What does Deep Learning tell us about Language?

Mark Johnson

Macquarie University

(with Peter Anderson, Paria Jamshidlou, Martha Palmer, Stephen Gould and The Australian Centre for Robotic Vision)

July, 2018

Deep Learning has revolutionized Natural Language Processing (NLP)

- Deep Learning is spectacularly successful for tasks such as:
  - *Machine translation*
  - *Speech recognition*
  - *Image captioning*
  - *Semantic parsing*
- Primary benefit is *economic*
  - Complex component pipelines replaced with end-to-end models
  - ⇒ Can build apps more *quickly* and *cheaply*

Two faces of NLP/CL

- **Natural Language Processing (NLP):**
  - Building computational devices that perform useful tasks
- **Computational Linguistics (CL):**
  - Understand the *computational nature of human language*
  - Human language processing is *computational* in a way that e.g., astronomy or geology aren't
- *Technology can be more advanced than science*
  - Steam engine $\Rightarrow$ Thermodynamics

Linguistics and CL/NLP

	Uses linguistic grammars	Uses linguistic representations
Symbolic NLP 1980s-1990s	✓	✓
Statistical NLP 1990s-2000s	✗	✓
Deep Neural NLP 2010s-??	✗	✗

- Steady move to *simpler representations* (e.g., dependency parses)
- “All dressed up, but no place to go”: no way to use complex linguistic representations

Changing goals of CL research

- **Symbolic NLP** (1980s - 1990s):
  - Implement linguistic analyses and linguistic theories
- **Statistical NLP** (1990s - 2000s):
  - Define/infer probability distributions over linguistic representations
  - Learn appropriate linguistic generalisations (nonparametric Bayes)
- **Deep Learning NLP** (2010s - ?):
  - Language and other modalities, e.g., vision (?)
  - Language in a broader context beyond the sentence (?)

What does Deep Learning tell us about Language?

- DL *hasn't changed our understanding of language*
  - Can't explain *why the language we hear is by and large the language we speak*
- Main contribution: *demonstrating that a neural net can do these tasks*
  - (this is basically all any computational model does)
  - Shows that *these tasks aren't indicators of intelligence*

  - Andrew Ng: "*If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future.*"

Discrete vs Continuous Categories

- Success of Neural Nets ⇒ Continuous distributed representations instead of discrete linguistic categories?
  - *Linguistic squish*: (Adj) *proud, opposite, near, like, in* (Prep) (Ross 1972)
- Perhaps linguistic knowledge or world knowledge is not discrete?
  - Probabilities are a systematic way of evaluating alternatives
- Or perhaps continuous representations are *mainly useful for learning*?
  - Gradient provides information about loss function in a region

Deep Learning NLP and Linguistics

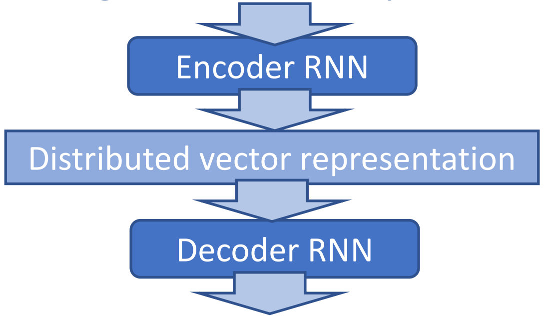
- ***End-to-end trainable systems don't need intermediate representations***
- *Distributed representations* factorize language better than one-hot representations
  - Amazing that SGD can learn such complex models from so little data
  - Handle *data sparsity* better than one-hot representations
  - Neural nets can learn and integrate *world knowledge* (?)
- Linguistic insights are still useful (sometimes!)
  - Especially for *data preparation* and *evaluation*

Case study: Semantic Parsing and Centering Theory

Semantic Parsing

- A Semantic Parser maps utterances to *executable* “logical forms” (e.g., database queries)
- Seq2seq translation model (Dong and Lapata 2016)

“*Find parking close to the Space Needle*”



[topic: [amenity: *parking*,
near: [poi: “*Space Needle*”]],
action: [show: Topic]]

Semantic Parser Logical Forms

- The standard output from a semantic parser is an "*intent + slots*" representation
  - *"How do I get from Bellevue to the Space Needle"*
  - Intent: *route*, From: *Bellevue*, To: *Space Needle*
- Our semantic parser uses *compositional logical forms*:
  - *"Find parking near a Thai restaurant that's not too far from the Space Needle"*

[topic: [amenity: *parking*,
near: [amenity: *restaurant*, cuisine: *Thai*,
near: [poi: *"Space Needle"*]],
action: [show: Topic]]

Centering theory

"Show me the Space Needle"

Forward-looking center

Backward-looking center

"Now find a parking lot near it"

Forward-looking center

Backward-looking center

"What time does it close?"

Forward-looking center

- Forward-looking center = Topic
  Backward-looking center = PrevTopic
- See Grosz (1995), Joshi, Prince and Walker (1998)

Centering theory and Semantic Parsing

- **Multi-intent requests:**
  "Get me the best Thai restaurant in Bellevue, and send its address to Phil"
  [topic: [amenity: restaurant, cuisine: Thai,
  in: [locality: Bellevue],
  rating: argmax],
  action: [*show: Topic*],
  action: [*send: [address: Topic*], recipient: Phil]]
- **Follow-up requests**:
  "What's the closest parking garage to it, and when does it close?"
  [*topic*: [*amenity: parking, nearest-to: PrevTopic*],
  action: [show: Topic],
  action: [show: [end: [open-hrs: Topic]]]]

Manufacturing data for semantic parsing

- Quantity and quality of training data determines performance of any machine learning system (including DNNs)
- ⇒ Crowd-sourcing for *manufacturing training data*
- Training data manufactured to cover desired range of linguistic constructions
- *Active learning* ensures that training data contains sufficient examples of each combination of linguistic constructions

- See Duong et al (2018) “Active learning for deep semantic parsing”

Semantic Parsing conclusions

- Uses generic DNN encoder / decoder modules, not POS tagger, parser, semantic interpreter, etc.
  - Intermediate representations are *distributed vectors,* not linguistic representations
- *Faster* to develop, *cheaper* to build, *better* performance
  - Surrounding technology now rate-limiting step
  - ⇒ *crowd-sourcing training data* with *active learning*
- Insights from Centering Theory let us handle Multi-Intent and Follow-Ups
  - *End-to-end dialog models* may make this redundant!

Case study: Do Deep Models learn Linguistic Constraints?

With thanks to Emily Bender and Tom Wasow

Why did symbolic NLP fall out of favour?

- Successfully implements linguistic constraints
  - No single catastrophic failure
- *Coverage* / *ambiguity* dilemma
  - $\Rightarrow$ loosen / tighten grammar
  - Probability provides a systematic solution
- Never had a convincing account of *robustness*
  - Linguistic theory focuses on *grammaticality*
- Overwhelmed by *lexical detail* and *world knowledge*
  - E.g., *make a cake, a fire, a dinner, an enemy, love, war, peace*
  - Sociology of field didn't reward grammar development

What do Deep NNs know about language?

- Distributed representations ⇒ DNNs are “*black boxes*”
- Symbolic and statistical models are “glass boxes”
  - In theory, but often not in practice
- Perhaps DNNs “understand” language only using world knowledge, ignoring linguistic constraints? (Shank 1990)
- Approach: *apply DNN models to examples where linguistic constraints force a particular interpretation*
  - Use AllenNLP parsing and semantic role labeling models

Semantic role labeling

- Semantic Role Labeling (SRL) identifies "*who did what to whom*" in a sentence
- We use the AllenNLP SRL model
  - This is the He et al (2017) model, which uses a deep BiLSTM, plus ELMO embeddings
- Algorithm overview:
  - Identify the predicates
  - For each predicate, identify its argument phrases
  - Use a beam decoder to find consistent analysis

Plausible vs Implausible roles

- *The dog bit the man.*

  ARGO ARG1

- *The man bit the dog.*

  ARGO ARG1

- *The company bought the investor.*

  ARGO ARG1

- *The court required the officials to leave the country.*

  ARGO ARG2 ARG1

- *The officials required the court to leave the country.*

  ARGO ARG2 ARG1

- *The country required the court to leave the officials.*

  ARGO ARG2 ARG1

WH-dependencies

- *The government should purchase the firm.*
  ARGO ARG1
- *Who should purchase the firm?*
  ARGO ARG1
- *What should the government purchase?*
  ARG1 ARGO

Long-range WH-dependencies

- *Analysts expected that the director would claim that the company proposed that the government should purchase the firm.*

- *Who did the analysts expect would claim that*

  ARGO

  the company proposed that the government should purchase the firm. ARG1

- *Who did analysts expect that the director would claim*

  ARGO ARGO

  proposed that the government should purchase the firm. ARG1

Lexical ideosyncracies

- *Which company did the analyst advise the investor to sell?*
  ARG1 ARGO
- AllenNLP SRL fails to find WH-phrase ARG1 dependency
- But it does find the WH-dependency with apparently irrelevant changes
  - *sell* ⇒ *buy*
  - *advise* ⇒ *suggest, persuade, promise, force, …*
  - *Which company* ⇒ *what*
  - *the analyst* ⇒ *analysts* and *the investor* ⇒ *investors*

WH-dependencies and argument structure

- *The manager wanted to talk to the director.*
  ARGO ARG2

- *Which director did the manager want to talk to?*
  ARGO ARG2

- *Which director did the manager want to talk?*
  ARG0 / ARG1 ARGO

- *Which director did the manager force to talk?*
  ARG0 / ARG1 ARGO

Pronouns and anaphora

- *The analyst said that the manager promoted himself.*
- *The analyst said that the manager promoted him.*

  ?

- *The director promised the manager to promote himself.*
- *The director persuaded the manager to promote himself.*
- *Which manager did the director persuade to promote him?*
- *Which manager did the director talk to?*
  - Doesn't happen with *Which manager did the director promote?*

Conclusions

- The DNN models *cover linguistic constructions*, given enough training data
  - They learn local approximations to linguistic constraints
  - Often fail on longer-range cases
    - Longer range cases are rare in real data
- DNN behavior seems *lexically idiosyncratic*
  - Not *capturing linguistic generalisations* involving syntactic categories

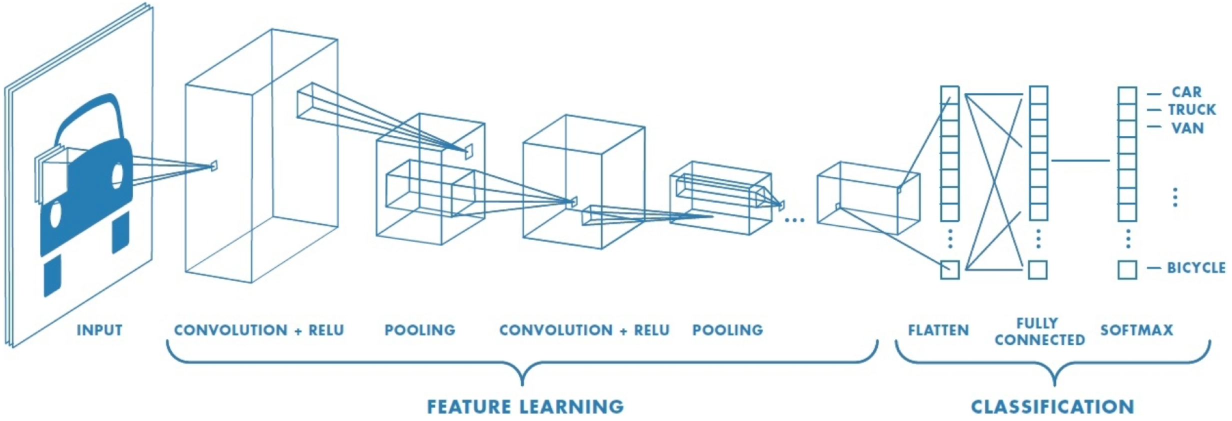
- Will humans experience a linguistic *“uncanny valley”* interacting with DNNs?

Parsing speech with disfluencies

- Speech disfluencies exhibit *crossing dependencies* that head-driven constituency parsers can't find
  - *I want a flight to Boston, uh, to Denver tomorrow*
  - ⇒ Specialized disfluency detectors
- The AllenNLP constituency parser, retrained on Switchboard data, *finds disfluencies while parsing*
  - 82% disfluency f-score without hyperparameter tuning
  - SOA: 89% disfluency f-score
- ⇒ *AllenNLP parser is not restricted to head-dependency relationships*

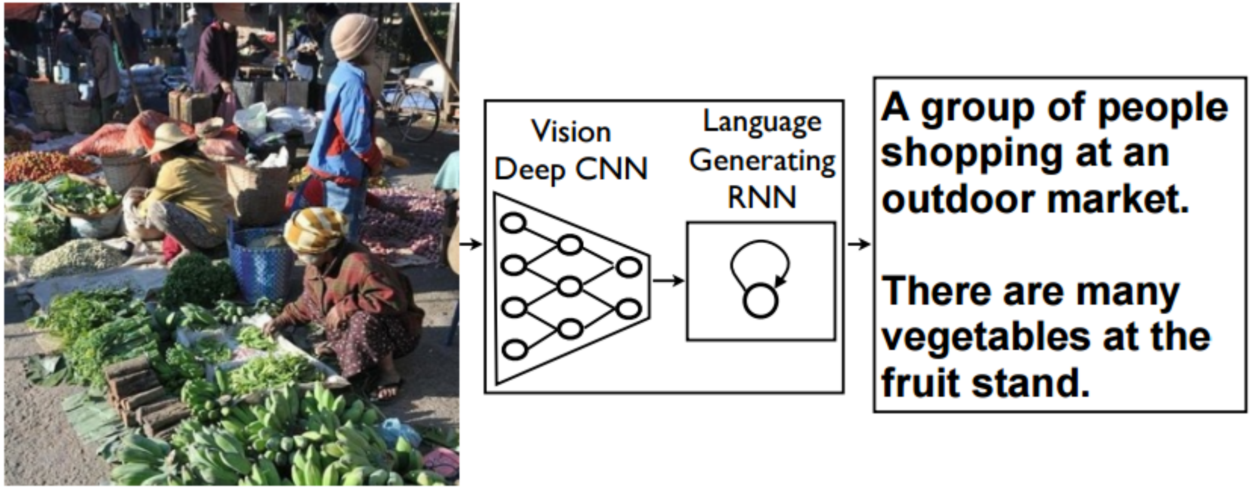
Case study: Semantic analysis for evaluating image captions

Image labeling



- Image labeling tags (objects in) images

Image captioning



- Caption image with a phrase or sentence

- Figure reproduced from Vinyals et al 2015.

COCO Captions test server

Table-C5 **Table-C40** Challenge2015

Copy to Clipboard | Export to CSV

Search:

	CIDEr-D	METEOR	Rouge-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	SPICE	date
Human	0.910	0.335	0.626	0.880	0.744	0.603	0.471	0.740	2015-03-23
panderson@MSR/ACRV	1.205	0.367	0.724	0.952	0.888	0.794	0.685	0.715	2017-07-22
DEEPAI	1.194	0.364	0.721	0.935	0.871	0.778	0.670	0.711	2017-07-22
TencentVision	1.224	0.366	0.722	0.947	0.884	0.786	0.673	0.704	2017-08-07
CASIA_IVA	1.188	0.362	0.719	0.934	0.870	0.776	0.669	0.702	2017-07-22
bmc-uestc	1.046	0.364	0.710	0.926	0.850	0.749	0.642	0.695	2017-08-02
CAP_BMC	1.047	0.365	0.710	0.924	0.848	0.749	0.645	0.693	2017-06-13
SenmaoYe	1.059	0.370	0.712	0.922	0.843	0.743	0.639	0.692	2017-04-29
Watson Multimodal	1.167	0.355	0.707	0.931	0.860	0.759	0.645	0.689	2017-03-17

Automatic caption evaluation

- Benchmark datasets require fast to compute, accurate and inexpensive evaluation metrics
- Good metrics can be used to help construct better models

The man at bat readies to swing at the pitch while the umpire looks on.

Caption evaluation task

Given a candidate caption $\boldsymbol{y}_i$ and a set of reference captions S_i, compute a similarity score between $\boldsymbol{y}_i$ and R_i.

<u>Candidate caption $\boldsymbol{y}_i$:</u>

`A teal green car with yellow and red flames painted on the front.`

Caption evaluation task

Given a candidate caption $\boldsymbol{y}_i$ and a set of reference captions S_i, compute a similarity score between $\boldsymbol{y}_i$ and R_i.

Candidate caption $\boldsymbol{y}_i$:

`A teal green car with yellow and red flames painted on the front.`

Reference captions R_i:

`An old green car with a flame design painted on the front of it.`

`A photograph of a european car.`

`An old school car with flames.`

`A picture of a car parked.`

`A car is painted with flames on the front.`

Caption evaluation task

Given a candidate caption $\boldsymbol{y}_i$ and a set of reference captions S_i, compute a similarity score between $\boldsymbol{y}_i$ and R_i.

Candidate caption $\boldsymbol{y}_i$:

`A teal green car with yellow and red flames painted on the front.`

Reference captions R_i:

`An old green car with a flame design painted on the front of it.`

`A photograph of a european car.`

`An old school car with flames.`

`A picture of a car parked.`

`A car is painted with flames on the front.`

Caption evaluation task

Given a candidate caption $\boldsymbol{y}_i$ and a set of reference captions S_i, compute a similarity score between $\boldsymbol{y}_i$ and R_i.

Candidate caption $\boldsymbol{y}_i$:

`A teal` **`green car`** `with yellow and red flames painted on the front.`

Reference captions R_i:

`An old` **`green car`** `with a flame design painted on the front of it.`

`A photograph of a european` **`car`**`.`

`An old school` **`car`** `with flames.`

`A picture of a` **`car`** `parked.`

`A` **`car`** `is painted with flames on the front.`

Caption evaluation task

Given a candidate caption $\boldsymbol{y}_i$ and a set of reference captions S_i, compute a similarity score between $\boldsymbol{y}_i$ and R_i.

Candidate caption $\boldsymbol{y}_i$:

`A teal` **`green car`** `with yellow and red` **`flames painted on the front`**`.`

Reference captions R_i:

`An old` **`green car`** `with a flame design painted on the front of it.`

`A photograph of a european` **`car`**`.`

`An old school` **`car`** `with flames.`

`A picture of a` **`car`** `parked.`

`A` **`car`** `is` **`painted with flames on the front`**`.`

N-grams

N=1: `A teal green car`

unigrams: `A, teal, green, car`

N=2: `A teal green car`

bigrams: `A teal, teal green, green car`

N=3: `A teal green car`

trigrams: `A teal green, teal green car`

N-gram evaluation metrics

- **BLEU:** Precision with brevity penalty, geometric mean over n-grams
- **ROUGE-L:** *F*-score based on Longest Common Substring
- **METEOR:** Align fragments, take harmonic mean of precision & recall
- **CIDEr:** Cosine similarity with TF-IDF weighting

N-gram limitations

'False positive'
(High n-gram similarity)

A young girl standing on top of a tennis court.

A giraffe standing on top of a green field.

'False negative'
(Low n-gram similarity)

A shiny metal pot filled with some diced veggies.

The pan on the stove has chopped vegetables in it.

…n-gram overlap is not necessary or sufficient for two sentences to mean the same

Source: MS COCO Captions dataset

Is this a good caption?

A young girl standing on top of a basketball court

Is this a good caption?

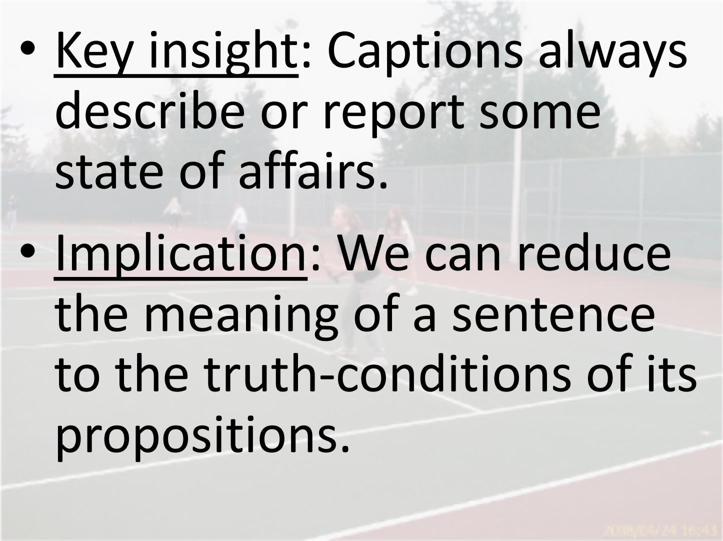
`A young girl standing on top of a basketball court`

Atomic propositions:

1. There is girl
2. The girl is young
3. The girl is standing
4. There is court
5. The court is for basketball
6. The girl is on the court

Is this a good caption?

`A young girl standing on top of a basketball court`



- Key insight: Captions always describe or report some state of affairs.
- Implication: We can reduce the meaning of a sentence to the truth-conditions of its propositions.

Atomic propositions:

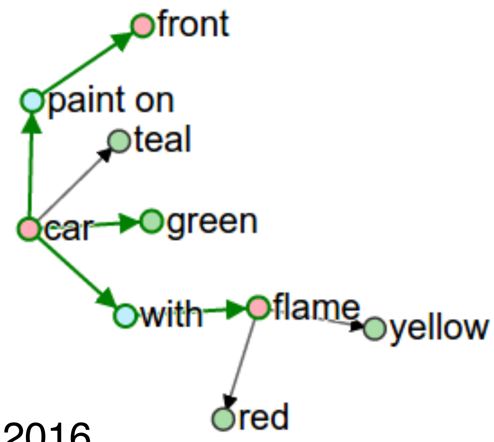
1. There is girl
2. The girl is young
3. The girl is standing
4. There is court
5. The court is for basketball
6. The girl is on the court

Our approach: SPICE

- Map candidate and reference captions to a "scene graph"
  - Extracted from a dependency parse

<u>Candidate caption $\boldsymbol{y}_i$:</u>

```
A teal green car with yellow and red flames painted on the
front.
```



Anderson et al. ECCV 2016

Our approach: SPICE

<u>Reference captions R_i:</u>

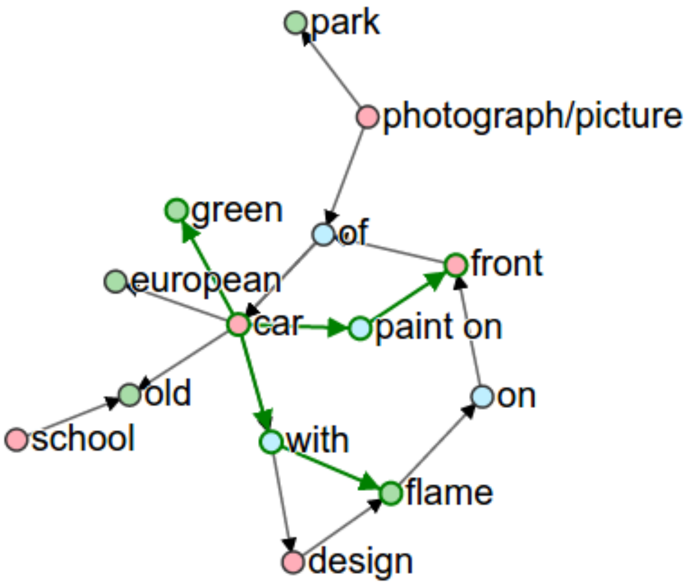
An old green car with a flame design painted on the front of it.

A photograph of a european car.

An old school car with flames.

A picture of a car parked.

A car is painted with flames on the front.

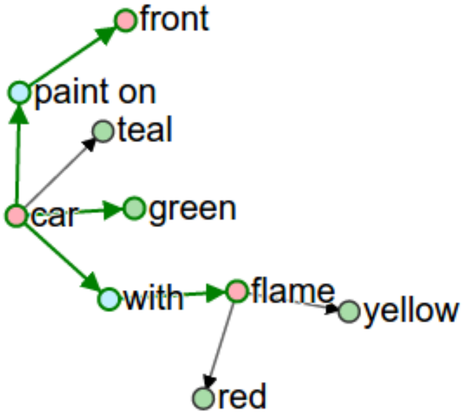


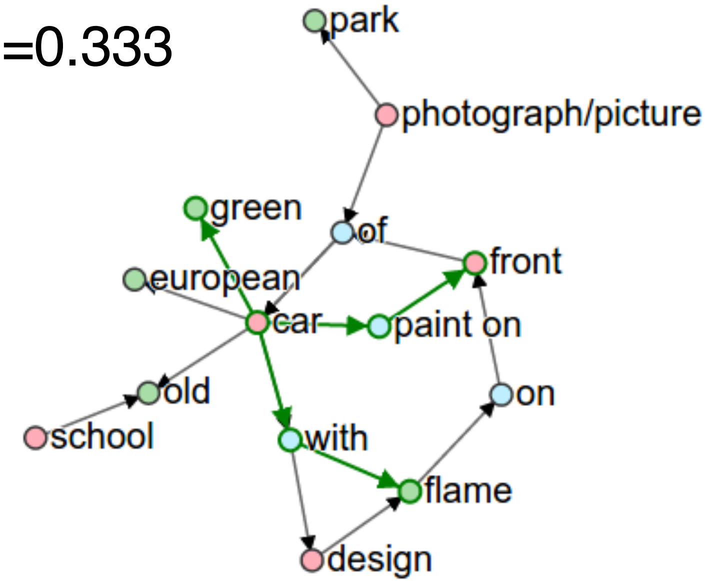
Anderson et al. ECCV 2016

Our approach: SPICE

- Report an F-score over matching tuples in the candidate and reference scene graphs

F= 0.444, Pr=0.667, Re=0.333

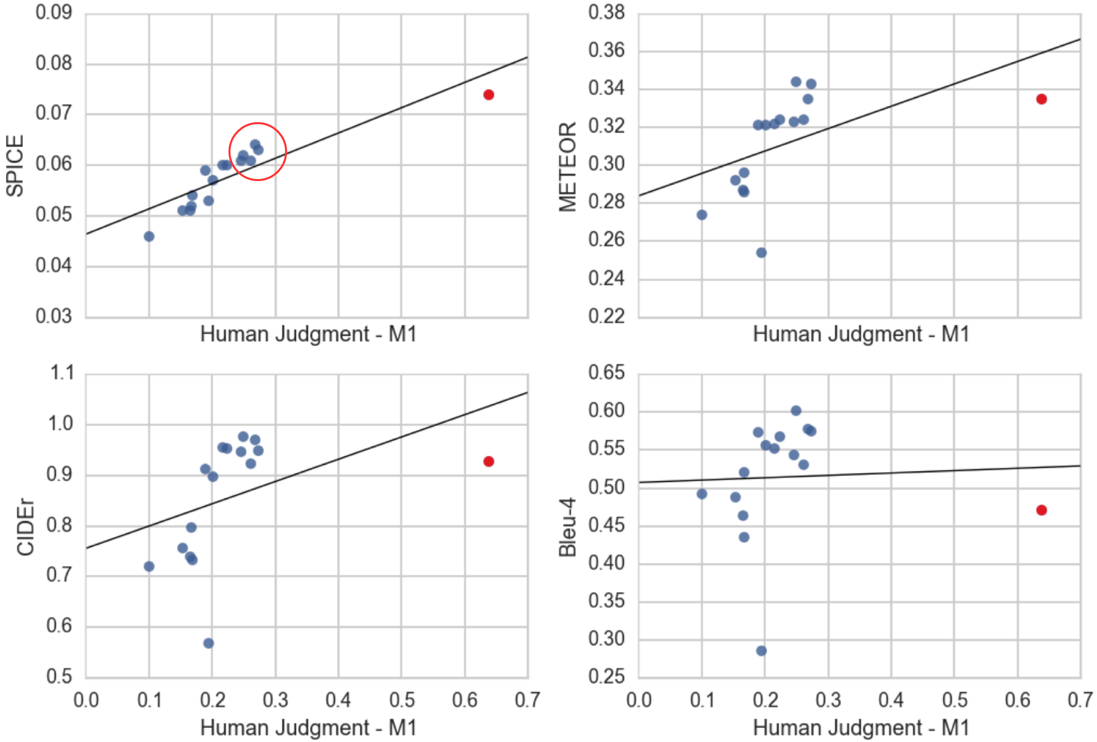




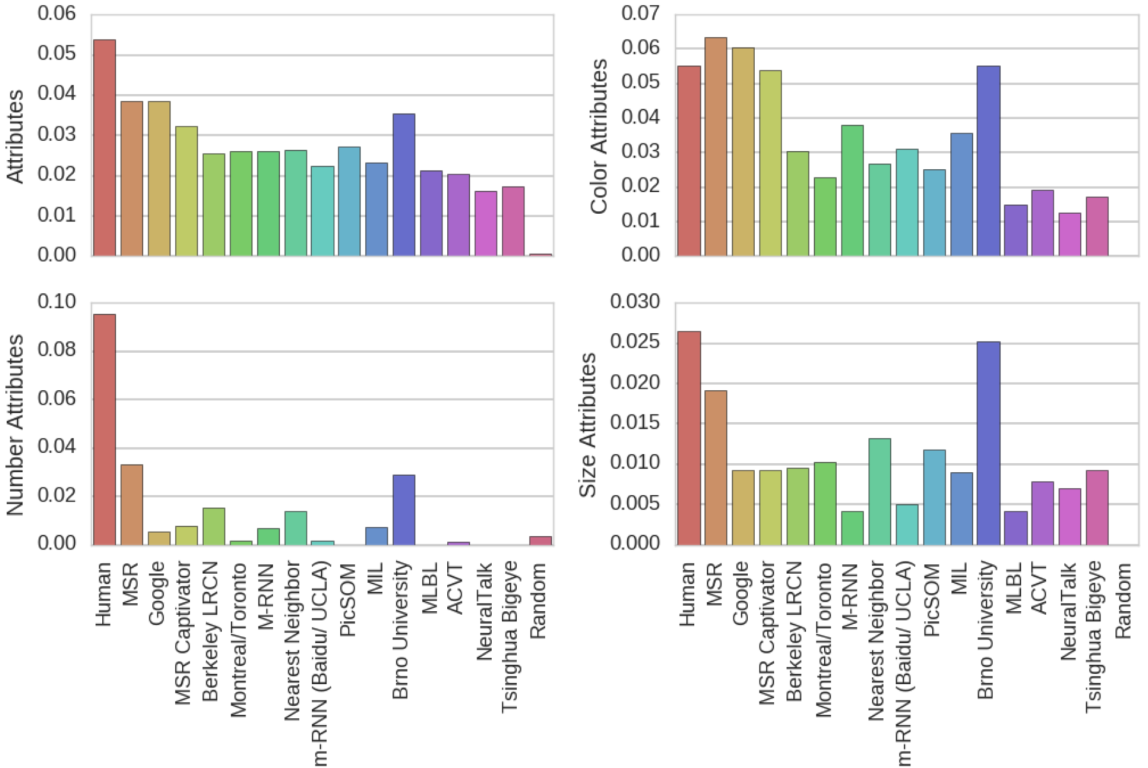
Anderson et al. ECCV 2016

Re-scoring the 2015 Captioning Challenge

SPICE picks human first, and the same top-5 models as human evaluators.



SPICE for error analysis



Conclusions

- Linguistics isn't used in design of image captioning model
- But linguistic insights let us formulate an evaluation metric that is *more consistent with human judgements*
  - Also permits a more refined evaluation

Case study: Image captioning and Semi-supervised learning

Image captioning “in the wild”

- How do we scale-up to millions of visual categories?

A zebra is laying down in the grass.

Integrating labeling and captioning

- *Image labeling* has abundant training data
  ⇒ good coverage of rare objects
- *Image captioning* has much less training data
  ⇒ poor coverage of rare objects
- Goal: use an broad-coverage *image labeler* to improve an *image captioning model*
- Approach:
  - *Image labeler* identifies key words
  - *Constrained decoder* forces captioner to use these keywords
  - *Pretrained word vectors* ensure keywords are used appropriately

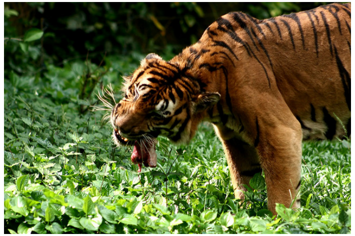
High-level idea

- Learn new visual concepts from labelled images (available in abundance)

Limited data

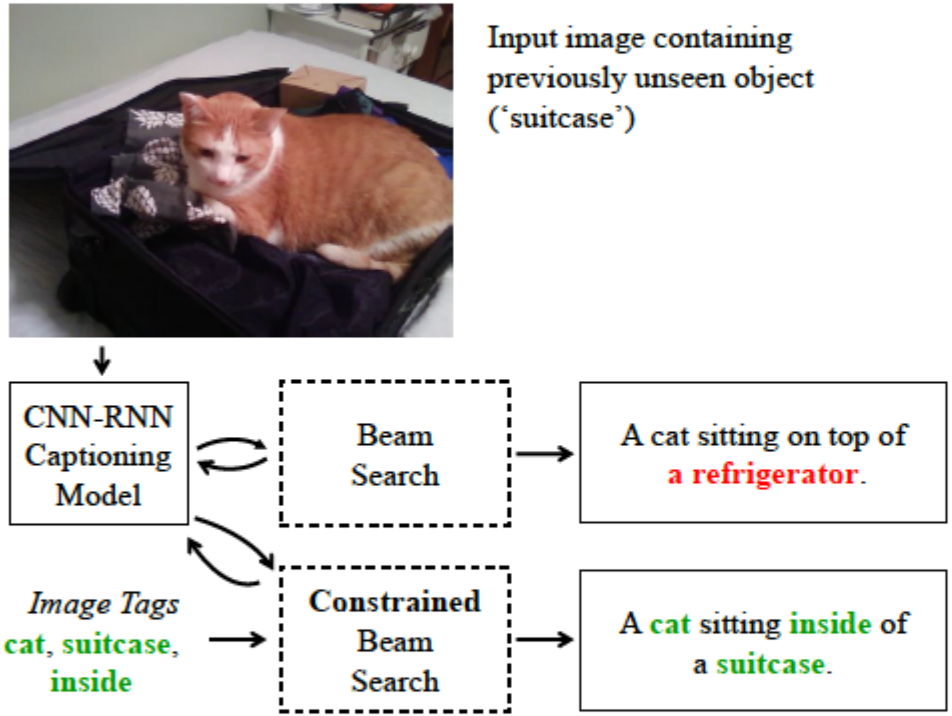
A very pretty zebra crossing a paved road.

Abundant data



tiger, plants, meat, eating

Using image labels to improve captions at run-time



Vocabulary expansion with pre-trained word vectors

- Introduce pretrained GloVe[2] 300D embeddings at both the LSTM input and output layers (W_e):

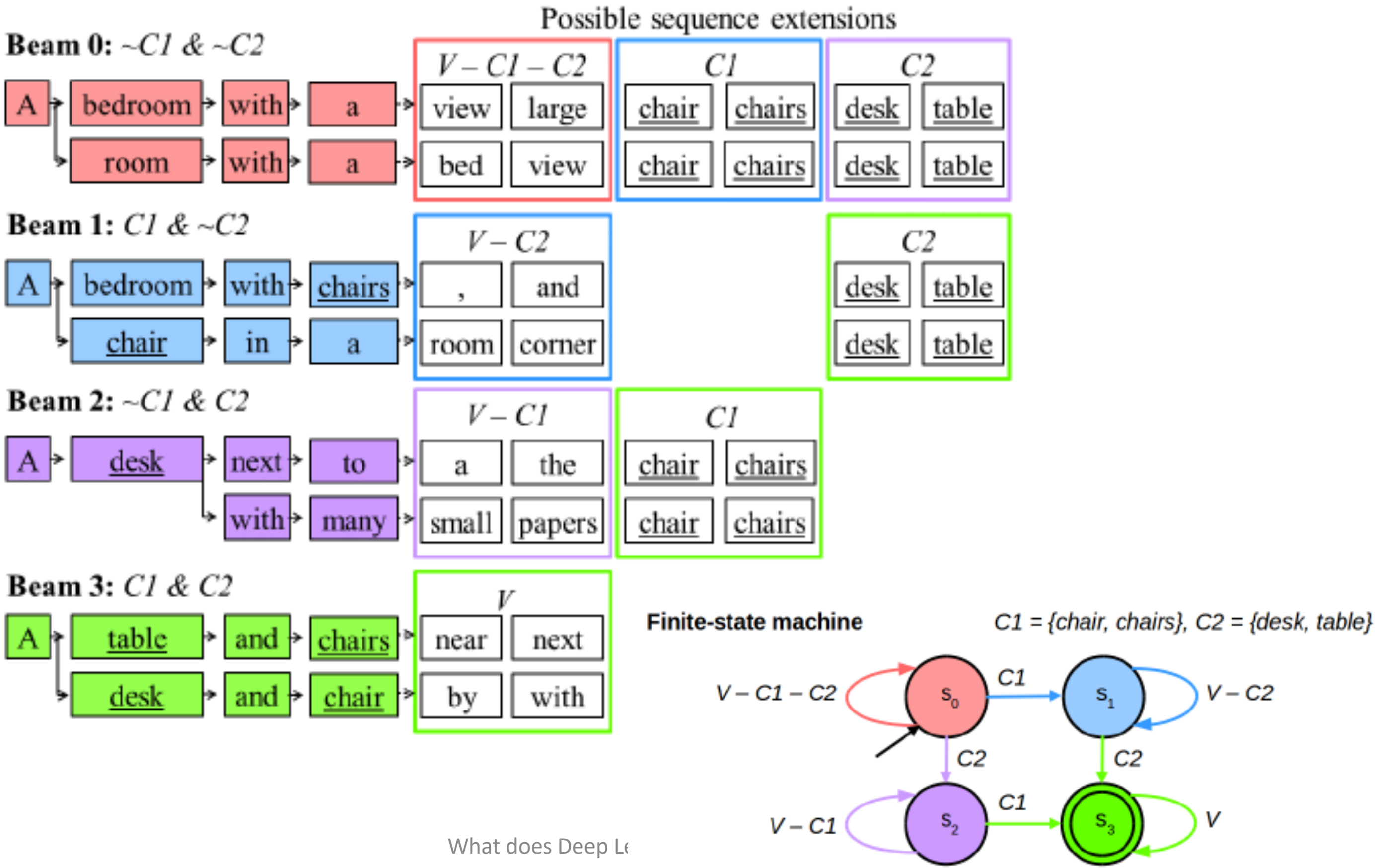
$$v_t = \tanh\left(W_v h_t^2 + b_v\right)$$

$$p(y_t \mid y_{t-1}, \ldots, y_1, I) = \mathrm{softmax}\left(W_e^T v_t\right)$$

- W_e fixed during training with minimal performance impact (using conventional cross-entropy loss).
- Model learns to predict 300D vectors v_t with a high dot-product similarity with the GloVe embedding of the correct output word.
- New vocabulary introduced at test time by concatenating the GloVe vector as an additional column to W_e

[2] 'GloVe: Global Vectors for Word Representation', Pennington et. al. EMNLP 2014

Constrained decoder with finite-state constraints



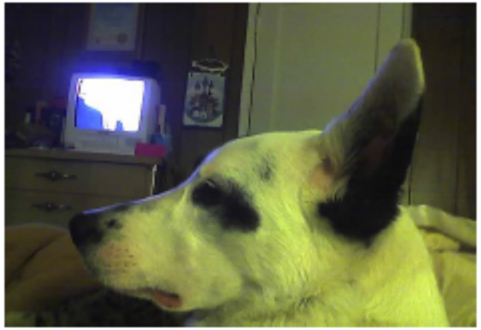
Examples

Base: A woman is playing tennis on a tennis court. **Tags:** tennis, player, ball, racket. **Base+T4:** A tennis player swinging a racket at a ball.

Base: A man standing next to a yellow train. **Tags:** bus, yellow, next, street. **Base+T4:** A man standing next to a yellow bus on the street.

Base: A close up of a cow on a dirt ground. **Tags:** zebra, zoo, enclosure, standing. **Base+T4:** A zebra standing in front of a zoo enclosure.

Failure cases



Base: A dog is sitting in front of a tv. **Tags:** dog, head, television, cat. **Base+T4:** A dog with a cat on its head watching television.

Base: A group of people playing a game of tennis. **Tags:** pink, tennis, crowd, ball. **Base+T4:** A crowd of people standing around a pink tennis ball.

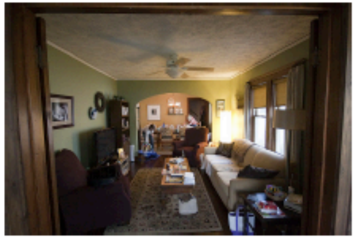
Semi-supervised learning

- The constrained captions are (usually) good
  - Especially with “gold” image tags
- Idea: *train the captioning model on captions produced by the constrained decoder*
  - Iterate this process (a la EM)
- Better than run-time constrained decoding alone
  - Even when constrained decoding is not used at run-time

Examples

A couple of zebra standing next to each other.
A close up of a giraffe with its head.

A white bus driving down a city street.
A food truck parked on the side of a road.



A brown couch sitting in a living room.
A living room filled with lots of furniture.

A microwave sitting on top of a counter.
A picture of an oven in a kitchen.

A set of pictures showing a slice of pizza.
A collage of four pictures of food.

A little girl holding a tennis racket.
A young girl is standing in the tennis court.

A group of people walking down a city street.
A group of people walking down a street.

A woman wearing a blue tie holding a yellow toothbrush.
A woman in the kitchen with a toothbrush in her hand.

Experimental results on COCO novel object captioning

	Training Data		CBS	Out-of-Domain Scores				In-Domain Scores		
	Captions	Labels	Labels	SPICE	METEOR	CIDEr	F1	SPICE	METEOR	CIDEr
1	◐			14.4	22.1	69.5	0.0	**19.9**	**26.5**	**108.6**
2	◐		▲	15.9	23.1	74.8	26.9	19.7	26.2	102.4
3	◐	●		**18.3**	**25.5**	**94.3**	**63.4**	18.9	25.9	101.2
4	◐	●	▲	18.2	25.2	92.5	62.4	19.1	25.9	99.5
5	◐		★	18.0	24.5	82.5	30.4	22.3	27.9	109.7
6	◐	●	★	20.1	26.4	95.5	65.0	21.7	27.5	106.6
7	●			20.1	27.0	111.5	69.0	20.0	26.7	109.5

● = full training set, ◐ = impoverished training set, ▲= constrained beam search (CBS) decoding with predicted labels, ★= CBS decoding with ground-truth labels

- On images with out-of-domain objects, *label constraints improve performance*
- Using label constraints at training time is better than at run time

Comparision with other models

Model	CNN	Out-of-Domain Scores				In-Domain Scores		
		SPICE	METEOR	CIDEr	F1	SPICE	METEOR	CIDEr
DCC [20]	VGG-16	13.4	21.0	59.1	39.8	15.9	23.0	77.2
NOC [21]	VGG-16	-	21.3	-	48.8	-	-	-
C-LSTM [22]	VGG-16	-	23.0	-	55.7	-	-	-
LRCN + CBS [19]	VGG-16	15.9	23.3	77.9	54.0	18.0	24.5	86.3
LRCN + CBS [19]	Res-50	16.4	23.6	77.6	53.3	18.4	24.9	88.0
NBT [23]	VGG-16	15.7	22.8	77.0	48.5	17.5	24.3	87.4
NBT + CBS [23]	Res-101	17.4	24.1	86.0	**70.3**	18.0	25.0	92.1
Ours	Res-101	**17.9**	**25.4**	**94.5**	63.0	**19.0**	**25.9**	**101.1**

- CBS = Constrained Beam Search

Training time constraints allow captioner to learn to recognize new objects

A woman holding a <u>tennis racket</u> on a court.

- *Visual attention* shows model has learnt to recognize the unknown phrase *tennis racket*.

Why semi-supervised learning is interesting

- Most DNN work explores *supervised learning* (or reinforcement learning) for *end-to-end tasks*
- *Unsupervised learning* is cognitively more relevant
  - Standard approach: *supervised learning on proxy tasks* (e.g., word vectors)
    - Success depends on how well proxy task is chosen
  - Distributed representations ⇒ unclear what is learnt
  - Pre-DNN Bayesian models *produce explicit output*
    - E.g., models learning word segmentation and word reference

Conclusions

Conclusions and future work

- Deep learning is a *faster, better, cheaper* way of building NLP applications
  - They *cover*, but don't *capture*, *linguistic generalisations*
  - ⇒ Linguistic *uncanny valley*?
  - Can't explain *why the language we hear is the language we speak*
- Linguistic insights are useful for:
  - Designing representations used by system
  - *Evaluating* NLP systems
  - Producing *training data* for NLP systems
- Most NLP DNNs are trained using *supervised learning* or *reinforcement learning*
  - Semi-supervised/unsupervised learning also useful
  - Might let us address *scientifically interesting questions*