Preliminaries

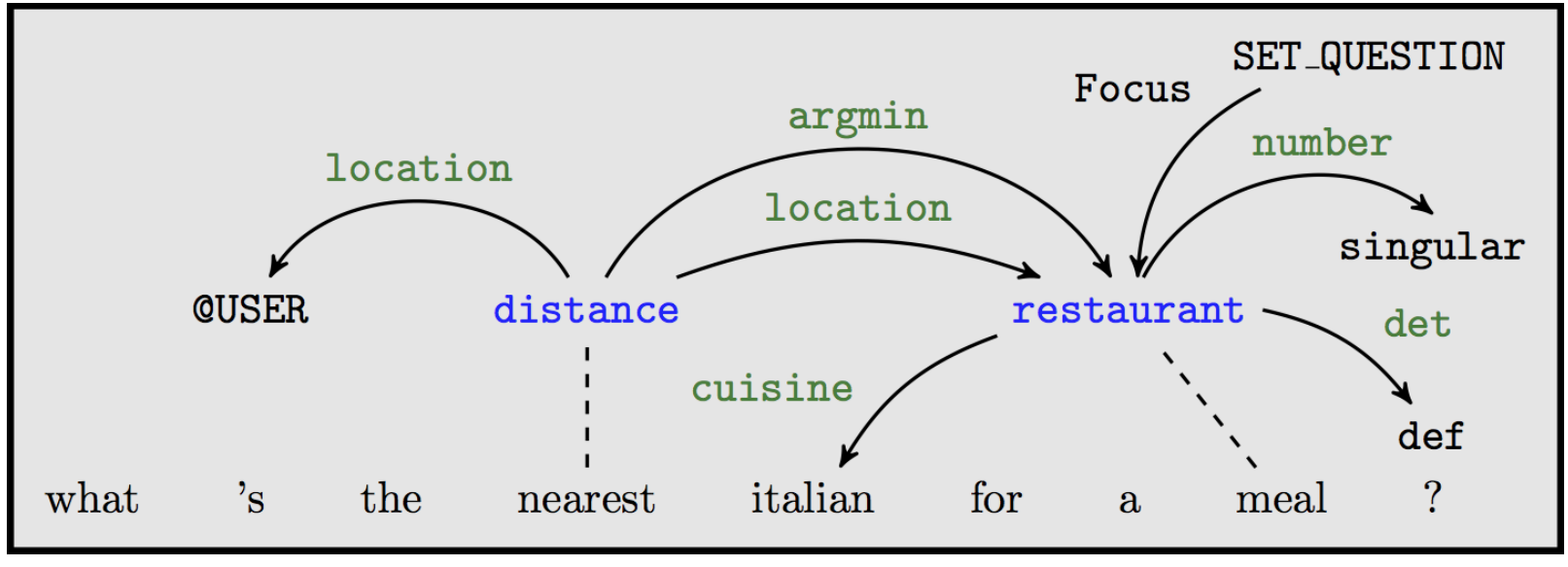
Structured prediction

I	studied	in	London	with	Sebastian	Riedel
PRP	VBD	IN	NNP	IN	NNP	NNP
O	O	O	B-LOC	O	B-PER	I-PER

- part of speech (PoS) tagging
- named entity recognition (NER)

Input: a sentence $\mathbf{x} = [x_1 \dots x_N]$
Output: a sequence of labels $\mathbf{y} = [y_1 \dots y_N] \in \mathcal{Y}^N$

Structured prediction



Semantic parsing, but also syntactic parsing, semantic role labeling, question answering over knowledge bases, etc.)

Input: a sentence $\mathbf{x} = [x_1 \ldots x_N]$
Output: a meaning representation graph $\mathbf{G} = (V, E) \in \mathcal{G}_{\mathbf{x}}$

Structured prediction

INPUT:

```
predicate= INFORM
name = "The Saffron Brasserie"
type = placetoeat
eattype = restaurant
area = riverside, "addenbrookes"
near = "The Cambridge Squash", "The Mill"
```

OUTPUT:

The Saffron Brasserie is a restaurant at the side of the river near the Cambridge Squash and the Mill in the area of Addenbrookes

Natural language generation (NLG), but also summarization, decoding in machine translation, etc.

Input: a meaning representation
Output: a sentence $\mathbf{w} = [w_1 \ldots w_N], w \in \mathcal{V} \cup END, w_N = END$

Two main paradigms

Joint modeling, a.k.a:

- global inference
- structured prediction

Incremental modeling, a.k.a:

- local
- greedy
- pipeline
- transition-based
- history-based

Joint modeling

A model (e.g. conditional random field) that scores complete outputs (e.g. label sequences):

$$\hat{\mathbf{y}} = \hat{y}_1 \ldots \hat{y}_N = \underset{Y \in \mathcal{Y}^N}{\arg\max} f(y_1 \ldots y_N, \mathbf{x})$$

- no error propagation
- exhaustive exploration of the search space
- large/complex search spaces are challenging
- efficient dynamic programming restricts modelling flexibility (i.e. Markov assumptions)

Incremental modeling

A classifier that predicts one label at a time giv en the previous predictions:

$$\hat{\mathbf{y}} = \begin{aligned} \hat{y}_1 &= \arg\max_{y \in \mathcal{Y}} f(y, \mathbf{x}), \\ \hat{y}_2 &= \arg\max_{y \in \mathcal{Y}} f(y, \mathbf{x}, \hat{y}_1), \cdots \\ \hat{y}_N &= \arg\max_{y \in \mathcal{Y}} f(y, \mathbf{x}, \hat{y}_1 \ldots \hat{y}_{N-1}) \end{aligned}$$

- use our favourite classifier
- no restrictions on features
- prone to error propagation (i.i.d. assumption brok en)
- local model not trained wrt the task-level loss

Imitation learning

Improve incremental modeling to:

- address error-propagation
- train wrt the task-level loss function

Meta-learning: use our favourite classifier and features, but generate better (non-i.i.d.) training data

But let's see some basic concepts first

Transition system

The **actions** $\mathcal{A}$ the classifier f can predict and their effect on the **state** which keeps track of the prediction: $S_{t+1} = (S_1, \alpha_1 \ldots \alpha_t)$

Input: $\mathbf{x}$
$state\ S_1 = initialize(\mathbf{x})$; $timestep\ t = 1$
while S_t not final **do**
  $action\ \alpha_t = \arg\max_{\alpha \in \mathcal{A}} f(\alpha, \mathbf{x})$
  $S_{c+1} = update(\alpha_t, S_t)$; $t = t + 1$
Output: $S_{final} = S_t$

- **PoS/NER tagging?** for each word in the sentence, left-to-right, predict a P oS tag which is added to the output
- **NLG?** predict a word from the vocabulary that is added to the output until the `END`

Task loss

Given S_{final}, how does it compare to the gold standard $\mathbf{y}$?

$$loss = L(S_{final}, \mathbf{y}) \geq 0$$

- **PoS tagging?** Hamming loss: number of incorrect tags
- **NER?** number of false positives and false negatives
- **NLG?** BLEU: % of n-grams predicted present in the gold reference(s) ($L = 1 - BLEU(s_{final}, \mathbf{y})$)

Goal: models minimizing the loss on unseen test data

Decomposable losses

A loss is **decomposable** if it is the sum of the losses for each action α_t independently of the future actions $[\alpha_{t+1} \dots \alpha_T]$:

$$L(S, \mathbf{y}) = \sum_{t=1}^{T} \ell(\alpha_t, \mathbf{y}, [\alpha_1 \dots \alpha_{t-1}])$$

I	studied	in	London	with	Sebastian	Riedel
PRP	VBD	IN	NNP	IN	NNP	NNP
O	O	O	B-LOC	O	B-PER	I-PER

Can we tell $\ell(\alpha_6, \cdot)$ for

- PoS tagging? **Yes!** $\ell(\alpha_6, \cdot) = 0$ no matter α_7
- NER? **No!** If α_7 is
  - I-PER: $\ell(\alpha_6, \cdot) = 0$ (correct)
  - O: $\ell(\alpha_6, \cdot) = 2$ (1 FP and 1 FN)
  - B-*: $\ell(\alpha_6, \cdot) = 3$ (2 FP and 1 FN)

Non-decomposable loss

Is BLEU score decomposable?

$$BLEU([\alpha_1 \dots \alpha_T], \mathbf{y}) = \prod_{n=1}^{N} \frac{\#\text{n-grams} \in ([\alpha_1 \dots \alpha_T] \cap \mathbf{y})}{\#\text{n-grams} \in [\alpha_1 \dots \alpha_T]}$$

No! Assuming N>1 and a word-by-word predictor.

Non-decomposability affects joint models: loss does not always decompose over the graphical model (Tarlow and Zemel, 2012 (http://www.cs.toronto.edu/~dtarlow/tarlow_zemel_aistats12.pdf))

Even F-score for binary classification is non-decomposable (Narasimhan et al., 2015 (http://jmlr.org/proceedings/papers/v37/narasimhana15.pdf))!

Expert policy

Returns the best action at the current state b y looking at the gold standard assuming future actions are also optimal:

$$\alpha^\star = \pi^\star(S_t, \mathbf{y}) = \arg\min_{\alpha \in \mathcal{A}} L(S_t(\alpha, \pi^\star), \mathbf{y})$$

I	studied	in	London	with	Sebastian	Riedel
PRP	VBD	IN	NNP	IN	NNP	NNP

PoS tagging: $\pi^\star(S_t, \mathbf{y}) = \pi^\star(S_t, [y_1 \ldots y_T]) = y_t$

Only available for the training data
(it cheats!): a human teacher
demonstrating how to perform the task

(http://www.salon.com/2016/10/06/what-makes-a-good-teacher-why-certifications-and-standards-dont-guarantee-quality-educators_partner/)

Expert policy

What action should $\pi^\star$ return?

I	studied	in	London	with	Sebastian	Riedel
O	O	O	B-LOC	O	B-PER	I-PER
O	O	O	B-LOC	O	O	O

Takes previous actions into account to be optimal (dynamic)

Finding the optimal action can be expensive; can be sub-optimal

Cost-sensitive classification

Classification: single correct label per-instance, e.g. B-PER

Multi-label classification: multiple correct labels per-instance, e.g. B-OR G, B-LOC

Cost-sensitive classification: each label has a cost, e.g.

O	B-PER	I-PER	B-LOC	I-LOC	B-ORG	I-LOC
1	2	2	0	1	0	1

Cost-sensitive classification

Imitation learning: learn a better model, i.e. action-predicting classifier b y imitating expert demonstrations

- learn that same mistakes are worse than others
- multiple actions can be optimal

Learning with costs:

- sample instances according to their cost to tr ain a classifier (Abe et al., 2004 (http://www.hunch.net/~jl/projects/reductions/mc2/p542-Abe.pdf))
- adjust the updates according the cost in error-driv en learning (Crammer et al. 2006 (http://jmlr.csail.mit.edu/papers/volume7/crammer06a/crammer06a.pdf))