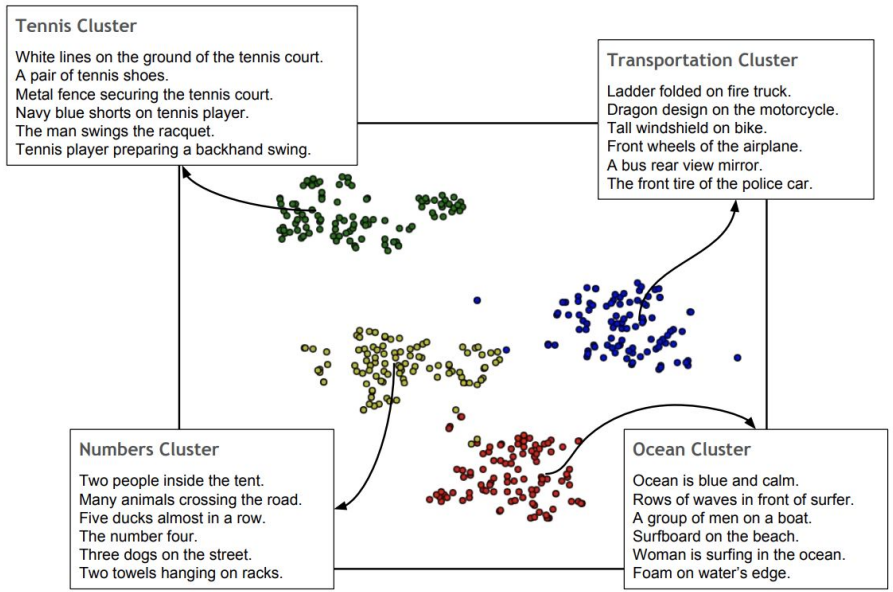


Fig. 18: (a) A plot of the most common visual concepts or phrases that occur in region descriptions. The most common phrases refer to universal visual concepts like "blue sky," "green grass," etc. (b) A plot of the most frequently used words in region descriptions. Each word is treated as an individual token regardless of which region description it came from. Colors occur the most frequently, followed by common objects like `man` and `dog` and universal visual concepts like "sky."

Region Statistics



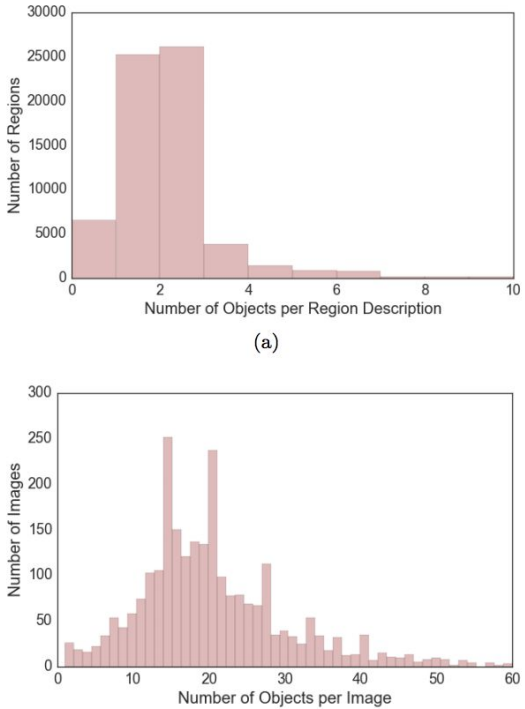
2: Objects & Bounding Boxes

1. Worker given a region and description, extracts objects and draws bounding boxes
2. Bounding boxes drawn to satisfy **both** coverage & quality (4px max error)
3. List of previously-extracted objects (from alternate descriptions) provided
    a. Workers told to join identical objects
    b. Stanford's dependency parser used to suggest most likely nouns

Object Statistics

	Visual Genome	ILSVRC Det. (Russakovsky et al., 2015)	MS-COCO (Lin et al., 2014)	Caltech101 (Fei-Fei et al., 2007)	Caltech256 (Griffin et al., 2007)	PASCAL Det. (Everingham et al., 2010)	Abstract Scenes (Zitnick and Parikh, 2013)
Images	108,077	476,688	328,000	9,144	30,608	11,530	10,020
Total Objects	3,843,636	534,309	2,500,000	9,144	30,608	27,450	58
Total Categories	33,877	200	80	102	257	20	11
Objects per Category	113.45	2671.50	27472.50	90	119	1372.50	5.27

Table 3: Comparison of Visual Genome objects and categories to related datasets.



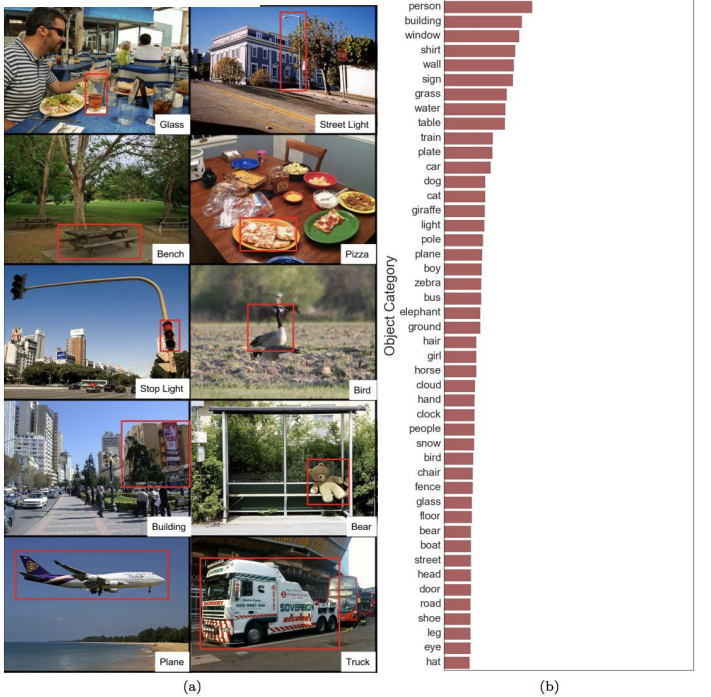


Fig. 22: (a) Examples of objects in Visual Genome. Each object is localized in its image with a tightly drawn bounding box. (b) Plot of the most frequently occurring objects in images. People are the most frequently occurring objects in our dataset, followed by common objects and visual elements like `building`, `shirt`, and `sky`.

3/4: Attributes / Relationships

Given a region description, region image, and object bounding boxes, workers extract attributes/relationships and identify the objects they apply to

Note: some descriptions have no objects, attributes, or relationships

Ex: “It is a sunny day”

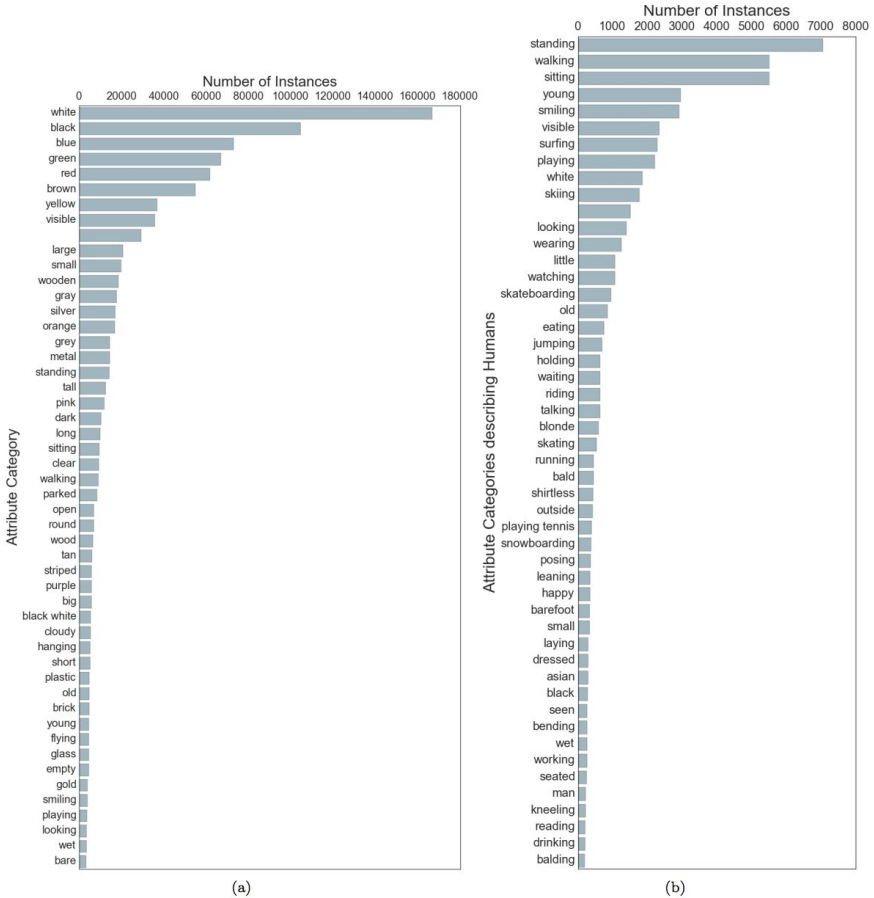


Fig. 24: (a) Distribution showing the most common attributes in the dataset. Colors (e.g. `white`, `red`) and materials (e.g. `wooden`, `metal`) are the most common. (b) Distribution showing the number of attributes describing people. State-of-motion verbs (e.g. `standing`, `walking`) are the most common, while certain sports (e.g. `skiing`, `surfing`) are also highly represented due to an image source bias in our image set.

Attribute/Relationship Statistics

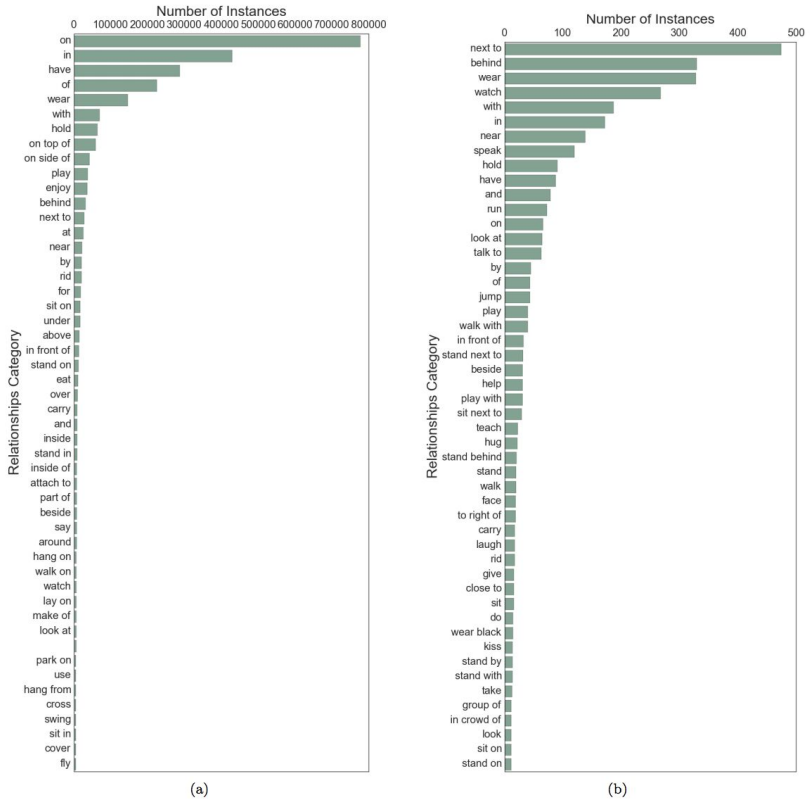
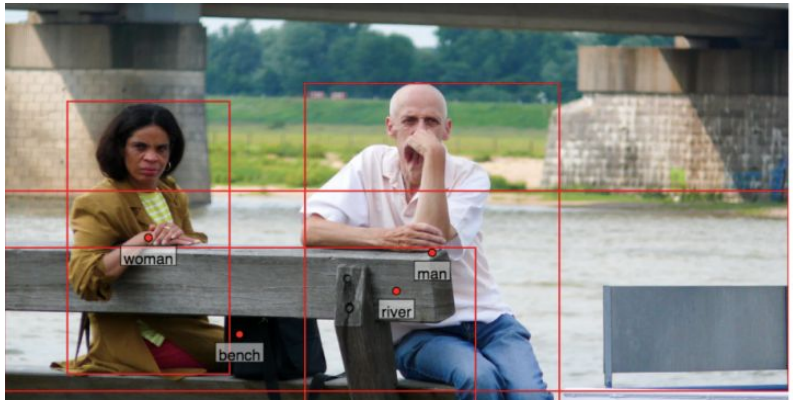
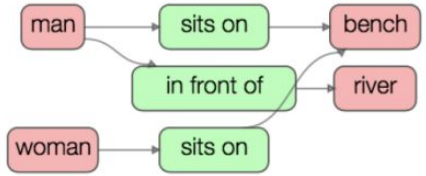


Fig. 27: (a) A sample of the most frequent relationships in our dataset. In general, the most common relationships are spatial (`on top of`, `on side of`, etc.). (b) A sample of the most frequent relationships involving humans in our dataset. The relationships involving people tend to be more action oriented (`walk`, `speak`, `run`, etc.).

5: Regional Graphs

Programmatically created based on worker identification of relationship and attribute links to particular objects in each region

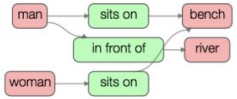




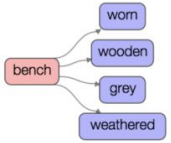
A man and a woman sit on a park bench along a river.

6: Scene Graph

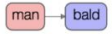
1. Combine objects from different regions with bounding box overlap of > 0.9
2. Ask workers to confirm identity
3. Take union of region graphs, merging at each repeated node



A man and a woman sit on a park bench along a river.



Park bench is made of gray weathered wood



The man is almost bald

(Scene Graph Creation)

7: Question-Answer Pairs

- Freeform Q-A: Worker creates 8 Q-A pairs (>3 categories) per image shown
- Region-based Q-A: random large (>5k pixels, >4 words in phrase) regions selected, workers create a Q-A pair for each

Questions must be precise, unique, unambiguous, and either of type 5Ws or “how”

Canonicalization

All objects, attributes, relationships, and noun phrases mapped to WordNet synsets:

1. Use NLP tools to extract noun phrases / relationship verbs, stem attributes
2. Map each to most frequent synsets
3. Use heuristics to correct common errors
4. Present top 5 potential synsets and definitions to workers for verification

Verification

Two processes used:

- Majority Voting: 3 workers vote on each annotation, 2 must verify correctness
- Rapid Judgements: verification method to speed up process by 10x

Background: Rapid Judgements

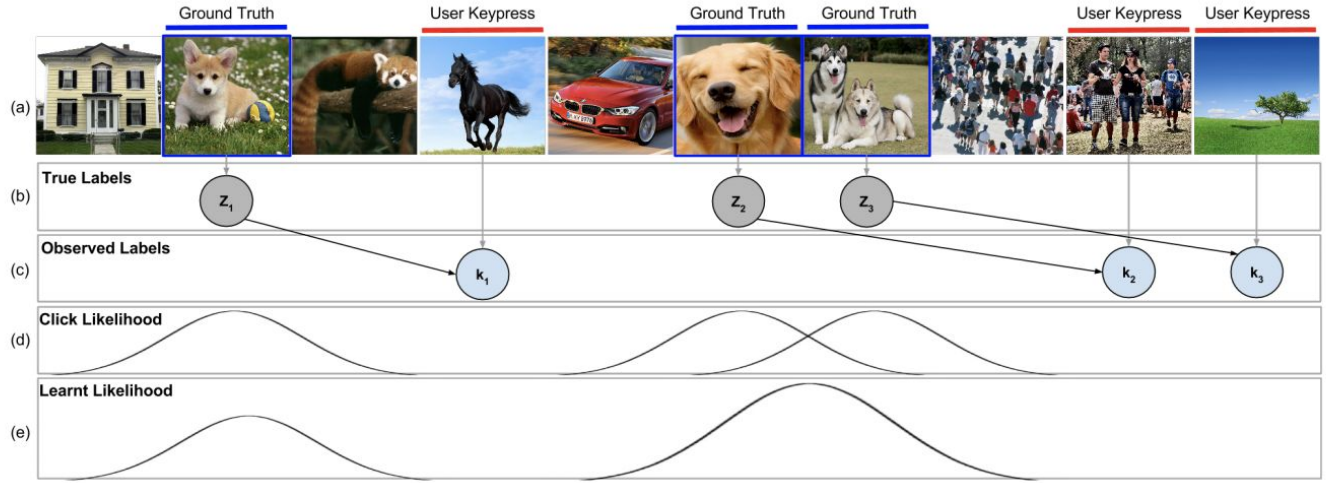
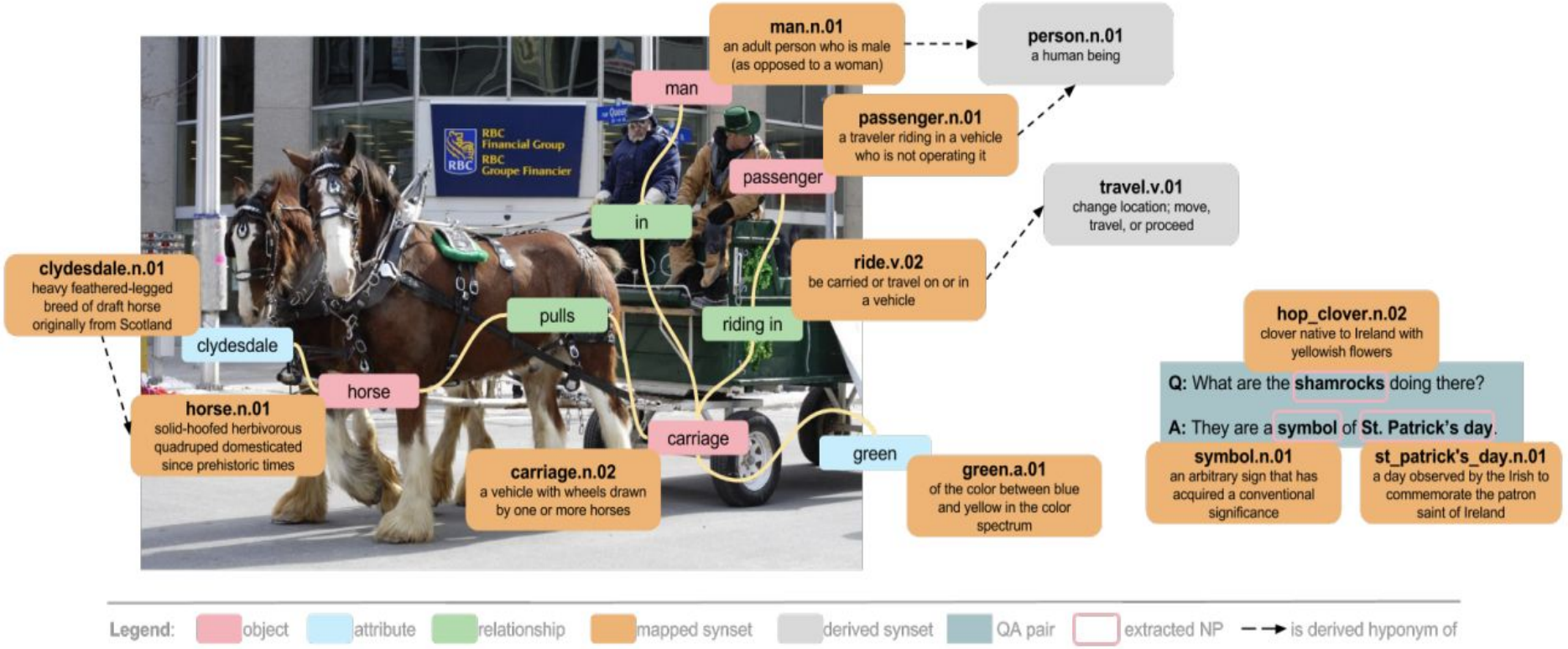


Figure 1: (a) Images are shown to workers at 100ms per image. Workers react whenever they see a dog. (b) The true labels are the ground truth dog images. (c) The workers' keypresses are slow and occur several images after the dog images have already passed. We record these keypresses as the observed labels. (d) Our technique models each keypress as a delayed Gaussian to predict (e) the probability of an image containing a dog from these observed labels.

Core idea: Show images super quickly (100ms) to workers, keypress when they see an object of a given class. Model the delay to predict the images with those objects

Original paper (+ above image): https://arxiv.org/pdf/1602.04506.pdf

Presentation: https://dl.acm.org/citation.cfm?id=2858115


man.n.01
an adult person who is male (as opposed to a woman)
person.n.01
a human being
man
passenger.n.01
a traveler riding in a vehicle who is not operating it
passenger
travel.v.01
change location; move, travel, or proceed
in
ride.v.02
be carried or travel on or in a vehicle
clydesdale.n.01
heavy feathered-legged breed of draft horse originally from Scotland
pulls
riding in
hop_clover.n.02
clover native to Ireland with yellowish flowers
clydesdale
Q: What are the shamrocks doing there?
horse
A: They are a symbol of St. Patrick's day.
horse.n.01
solid-hoofed herbivorous quadruped domesticated since prehistoric times
carriage
green
symbol.n.01
an arbitrary sign that has acquired a conventional significance
st_patrick's_day.n.01
a day observed by the Irish to commemorate the patron saint of Ireland
carriage.n.02
a vehicle with wheels drawn by one or more horses
green.a.01
of the color between blue and yellow in the color spectrum
RBC Financial Group
RBC Groupe Financier
Legend:
object
attribute
relationship
mapped synset
derived synset
QA pair
extracted NP
is derived hyponym of

	Images	Descriptions per Image	Total Objects	# Object Categories	Objects per Image	# Attributes Categories	Attributes per Image	# Relationship Categories	Relationships per Image	Question Answers
YFCC100M (Thomee et al., 2016)	100,000,000	-	-	-	-	-	-	-	-	-
Tiny Images (Torralba et al., 2008)	80,000,000	-	-	53,464	1	-	-	-	-	-
ImageNet (Deng et al., 2009)	14,197,122	-	14,197,122	21,841	1	-	-	-	-	-
ILSVRC Detection (2012) (Russakovsky et al., 2015)	476,688	-	534,309	200	2.5	-	-	-	-	-
MS-COCO (Lin et al., 2014)	328,000	5	27,472	80	-	-	-	-	-	-
Flickr 30K (Young et al., 2014)	31,783	5	-	-	-	-	-	-	-	-
Caltech 101 (Fei-Fei et al., 2007)	9,144	-	9,144	102	1	-	-	-	-	-
Caltech 256 (Griffin et al., 2007)	30,608	-	30,608	257	1	-	-	-	-	-
Caltech Pedestrian (Dollar et al., 2012)	250,000	-	350,000	1	1.4	-	-	-	-	-
Pascal Detection (Everingham et al., 2010)	11,530	-	27,450	20	2.38	-	-	-	-	-
Abstract Scenes (Zitnick and Parikh, 2013)	10,020	-	58	11	5	-	-	-	-	-
aPascal (Farhadi et al., 2009)	12,000	-	-	-	-	64	-	-	-	-
Animal Attributes (Lampert et al., 2009)	30,000	-	-	-	-	1,280	-	-	-	-
SUN Attributes (Patterson et al., 2014)	14,000	-	-	-	-	700	700	-	-	-
Caltech Birds (Wah et al., 2011)	11,788	-	-	-	-	312	312	-	-	-
COCO Actions (Ronchi and Perona, 2015)	10,000	-	-	-	5.2	-	-	156	20.7	-
Visual Phrases (Sadeghi and Farhadi, 2011)	-	-	-	-	-	-	-	17	1	-
VisKE (Sadeghi et al., 2015)	-	-	-	-	-	-	-	6500	-	-
DAQUAR (Malinowski and Fritz, 2014)	1,449	-	-	-	-	-	-	-	-	12,468
COCO QA (Ren et al., 2015a)	123,287	-	-	-	-	-	-	-	-	117,684
Baidu (Gao et al., 2015)	120,360	-	-	-	-	-	-	-	-	250,569
VQA (Antol et al., 2015)	204,721	-	-	-	-	-	-	-	-	614,163
Visual Genome	108,077	50	3,843,636	33,877	35	68,111	26	42,374	21	1,773,258

Table 1: A comparison of existing datasets with Visual Genome. We show that Visual Genome has an order of magnitude more descriptions and question answers. It also has a more diverse set of object, attribute, and relationship classes. Additionally, Visual Genome contains a higher density of these annotations per image. The number of distinct categories in Visual Genome are calculated by lower-casing and stemming names of objects, attributes and relationships.

Table Credit: *Krishna et*

	Images	Descriptions per Image	Total Objects	# Object Categories	Objects per Image	# Attributes Categories	Attributes per Image	# Relationship Categories	Relationships per Image	Question Answers
YFCC100M (Thomee et al., 2016)	100,000,000	-	-	-	-	-	-	-	-	-
Tiny Images (Torralba et al., 2008)	80,000,000	-	-	53,464	1	-	-	-	-	-
ImageNet (Deng et al., 2009)	14,197,122		14,197,122	21,841	1	-	-	-	-	-
ILSVRC Detection (2012) (Russakovsky et al., 2015)	476,688	-	534,309	200	2.5	-	-	-	-	-
MS-COCO (Lin et al., 2014)	328,000	5	27,472	80	-	-	-	-	-	-
Flickr 30K (Young et al., 2014)	31,783	5	-	-	-	-	-	-	-	-
Caltech 101 (Fei-Fei et al., 2007)	9,144	-	9,144	102	1	-	-	-	-	-
Caltech 256 (Griffin et al., 2007)	30,608	-	30,608	257	1	-	-	-	-	-
Caltech Pedestrian (Dollar et al., 2012)	250,000	-	350,000	1	1.4	-	-	-	-	-
Pascal Detection (Everingham et al., 2010)	11,530		27,450	20	2.38	-	-	-	-	-
Abstract Scenes (Zitnick and Parikh, 2013)	10,020	-	58	11	5	-	-	-	-	-
aPascal (Farhadi et al., 2009)	12,000	-	-	-	-	64	-	-	-	-
Animal Attributes (Lampert et al., 2009)	30,000	-	-	-	-	1,280	-	-	-	-
SUN Attributes (Patterson et al., 2014)	14,000	-	-	-	-	700	700	-	-	-
Caltech Birds (Wah et al., 2011)	11,788	-	-	-	-	312	312	-	-	-
COCO Actions (Ronchi and Perona, 2015)	10,000	-	-	-	5.2		-	156	20.7	-
Visual Phrases (Sadeghi and Farhadi, 2011)	-	-	-	-	-	-	-	17	1	-
VisKE (Sadeghi et al., 2015)	-	-	-	-	-	-	-	6500	-	-
DAQUAR (Malinowski and Fritz, 2014)	1,449	-	-	-	-	-	-	-	-	12,468
COCO QA (Ren et al., 2015a)	123,287	-	-	-	-	-	-	-	-	117,684
Baidu (Gao et al., 2015)	120,360	-	-	-	-	-	-	-	-	250,569
VQA (Antol et al., 2015)	204,721	-	-	-	-	-	-	-	-	614,163
Visual Genome	108,077	50	3,843,636	33,877	35	68,111	26	42,374	21	1,773,258

Table 1: A comparison of existing datasets with Visual Genome. We show that Visual Genome has an order of magnitude more descriptions and question answers. It also has a more diverse set of object, attribute, and relationship classes. Additionally, Visual Genome contains a higher density of these annotations per image. The number of distinct categories in Visual Genome are calculated by lower-casing and stemming names of objects, attributes and relationships.

Table Credit: *Krishna et*

	Images	Descriptions per Image	Total Objects	# Object Categories	Objects per Image	# Attributes Categories	Attributes per Image	# Relationship Categories	Relationships per Image	Question Answers
YFCC100M (Thomee et al., 2016)	100,000,000	-	-	-	-					
Tiny Images (Torralba et al., 2008)	80,000,000	-	-	53,464	1					
ImageNet (Deng et al., 2009)	14,197,122	-	14,197,122	21,841	1					
ILSVRC Detection (2012) (Russakovsky et al., 2015)	476,688	-	534,309	200	2.5					
MS-COCO (Lin et al., 2014)	328,000	5	27,472	80	-					
Flickr 30K (Young et al., 2014)	31,783	5	-	-	-					
Caltech 101 (Fei-Fei et al., 2007)	9,144	-	9,144	102	1					
Caltech 256 (Griffin et al., 2007)	30,608	-	30,608	257	1					
Caltech Pedestrian (Dollar et al., 2012)	250,000	-	350,000	1	1.4					
Pascal Detection (Everingham et al., 2010)	11,530	-	27,450	20	2.38					
Abstract Scenes (Zitnick and Parikh, 2013)	10,020	-	58	11	5					
aPascal (Farhadi et al., 2009)	12,000	-	-	-	-					
Animal Attributes (Lampert et al., 2009)	30,000	-	-	-	-	1,280	-	-	-	-
SUN Attributes (Patterson et al., 2014)	14,000	-	-	-	-	700	700	-	-	-
Caltech Birds (Wah et al., 2011)	11,788	-	-	-	-	312	312	-	-	-
COCO Actions (Ronchi and Perona, 2015)	10,000	-	-	-	5.2	-	-	156	20.7	-
Visual Phrases (Sadeghi and Farhadi, 2011)	-	-	-	-	-	-	-	17	1	-
VisKE (Sadeghi et al., 2015)	-	-	-	-	-	-	-	6500	-	-
DAQUAR (Malinowski and Fritz, 2014)	1,449	-	-	-	-	-	-	-	-	12,468
COCO QA (Ren et al., 2015a)	123,287	-	-	-	-	-	-	-	-	117,684
Baidu (Gao et al., 2015)	120,360	-	-	-	-	-	-	-	-	250,569
VQA (Antol et al., 2015)	204,721	-	-	-	-	-	-	-	-	614,163
Visual Genome	108,077	50	3,843,636	33,877	35	68,111	26	42,374	21	1,773,258

Visual Genome vs. MS-COCO:
Both based on real-world images, with segmentation and descriptions

- 108k vs. 300k photos
  - Subset of COCO!
- 34k object classes vs. 80 object classes
- >50 regional descriptions vs. 5 sentences about each image

Table 1: A comparison of existing datasets with Visual Genome. We show that Visual Genome has an order of magnitude more descriptions and question answers. It also has a more diverse set of object, attribute, and relationship classes. Additionally, Visual Genome contains a higher density of these annotations per image. The number of distinct categories in Visual Genome are calculated by lower-casing and stemming names of objects, attributes and relationships.

Table Credit: *Krishna et al.*

	Images	Descriptions per Image	Total Objects	# Object Categories	Objects per Image	# Attributes Categories	Attributes per Image	# Relationship Categories	Relationships per Image	Question Answers
YFCC100M (Thomee et al., 2016)	100,000,000	-	-	-	-					
Tiny Images (Torralba et al., 2008)	80,000,000	-	-	53,464	1					
ImageNet (Deng et al., 2009)	14,197,122	-	14,197,122	21,841	1					
ILSVRC Detection (2012) (Russakovsky et al., 2015)	476,688	-	534,309	200	2.5					
MS-COCO (Lin et al., 2014)	328,000	5	27,472	80	-					
Flickr 30K (Young et al., 2014)	31,783	5	-	-	-					
Caltech 101 (Fei-Fei et al., 2007)	9,144	-	9,144	102	1					
Caltech 256 (Griffin et al., 2007)	30,608	-	30,608	257	1					
Caltech Pedestrian (Dollar et al., 2012)	250,000	-	350,000	1	1.4					
Pascal Detection (Everingham et al., 2010)	11,530	-	27,450	20	2.38					
Abstract Scenes (Zitnick and Parikh, 2013)	10,020	-	58	11	5					
aPascal (Farhadi et al., 2009)	12,000	-	-	-	-					
Animal Attributes (Lampert et al., 2009)	30,000	-	-	-	-	1,280	-	-	-	-
SUN Attributes (Patterson et al., 2014)	14,000	-	-	-	-	700	700	-	-	-
Caltech Birds (Wah et al., 2011)	11,788	-	-	-	-	312	312	-	-	-
COCO Actions (Ronchi and Perona, 2015)	10,000	-	-	-	5.2	-	-	156	20.7	-
Visual Phrases (Sadeghi and Farhadi, 2011)	-	-	-	-	-	-	-	17	1	-
VisKE (Sadeghi et al., 2015)	-	-	-	-	-	-	-	6500	-	-
DAQUAR (Malinowski and Fritz, 2014)	1,449	-	-	-	-	-	-	-	-	12,468
COCO QA (Ren et al., 2015a)	123,287	-	-	-	-	-	-	-	-	117,684
Baidu (Gao et al., 2015)	120,360	-	-	-	-	-	-	-	-	250,569
VQA (Antol et al., 2015)	204,721	-	-	-	-	-	-	-	-	614,163
Visual Genome	108,077	50	3,843,636	33,877	35	68,111	26	42,374	21	1,773,258

Table 1: A comparison of existing datasets with Visual Genome. We show that Visual Genome has an order of magnitude more descriptions and question answers. It also has a more diverse set of object, attribute, and relationship classes. Additionally, Visual Genome contains a higher density of these annotations per image. The number of distinct categories in Visual Genome are calculated by lower-casing and stemming names of objects, attributes and relationships.

Visual Genome vs. VQA:

Both include open ended question-answer pairs on real-world images. Multiple pairs per image.

- 1.8M vs. 614k Q-A pairs
- 57% vs. 89% of answers are single-word
  - 39% of VQA answers are yes/no!

Table Credit: *Krishna et*

Key Metrics

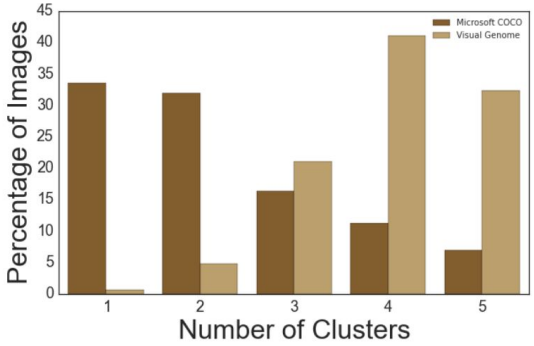
More Dense:

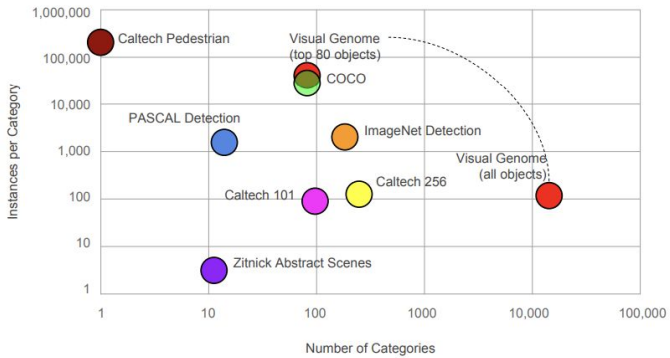
- 35 objects per image (OM+)
- 144k unique objects, relationships, and attributes (OM+)
- 1.4M Q-A pairs (more than any other)

Key Metrics

More Comprehensive/Diverse:

- More object categories (34k total)
- Object-specific attributes: size, pose, emotion, etc.
- More semantically diverse captions, but still imperfect (2x more men annotated than women)





Semantic Diversity Detection

1. Use word2vec to convert each word to a 300-dimensional vector
2. Hierarchical agglomerative clustering on vector representations -> 71 clusters
3. 5 descriptions randomly chosen per image

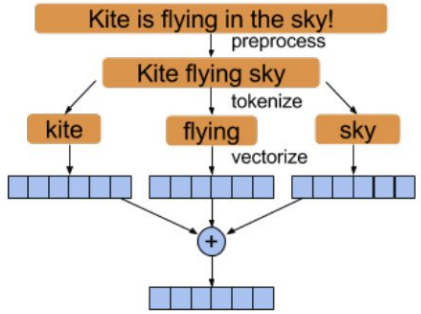
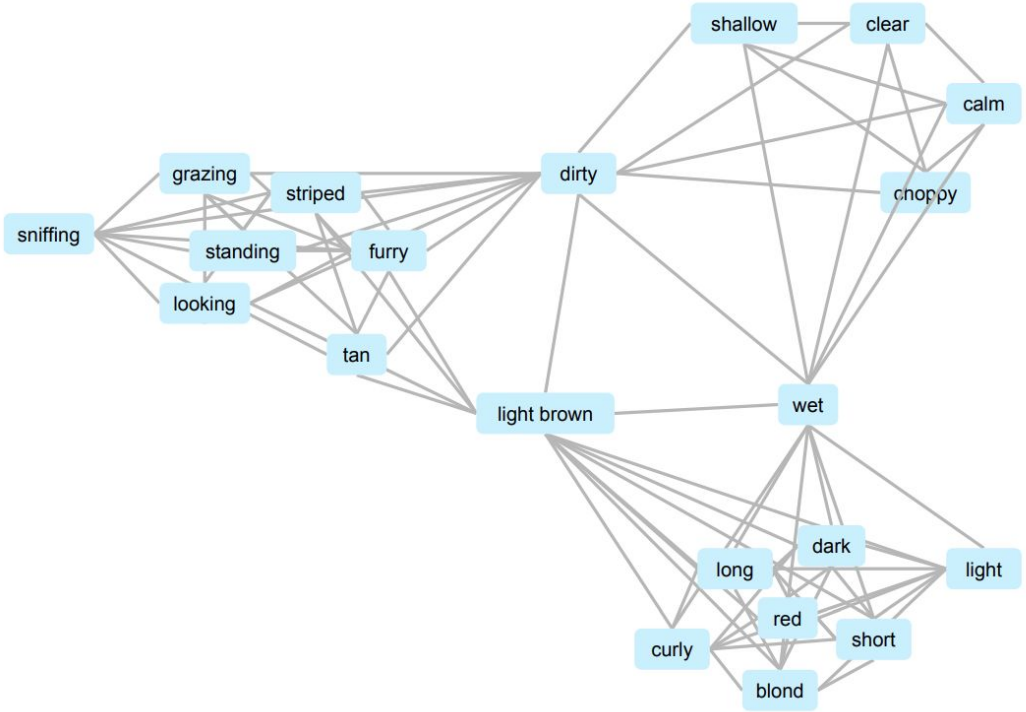


Fig. 17: The process used to convert a region description into a 300-dimensional vectorized representation.

Findings: **Attribute Graphs**



Findings: **Affordances**

Using typical relationships, can learn common sense knowledge like couches have pillows on them, zebras eat hay, etc.

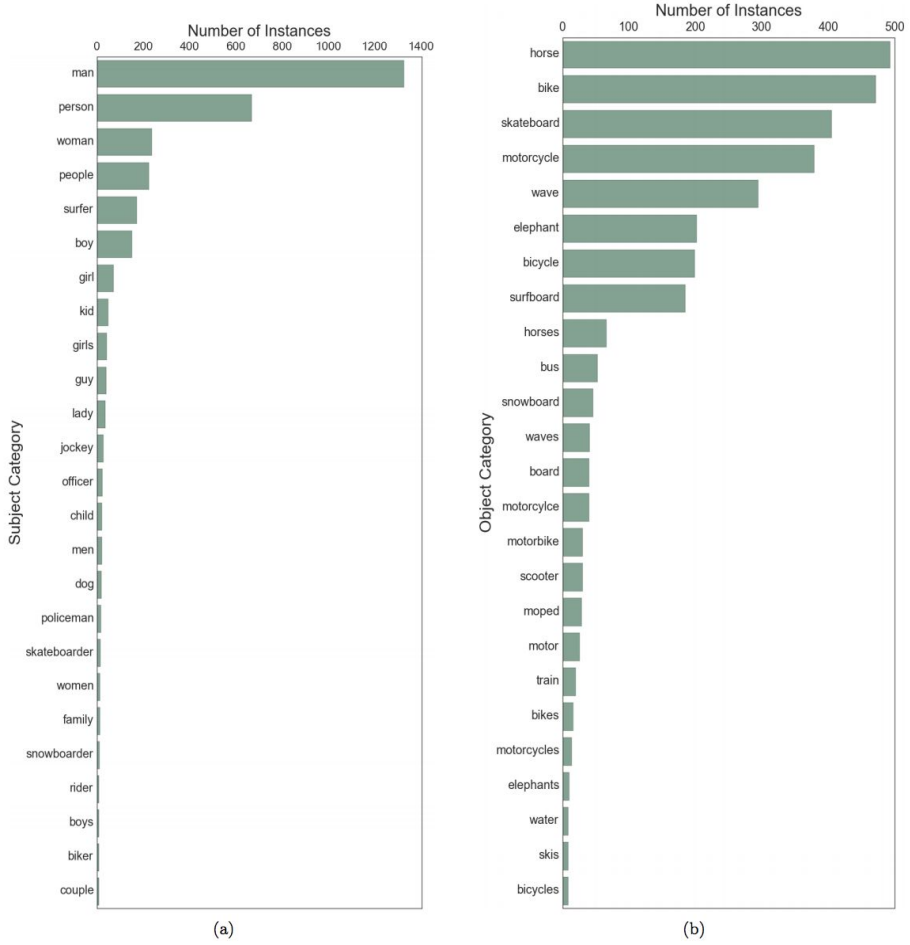


Fig. 28: (a) Distribution of subjects for the relationship `riding`. (b) Distribution of objects for the relationship `riding`. Subjects comprise of people-like entities like `person`, `man`, `policeman`, `boy`, and `skateboarder` that can ride other objects. On the other hand, objects like `horse`, `bike`, `elephant` and `motorcycle` are entities that can afford `riding`.

Question: Have any researchers taken advantage of these kinds of affordance relationships?

Visual Relationship Detection with Language Priors (link)

By: Lu et al., 2016

3.

SCENE GRAPH APPLICATIONS

1. IR: Key Challenge
2. IR: Dataset
3. IR: Implementation
4. IR: Experiments/Results
5. MP: Key Challenge
6. MP: Implementation
7. MP: Experiments/Results

Image Retrieval using Scene Graphs

By: Johnson et al., 2015
Note: Johnson also collaborated on VG, and will have his faculty interview here on March 29th!

For the following slides: Images Credit: *Johnson et al.*

Key Challenge:
Using Scene Graphs for Image Retrieval

Intuition: Scene graphs can represent what you actually want better than plaintext

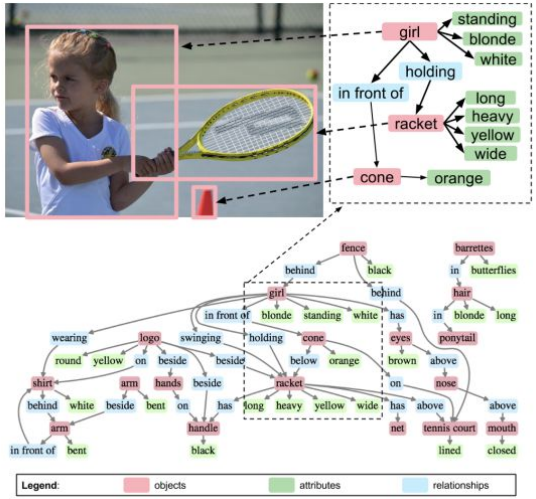
Problem: Given a scene graph query, identify the image that best matches it

Evaluation:

1. Performance on hyper-precise graphs
2. Performance on more simple, open-ended graphs
3. Performance generating accurate object localizations

(a) Results for the query on a popular image search engine.

(b) Expected results for the query.



Scene Graph Formalization

Scene graphs include objects, attributes, and relationships, and are *grounded* to an image.

C: Set of object classes

A: Set of attribute types

R: Set of relationships

$o_i = (c_i, A_i)$, one object

$O = \{o_1, \ldots, o_n\}$, all objects

$E \subseteq O \times \mathcal{R} \times O$, set of edges

Scene Graph G = (O, E)

B: Set of bounding boxes

Grounding: $\gamma : O \rightarrow B$

γ_o: grounding of object o to bounding box b

Dataset

- 5k images, intersection of MS-COCO + YFCC100m
- Uses AMT workers to write object, attributes, and relationships with an open vocabulary
- Uses AMT workers to draw bounding boxes
- Uses AMT workers to verify all attributions

For experiments, Johnson et al. discarded object + attribute classes with < 50 occurrences and relationships with < 30 occurrences

Note: This paper was written prior to Visual Genome's release

Dataset Details

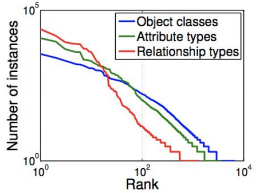


Figure 4: Objects, attributes and relations reveal a Zipf distribution when ordered by number of labeled instances.

	Full dataset	Experiments Sect. 6	COCO 2014 [42]	ILSVRC 2014 (Det) [54]	Pascal VOC [15]
Object classes	**6,745**	**266**	80	200	20
Attribute types	3,743	145	-	-	-
Relationship types	1,310	68	-	-	-
Object instances	93,832	69,009	**886,284**	534,309	27,450
Attribute instances	110,021	94,511	-	-	-
Relationship instances	112,707	109,535	-	-	-
Instances per obj. class	13.9	259.4	**11,087.5**	2,672.5	1,372
Instances per attr. type	29.4	651.8	-	-	-
Instances per rel. type	86.0	1,610.8	-	-	-
Objects per image	**18.8**	**13.8**	7.2	1.1	2.4
Attributes per image	22.0	18.9	-	-	-
Relationships per image	22.5	21.9	-	-	-
Attributes per object	1.2	1.0	-	-	-
Relationships per object	2.4	2.3	-	-	-

Table 1: Aggregate statistics for our *real-world scene graphs* dataset, for the full dataset and the restricted sets of object, attribute, and relationship types used in experiments.

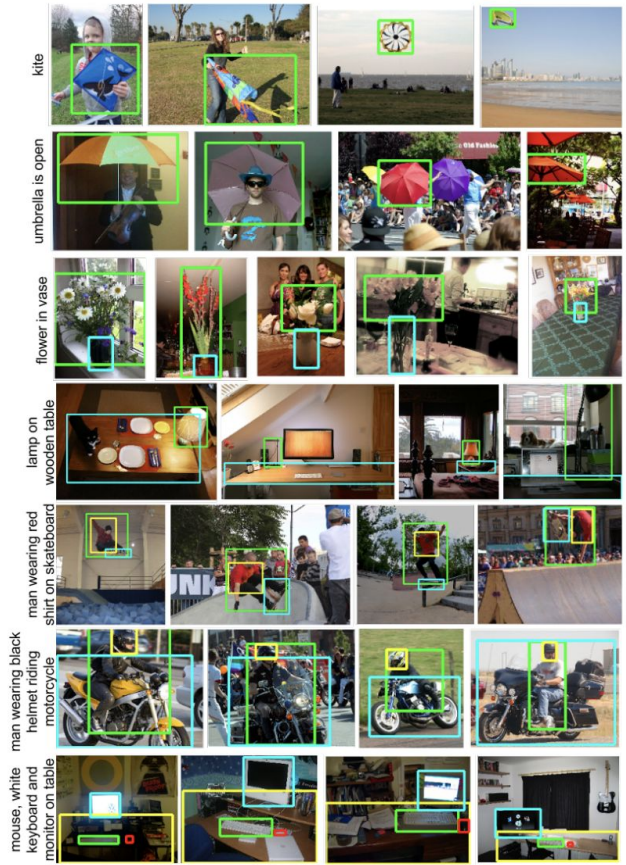


Figure 3: Examples of scene sub-graphs of increasing complexity (top to bottom) from our dataset, with attributes and up to 4 different objects.

Dataset Findings

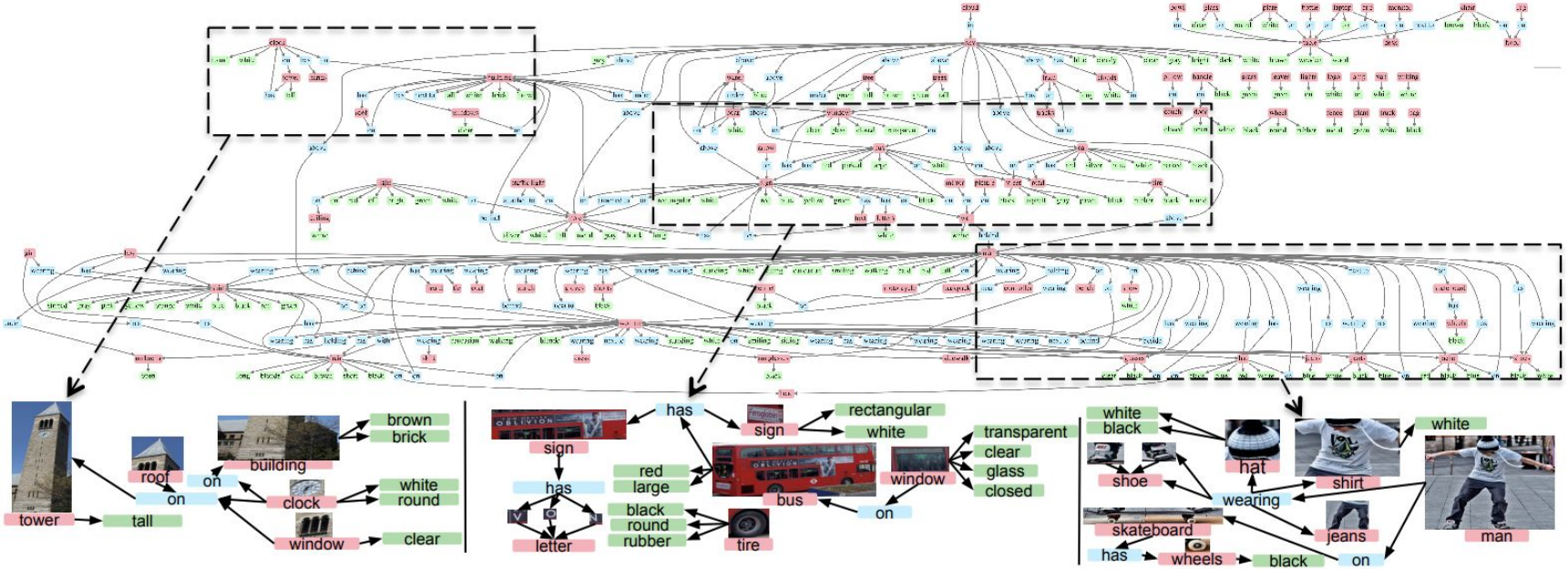


Figure 5: An *aggregate* scene graph computed using our entire dataset. We visualize the 150 most frequently occurring (object, relationship, object) and (object, attribute) tuples. We also provide 3 examples of scene graphs grounded in images that contribute to the sub-graphs within the dashed rectangles of the aggregated graph. Best viewed with magnification.

CRF Formulation

Task: Given a scene graph, want to retrieve images

Solution: For a given graph, measure ‘agreement’ between it and all unannotated images

- Use a Conditional Random Field (CRF) to model distribution over all possible groundings
- Use Maximum a Posteriori (MAP) inference to find most likely grounding
- Use the likelihood of this ‘best’ grounding as a measure of agreement

CRF Formulation

$$P(\gamma \mid G, B) = \prod_{o \in O} P(\gamma_o \mid o) \prod_{(o,r,o') \in E} P(\gamma_o, \gamma_{o'} \mid o, r, o').$$

Using Bayes rule + since $P(y_o)$ and $P(o)$ are constants for MAP inference:

$$\gamma^* = \arg\max_{\gamma} \prod_{o \in O} P(o \mid \gamma_o) \prod_{(o,r,o') \in E} P(\gamma_o, \gamma_{o'} \mid o, r, o').$$

CRF Formulation

1: $$P(o \mid \gamma_o) = P(c \mid \gamma_o) \prod_{a \in A} P(a \mid \gamma_o).$$

$P(c \mid \gamma_o)$ and $P(a \mid \gamma_o)$. are probabilities that box y_o has object o or attribute a.

1. Use R-CNN to train detectors for 266 object classes and 145 attribute types
2. Obtain SVM classification scores
3. Use Platt scaling to convert this to probabilities

CRF Formulation

2: $P(\gamma_o, \gamma_{o'} \mid o, r, o')$

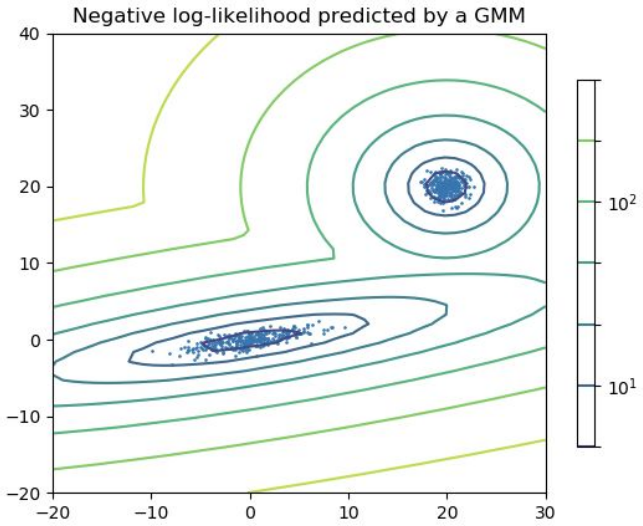
Train a Gaussian Mixture Model (GMM) to model:
$P(f(\gamma_o, \gamma_{o'}) \mid c, r, c')$ and $P(f(\gamma_o, \gamma_{o'}) \mid r)$

(use the latter if <30 instances of (c, r, c')).

Use Platt scaling to convert GMM output to probabilities

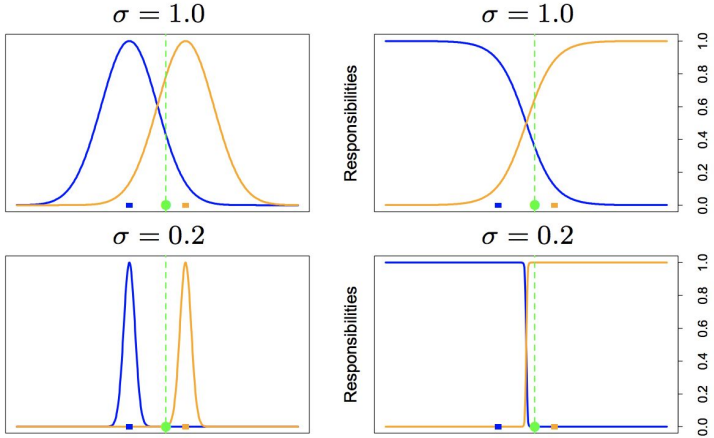
$$f(\gamma_o, \gamma_{o'}) = \left((x - x')/w,\ (y - y')/h,\ w'/w,\ h'/h\right)$$

Background: Gaussian Mixture Models



Mixture Model: $f(x) = (1-\pi)g_1(x) + \pi g_2(x)$

Gaussian mixture: $g_j(x) = \phi_{\theta_j}(x),\ \theta_j = (\mu_j, \sigma_j^2)$



Model data points into a number of Gaussian distributions with unknown parameters

Fantastic overview:
https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

Image (left): http://scikit-learn.org/stable/modules/mixture.html

Image (right): http://statweb.stanford.edu/~tibs/stat315a/LECTURES/em.pdf

Implementation

Training: Learn from a set of images with associated grounded scene graphs

Testing: given a scene graph + unannotated images,

1. For each image,
    a. Generate candidate boxes using Geodesic Object Proposals (GOP)
    b. Use CRF + MAP to identify best grounding and output probability of match
2. Return ranked list of images by probability

Background: Geodesic Object Proposals

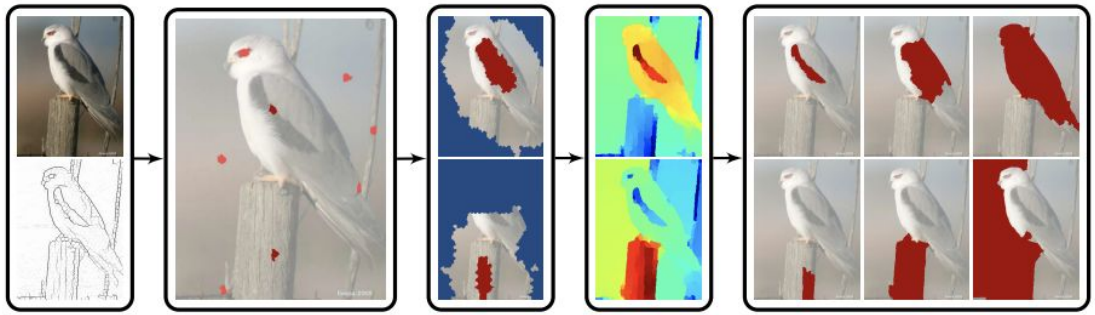


Fig. 2: Overall proposal generation pipeline. (a) Input image with a computed superpixel segmentation and a boundary probability map. (b) Seeds placed by the presented approach. (c) Foreground and background masks generated by the presented approach for two of these seeds. (d) Signed geodesic distance transforms for these masks. (e) Object proposals, computed by identifying critical level sets in each SGDT.

Creates a probability map for boundaries, then places geodesic seeds. Uses those seeds to make maps, then uses a geodesic distance transform for final object proposals.

Note: The paper found "[Selective Search (SS)] achieves the highest object recall on our dataset; however we use [Geodesic Object Proposals] GOP for all experiments as it provides the best trade-off between object recall (≈ 70% vs ≈ 80% for SS) and number of regions per image (632 vs 1720 for SS).

Original paper (+ above image):
http://www.philkr.net/papers/2014-10-01-eccv/2014-10-01-eccv.pdf

Evaluation

Models Used:

- **SG-obj-attr-rel:** Our model. Includes unary object and attribute potentials and binary relationship potentials.
- **SG-obj-attr:** Our model, using only object and attribute potentials.
- **SG-obj:** Our model, using only object potentials. Equivalent to R-CNN
- **Rand:** Random permutation

Metrics Used:

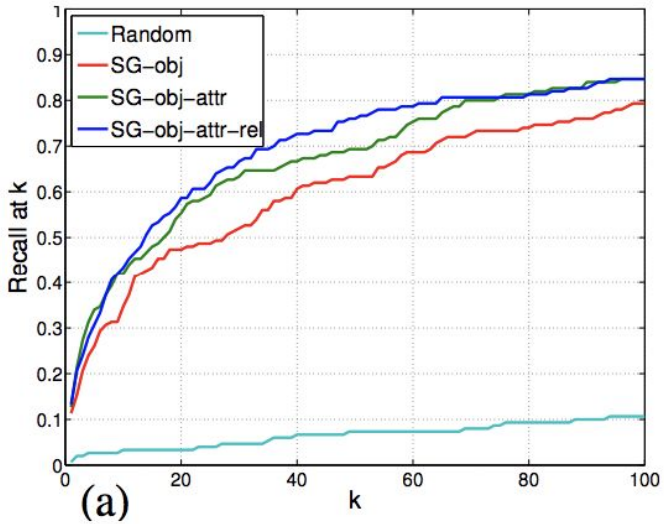
- **Med r:** Median rank for true image/highest true image
- **R @ N:** Recall at rank N

Results

1: Full scene-graph queries

Pick an image from the test set. Query test set with its scene graph. Record rank of the true image.

		Rand	SIFT [43]	GIST [48]	CNN [34]	SG-obj [24]	SG-obj-attr	SG-obj-attr-rel
(a)	Med r	420	-	-	-	28	17.5	**14**
	R@1	0	-	-	-	0.113	0.127	**0.133**
	R@5	0.007	-	-	-	0.260	**0.340**	0.307
	R@10	0.027	-	-	-	0.347	0.420	**0.433**

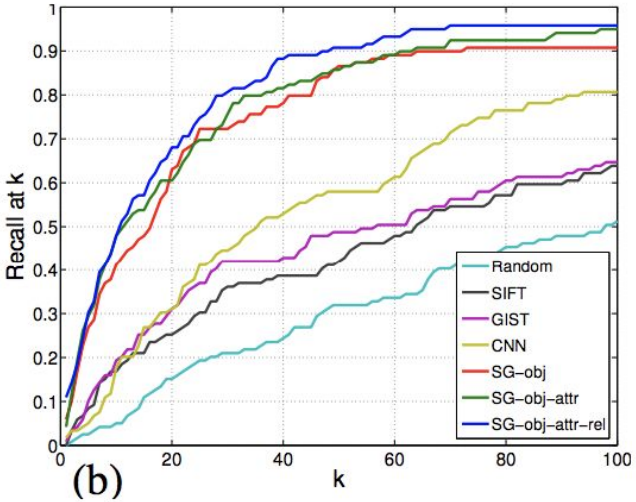


Results

2: Partial scene-graph queries

Mine dataset for recurring (>5x) scene subgraphs.
For each subgraph, find all images that match it.
Query test set and record highest-ranked TP image

		Rand	SIFT [43]	GIST [48]	CNN [34]	SG-obj [24]	SG-obj-attr	SG-obj-attr-rel
(b)	Med r	94	64	57	36	17	12	**11**
	R@1	0	0	0.008	0.017	0.059	0.042	**0.109**
	R@5	0.034	0.084	0.101	0.050	0.269	0.294	**0.303**
	R@10	0.042	0.168	0.193	0.176	0.412	**0.479**	**0.479**

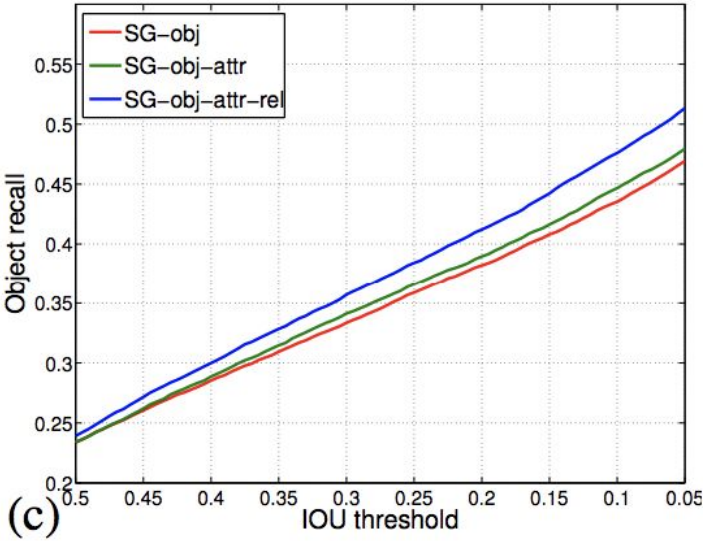


Results

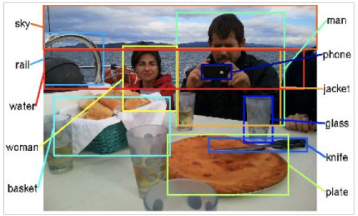
3: Partial scene-graph queries

Evaluate median IoU across all objects in all test images and fraction of objects with IoUs above thresholds. *Note: they're quite low!*

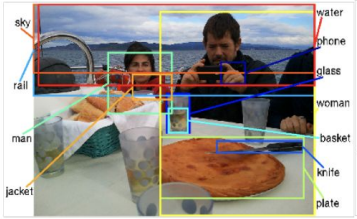
		Rand	SIFT [43]	GIST [48]	CNN [34]	SG-obj [24]	SG-obj-attr	SG-obj-attr-rel
(c)	Med IoU	-	-	-	-	0.014	0.026	**0.067**
	R@0.1	-	-	-	-	0.435	0.447	**0.476**
	R@0.3	-	-	-	-	0.334	0.341	**0.357**
	R@0.5	-	-	-	-	0.234	0.234	**0.239**



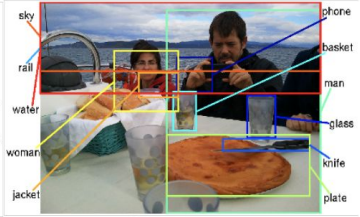
Results



Ground Truth

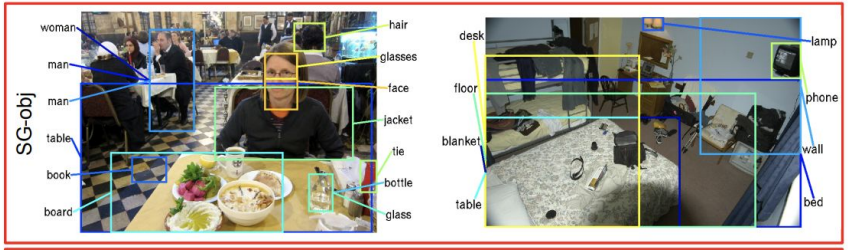


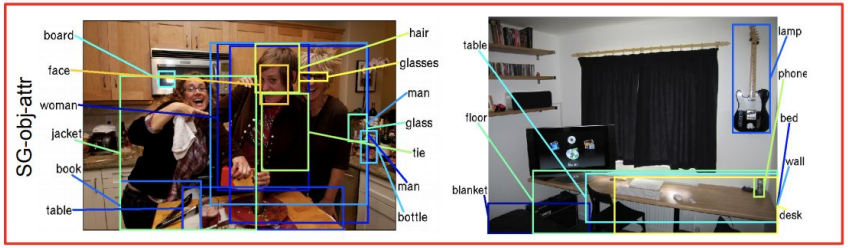
SG-obj

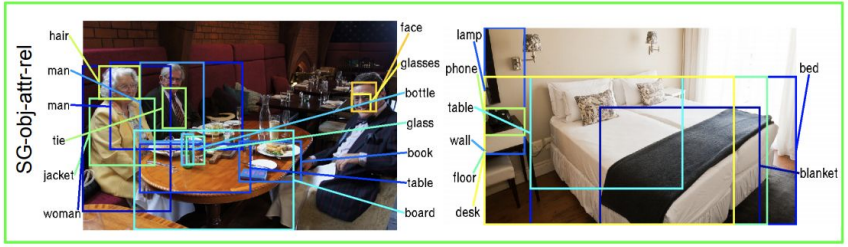


SG-obj-attr-rel

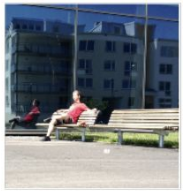
(c)







Left: experiment 1. Top: experiment 3. Bottom: experiment 2.



sitting man on bench

Question: Scene graphs are clunky and complex, and won't likely be used for real-world image retrieval. What do we do instead?

Generating Semantically Precise Scene Graphs from Textual Descriptions for Improved Image Retrieval (link)

By: Schuster et al., 2015

Scene Graph Generation by Iterative Message Passing

By: Xu et al., 2017

For the following slides: Images Credit: *Xu et al.*

Key Challenge:
Generating Scene Graphs from Images

Problem:

Create an end-to-end trainable model that, given an image, outputs a scene graph with object classes, bounding boxes, and relationships

Central Intuition:

Use the **surrounding context** for reasoning; why not use object predictions to predict relationships (and vice versa)?

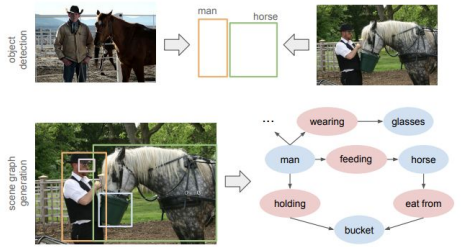


Figure 1. Object detectors perceive a scene by attending to individual objects. As a result, even a perfect detector would produce similar outputs on two semantically distinct images (first row). We propose a scene graph generation model that takes an image as input, and generates a visually-grounded scene graph (second row, right) that captures the objects in the image (blue nodes) and their pairwise relationships (red nodes).

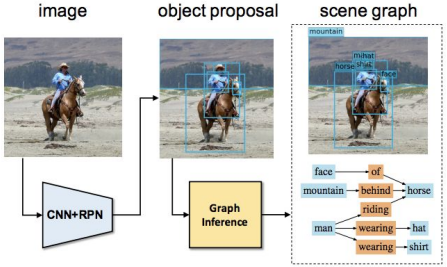
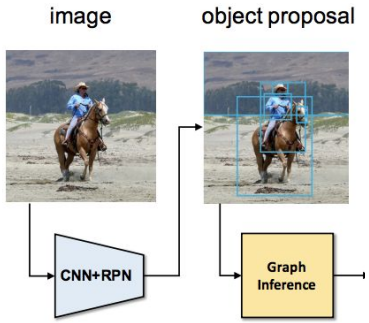


Figure 2. An overview of our model architecture. Given an image as input, the model first produces a set of object proposals using a Region Proposal Network (RPN) [32], and then passes the extracted features of the object regions to our novel graph inference module. The output of the model is a *scene graph* [18], which contains a set of localized objects, categories of each object, and relationship types between each pair of objects.

Scene Graph Generation

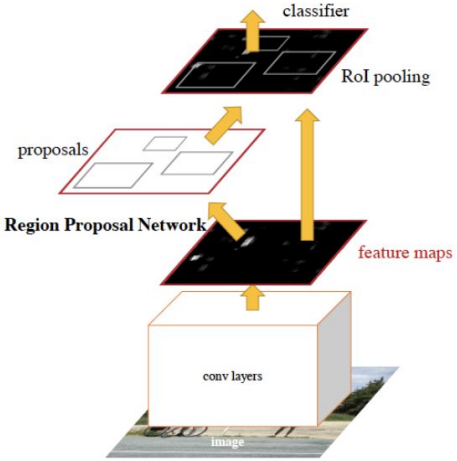
Given an image I:

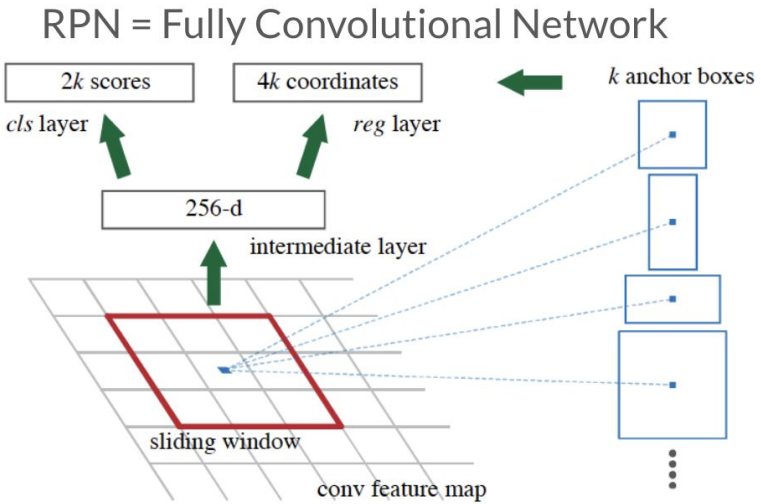
1. Use a Region Proposal Network to generate a set of proposed bounding boxes B_I
2. **Infer relevant labels:**
   a. For each bbox, an object class
   b. For each bbox, offsets for refining position
   c. For each pair, a relationship variable



(Part 2 is the central undertaking of this paper…)

Background: Region Proposal Network





Slides across the conv feature map of an image of any size, feeds features into a box-regression and box-classification layer, outputs set of object proposals

Above image: Berthy/Riley's slides

From Faster-RCNN: https://arxiv.org/pdf/1506.01497.pdf

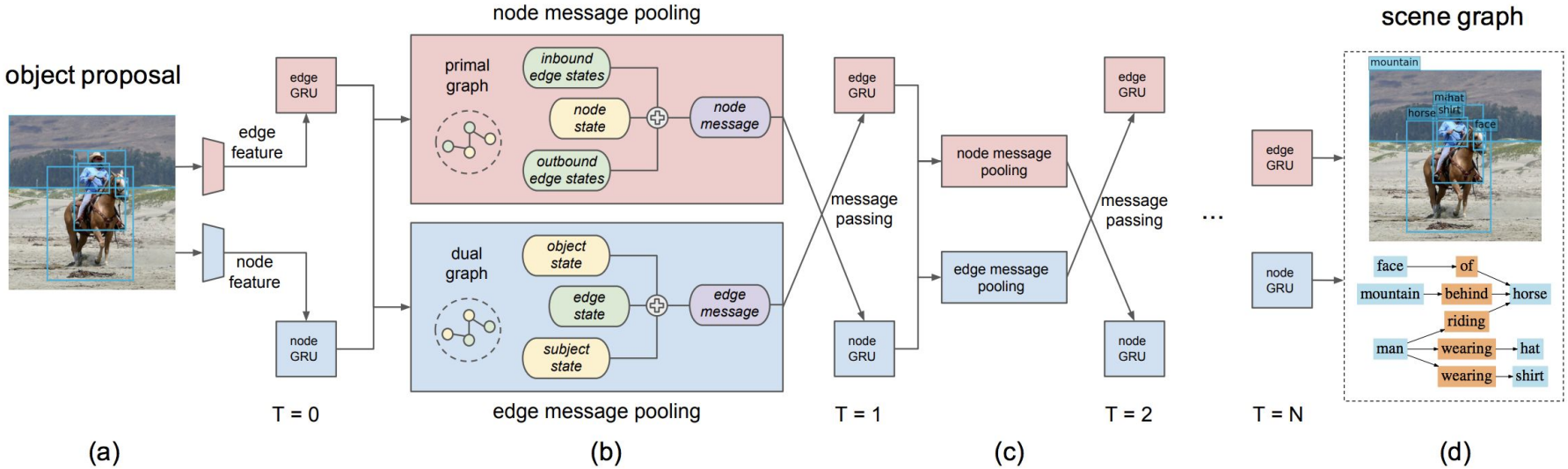
Task Formalization

$$\mathbf{x} = \{x_i^{cls}, x_i^{bbox}, x_{i \to j} | i = 1 \ldots n, j = 1 \ldots n, i \neq j\}$$

$$\Pr(\mathbf{x}|I, B_I) = \prod_{i \in V} \prod_{j \neq i} \Pr(x_i^{cls}, x_i^{bbox}, x_{i \to j} | I, B_I).$$

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \Pr(\mathbf{x}|I, B_I)$$

Notation: Q(x | .) is the probability of x; depends only on the current states of all nodes + edges

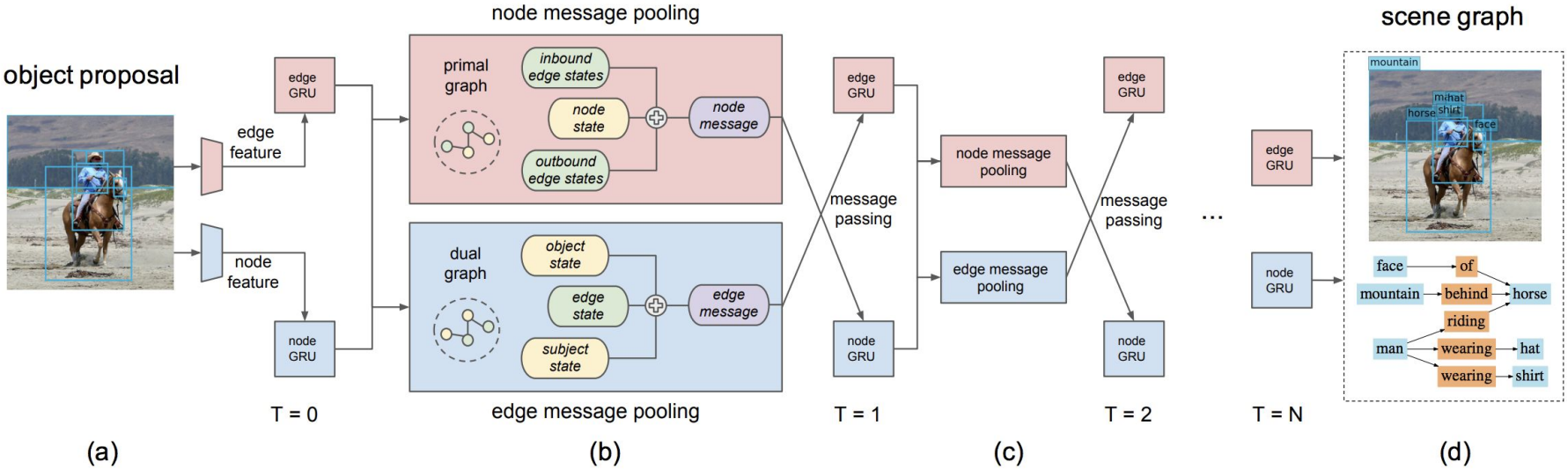


(a): Each node, edge has a corresponding Gated Recurrent Unit (GRU)

Each of these units have hidden states h_i (node) or $h_{i \to j}$ (edge)

Using ROI-Pooling, we extract visual features for each bbox -

f_i^v is the feature for $bbox_i$, while f_i^e is for the union of $boxes_{i,j}$



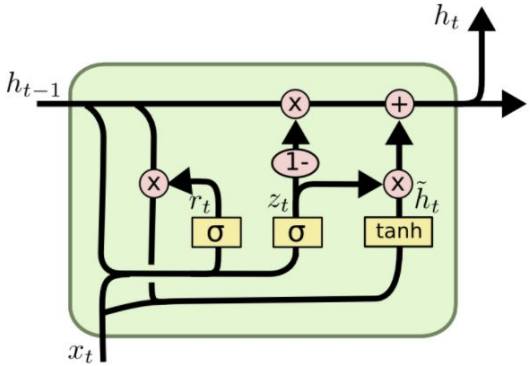
(a): Then, the mean field distribution is initially just:

$$Q(\mathbf{x}|I, B_I) = \prod_{i=1}^{n} Q(x_i^{cls}, x_i^{bbox}|h_i)Q(h_i|f_i^v)$$

$$\prod_{j \neq i} Q(x_{i \to j}|h_{i \to j})Q(h_{i \to j}|f_{i \to j}^e)$$

Background: Gated Recurrent Units

A slightly more dramatic variation on the LSTM is the Gated Recurrent Unit, or GRU, introduced by Cho, et al. (2014). It combines the forget and input gates into a single "update gate." It also merges the cell state and hidden state, and makes some other changes. The resulting model is simpler than standard LSTM models, and has been growing increasingly popular.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

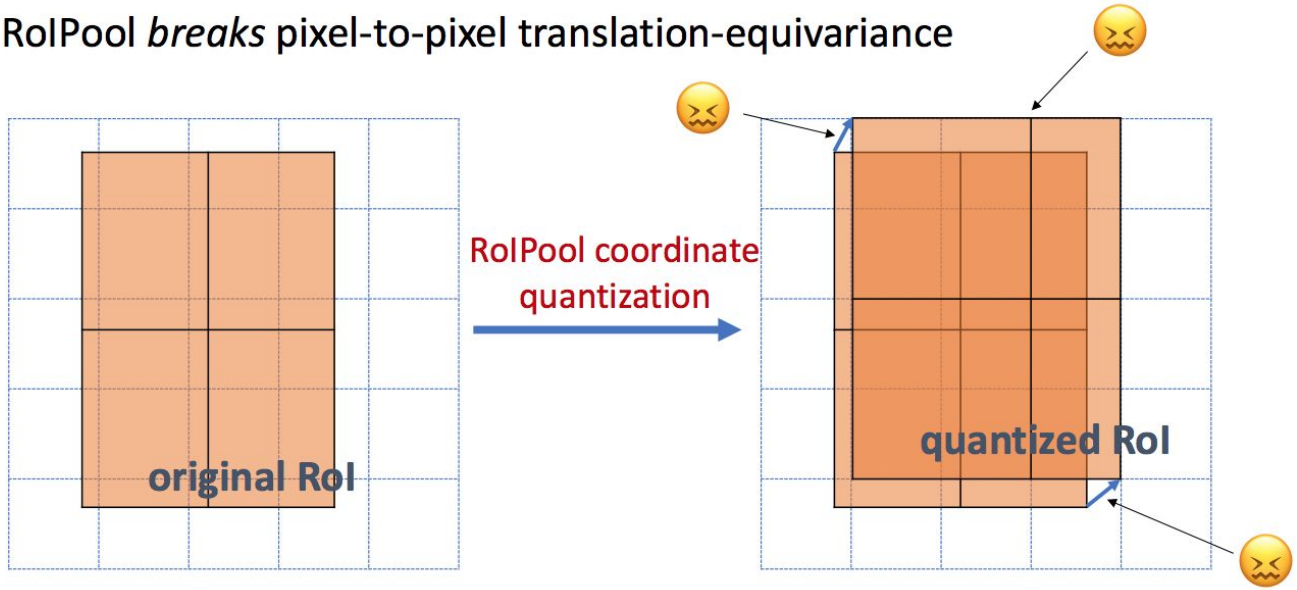
Above image: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Original paper: https://arxiv.org/pdf/1406.1078.pdf

Comparison to LSTMs: https://arxiv.org/pdf/1412.3555v1.pdf

Background: ROI-Pooling

- RoIPool *breaks* pixel-to-pixel translation-equivariance

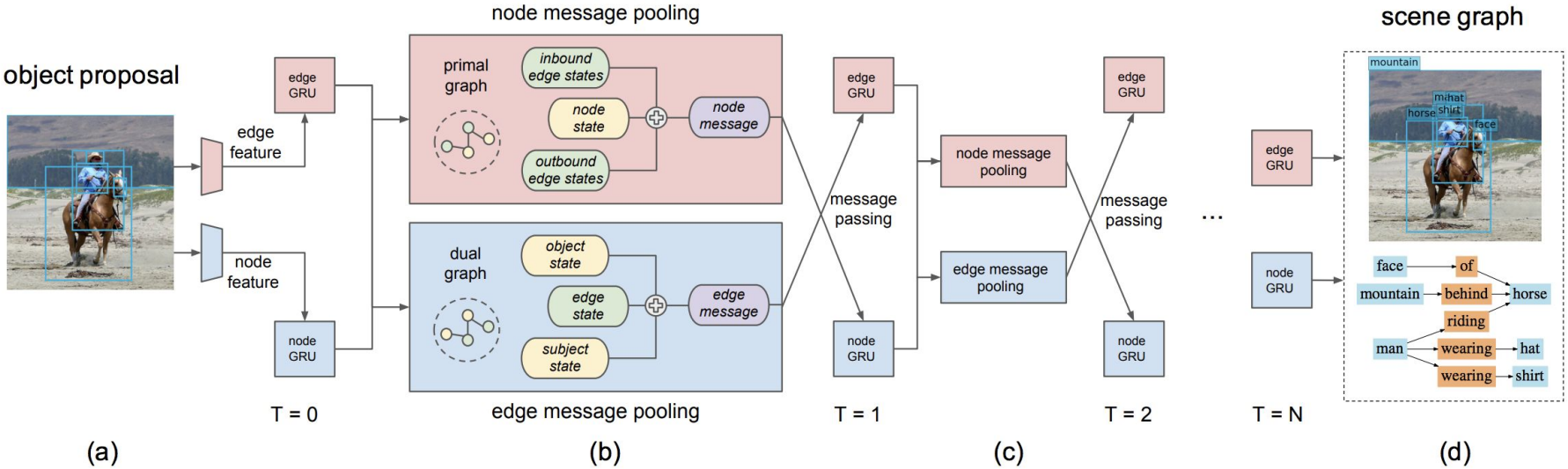


ROI-Pooling: method to efficiently max-pool across inputs of different sizes

Above Image: Berthy/Riley's Slides

Great overview: https://blog.deepsense.ai/region-of-interest-pooling-explained/

From Fast-RCNN: https://arxiv.org/pdf/1504.08083.pdf

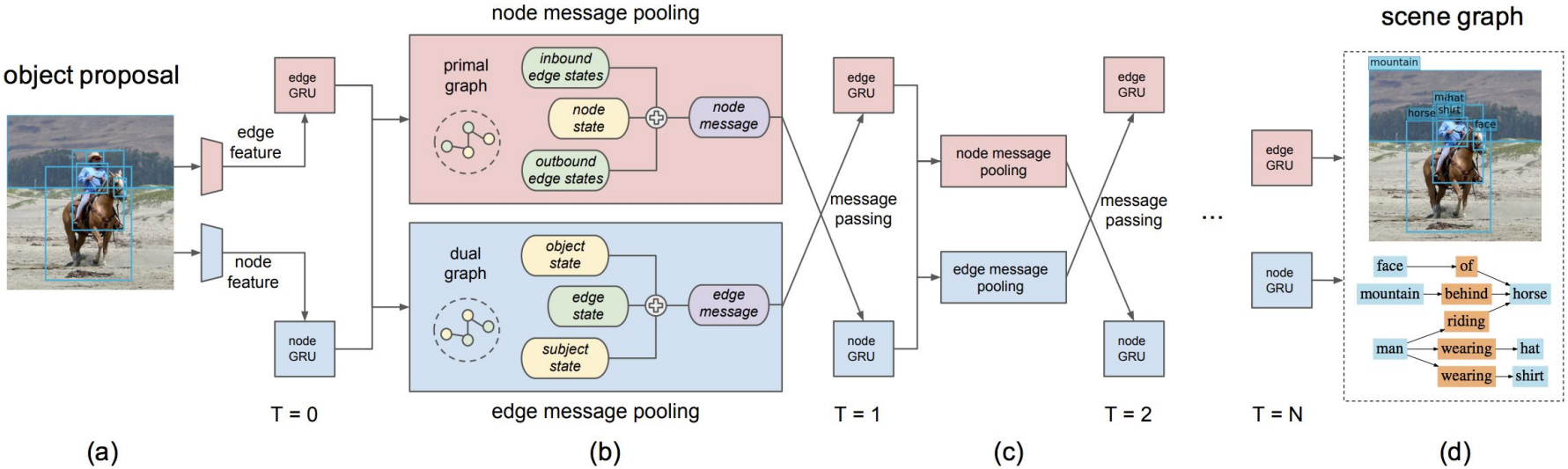


(b): (a) alone is sufficient as an RNN for graph inference. Now, we incorporate message passing for *contextual understanding*.

Bipartite graph: nodes and edges affect each other inter-class-wise

m_i for i^{th} node: h_i, all outbound/inbound edges $h_{i \to j}$ and $h_{j \to i}$

$m_{i \to j}$ for $i \to j^{th}$ edge: $h_{i \to j}$, endpoint nodes h_i and h_j

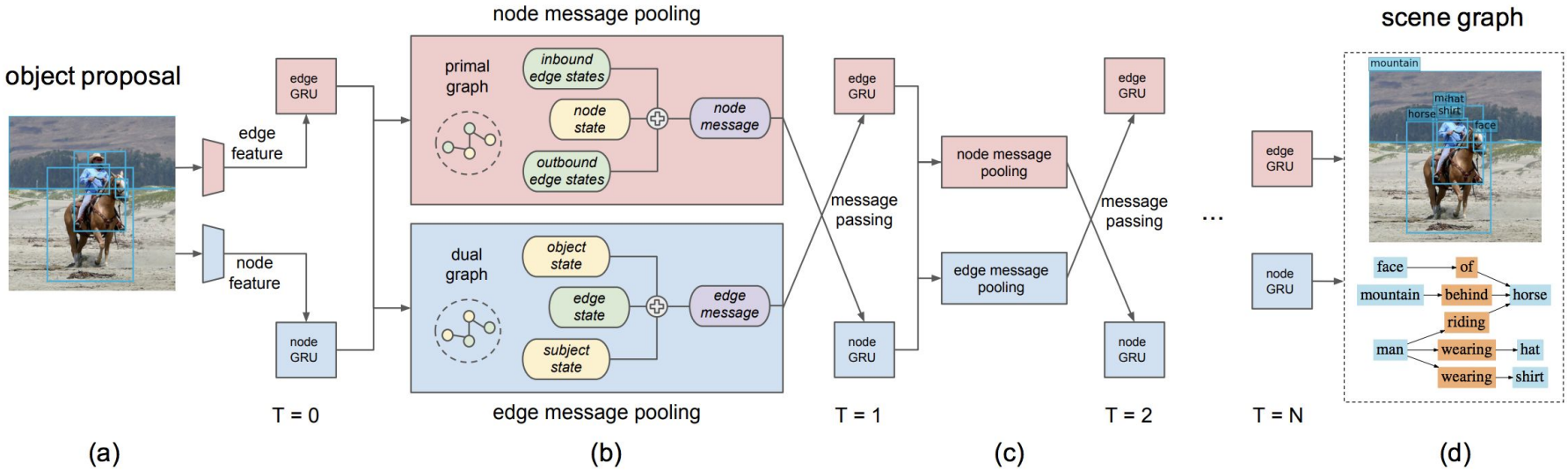


(b): Each node and edge gets multiple messages. Xu et al. use a novel message pooling function to weight each message and fuse them:

$$m_i = \sum_{j:i\to j} \sigma(\mathbf{v}_1^T[h_i, h_{i\to j}])h_{i\to j} + \sum_{j:j\to i} \sigma(\mathbf{v}_2^T[h_i, h_{j\to i}])h_{j\to i} \quad (3)$$

$$m_{i\to j} = \sigma(\mathbf{w}_1^T[h_i, h_{i\to j}])h_i + \sigma(\mathbf{w}_2^T[h_j, h_{i\to j}])h_j \quad (4)$$

w, v are learnable params
***σ** is the sigmoid func.*



(c): Repeat this process with multiple layers. Finally, similarly to faster R-CNN:

- Softmax layer for final object and relationship scores
- Fully-connected layer for bounding box offsets for each obj class

Former uses cross-entropy loss; latter uses l_1 loss.

Implementation

Training: tune the fully connected layers and GRUs

Testing: Given an image I,

1. Use a Region Proposal Network to generate a set of proposed bounding boxes B_I
2. Use a pretrained VGG-16 network to extract visual features
3. Non-Max Suppression to filter boxes down to object proposals
4. Predict outputs for all boxes, edges; create graph

Evaluation

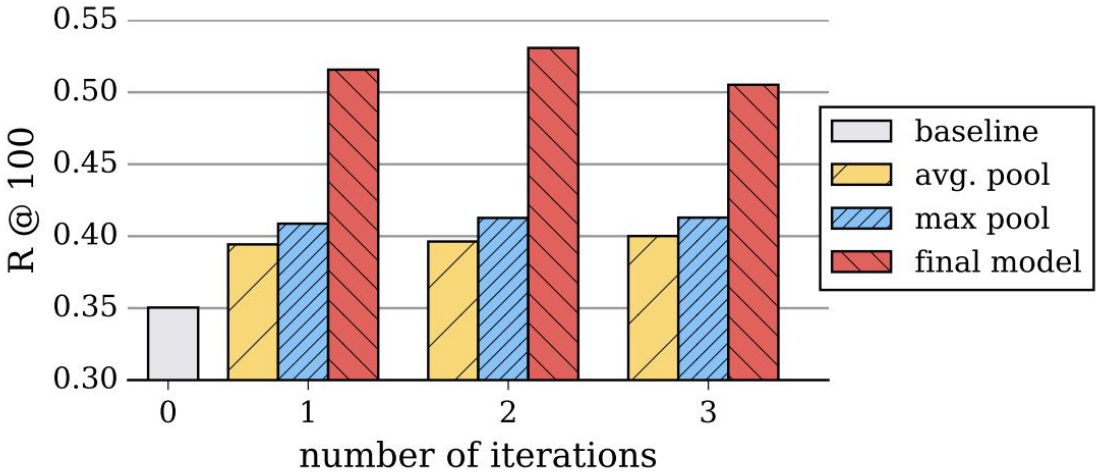
Three tasks:

1. Predicate classification: predict predicates for all pairwise relationships
2. Scene graph classification: predict the predicate and associated object categories for all relationships
3. Scene graph generation: detect a set of objects (0.5 IoU overlap), predict predicates between them

R@50/100: fraction of ground truth relationships in top x most confident predictions for an image (higher = better)

Dataset: Visual Genome, cleaned up (!)

Results



Findings:

- Performance stagnates after 2 iterations
- Novel pooling method very effective

Results

		[26]	avg. pool	max pool	final
PREDCLS	R@50	27.88	32.39	34.33	**44.75**
	R@100	35.04	39.63	41.99	**53.08**
SGCLS	R@50	11.79	15.65	16.31	**21.72**
	R@100	14.11	18.27	18.70	**24.38**
SGGEN	R@50	0.32	2.70	3.03	**3.44**
	R@100	0.47	3.42	3.71	**4.24**

Findings:

- Outperformed a model using only local info ([26])
- Novel pooling method very effective

Results

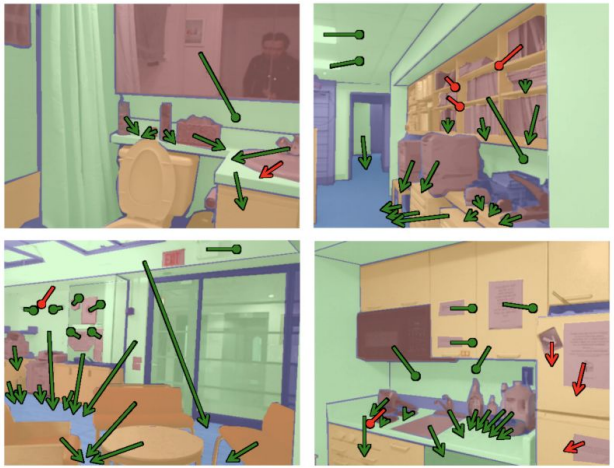
Table 2. Predicate classification recall. We compare our final model (trained with two iterations) with Lu *et al.* [26]. Top 20 most frequent types (sorted by frequency) are shown. The evaluation metric is recall@5.

predicate	[26]	ours	predicate	[26]	ours
on	**99.71**	99.25	under	28.64	**52.73**
has	**98.03**	97.25	sitting on	31.74	**50.17**
in	80.38	**88.30**	standing on	44.44	**61.90**
of	82.47	**96.75**	in front of	26.09	**59.63**
wearing	**98.47**	98.23	attached to	8.45	**29.58**
near	85.16	**96.81**	at	54.08	**70.41**
with	31.85	**88.10**	hanging from	0.00	0.00
above	49.19	**79.73**	over	**9.26**	0.00
holding	61.50	**80.67**	for	12.20	**31.71**
behind	79.35	**92.32**	riding	72.43	**89.72**

Figure 5. Sample predictions from the baseline model and our final model trained with different numbers of message passing iterations. The models take images and object bounding boxes as input, and produce object class labels (blue boxes) and relationship predicates between each pair of objects (orange boxes). In order to keep the visualization interpretable, we only show the relationship (edge) predictions for the pairs of objects (nodes) that have ground-truth relationship annotations.

Results

	Support Accuracy		PredCls	
	t-ag	t-aw	R@50	R@100
Silberman *et al.* [28]	75.9	72.6	-	-
Liao *et al.* [24]	88.4	82.1	-	-
Baseline [26]	87.7	85.3	34.1	50.3
Final model (ours)	**91.2**	**89.0**	**41.8**	**55.5**



From the NYU Depth v2 set:

- Attempt to predict support relation type and struct class of each object
- State of the art results using only RGB images (not RGB-D!)

Question: What are some more novel approaches to scene graph creation from images (ie. who's beaten Xu et al. 2017)?

Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction (link)

By: Herzig et al., 2018

Note: This paper was published less than two weeks ago; while it outperforms Xu et al., the methods are unverified by others

Implications

Scene graphs are pretty broadly useful - they've been successfully used for:

- Image Retrieval (we've seen this)
- 3D Scene Synthesis (brief mention in Johnson et al.)
- Visual Question Answering (coming up!)

We've now learned of methods to find images from scene graphs and scene graphs from images, and of a dense dataset that can be used to improve performance further.

/end