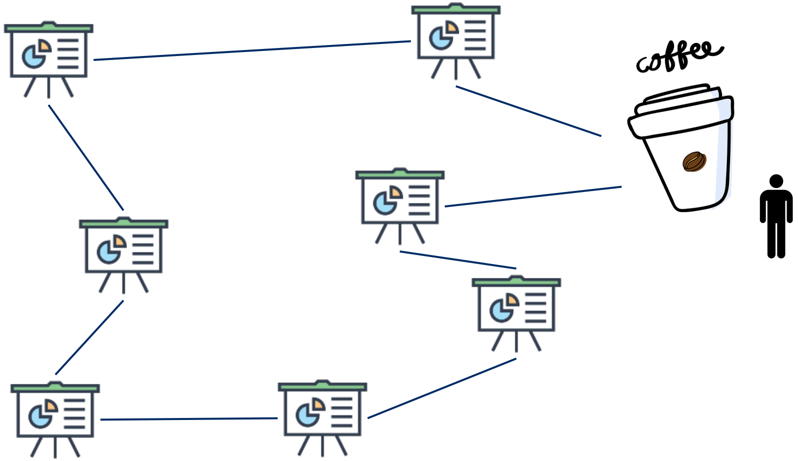
Attention, Learn to Solve Routing Problems!

Wouter Kool, Herke van Hoof, Max Welling

AMLAB Amsterdam Machine Learning Lab

ORTEC OPTIMIZE YOUR WORLD

University of Amsterdam



Travelling Scientist Problem (TSP)

International Conference on Learning Representations 2019

Published as a conference paper at ICLR 2019

ATTENTION, LEARN TO SOLVE ROUTING PROBLEMS!

Wouter Kool
University of Amsterdam
w.w.m.kool@uva.nl

Herke van Hoof
University of Amsterdam
h.c.vanhoof@uva.nl

Max Welling
University of Amsterdam
m.welling@uva.nl

TSP* is (NP-)hard!

* Travelling Salesman Problem (TSP)

What does it mean?

Finding *optimal* solutions for *all* problem instances

* unless P = NP

Finding *acceptable* solutions for *relevant* problem instances

We use HEURISTICS

Can be seen as ‘rules of thumb’

‘next location should be nearby’

Crafting of heuristics is similar to feature engineering

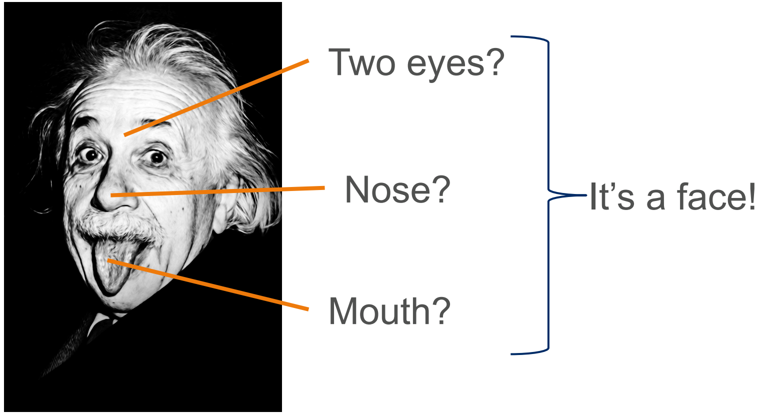
HARD
WORK

Computer Vision Features
(SIFT, etc.)

Feature engineering
- Needs expert knowledge
- Time consuming hand-tuning

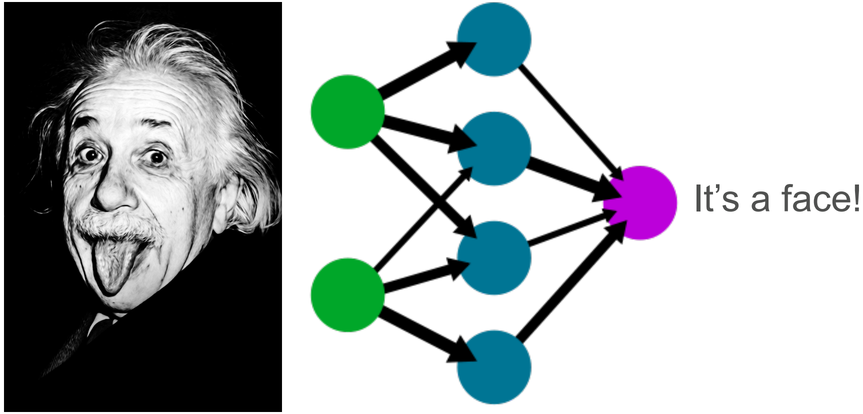
So what do we do?

Crafting of heuristics is similar to feature engineering



Traditional approach

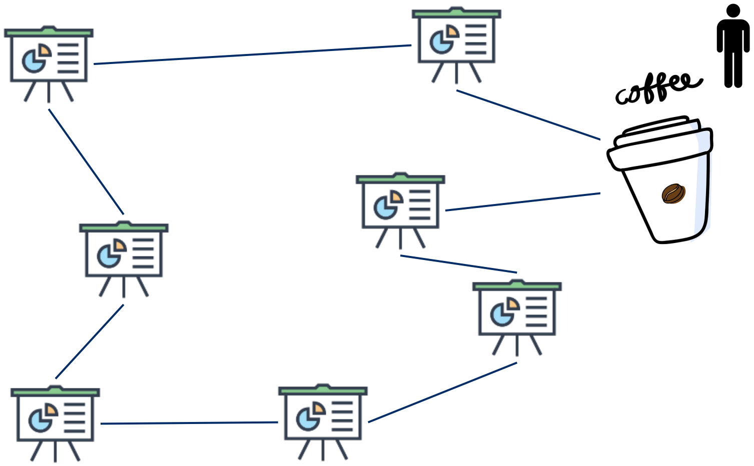
Feature engineering



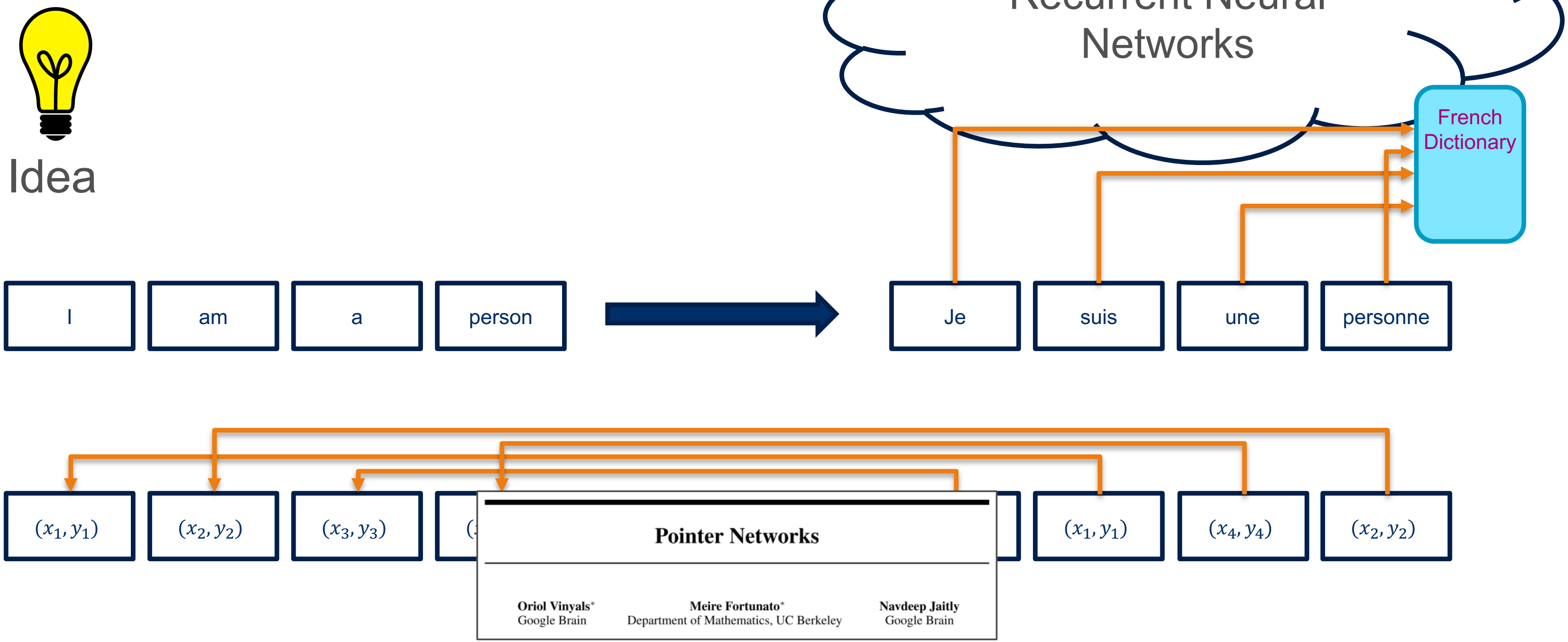
Deep Learning

No feature engineering

Back to our problem



‘Translate’ problem into solution



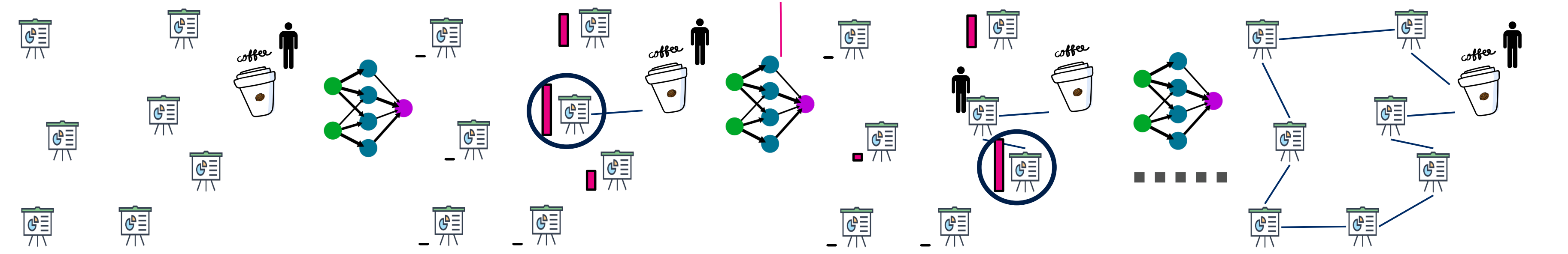
ORTEC

How does that work?

Instance $s = ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$

Model $p_{\boldsymbol{\theta}}(\pi_t | s, \pi_{<t}) = p_{\boldsymbol{\theta}}(\text{next node} \mid \text{partial tour})$

Solution $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots)$ with length $L(\boldsymbol{\pi})$



Sample $\pi_1 \sim p_{\boldsymbol{\theta}}(\pi_1 | s)$

Sample $\pi_2 \sim p_{\boldsymbol{\theta}}(\pi_2 | s, \pi_1)$

Sample $\pi_t \sim p_{\boldsymbol{\theta}}(\pi_t | s, \pi_{<t})$

How to optimize $\boldsymbol{\theta}$?

Randomized algorithm with expected cost:

$$E_{p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|s)}[L(\boldsymbol{\pi})]$$

NEURAL COMBINATORIAL OPTIMIZATION WITH REINFORCEMENT LEARNING

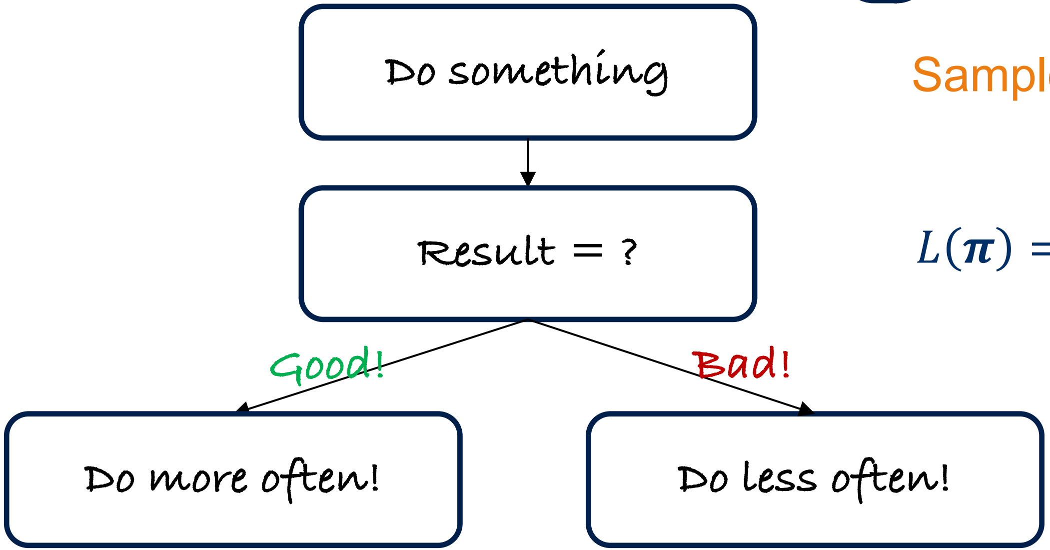
Irwan Bello*, Hieu Pham*, Quoc V. Le, Mohammad Norouzi, Samy Bengio
Google Brain
{ibello,hyhieu,qvl,mnorouzi,bengio}@google.com

REINFORCE (for dummies)

Repeat

Sample $\boldsymbol{\pi} \sim p_{\boldsymbol{\theta}}(\cdot\,|s)$

$L(\boldsymbol{\pi}) = 7.43$



Increase $p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|s)$

Decrease $p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|s)$

We need a *baseline* to compare against: *rollout* earlier model

$\boldsymbol{\pi}^{bl} \sim p_{\boldsymbol{\theta}^{bl}}(\cdot\,|s)$ **(greedy!)**

$L(\boldsymbol{\pi}^{bl}) = 6.89$

What's the model architecture?

$$p_{\theta}(\pi_t | s, \pi_{<t})$$

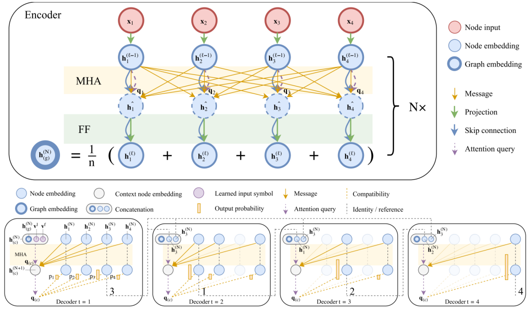
Read the paper…

ATTENTION, LEARN TO SOLVE ROUTING PROBLEMS!

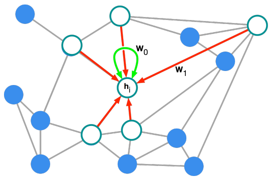
Wouter Kool
University of Amsterdam
w.w.m.kool@uva.nl

Herke van Hoof
University of Amsterdam
h.c.vanhoof@uva.nl

Max Welling
University of Amsterdam
m.welling@uva.nl



Graph convolutions



+

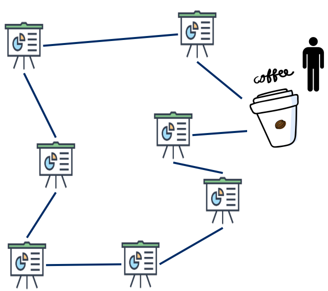
Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

ORTEC

Experiments

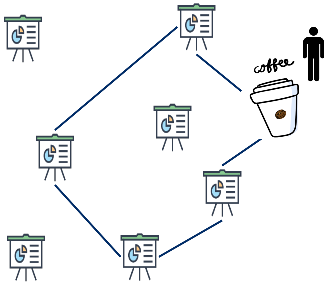
Travelling Salesman Problem (TSP)



Minimize length

Visit all nodes

Orienteering Problem (OP)



Maximize total prize

Max length constraint

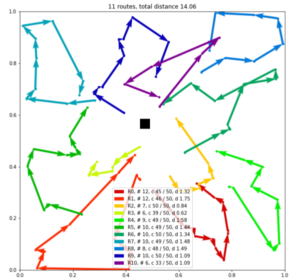
(Stochastic) Prize Collecting TSP ((S)PCTSP)

Minimize length +

penalties of unvisited nodes

Collect minimum total prize

Vehicle Routing Problem (VRP)



Minimize length

Visit all nodes

Total route demand must fit vehicle capacity

Train for each problem, *same hyperparameters*!

Results Attention Model + Rollout Baseline

- Improves over classical heuristics!
- Improves over prior learned heuristics!
  - Attention Model improves
  - Rollout helps significantly
- Gets close to single-purpose SOTA (20 to 100 nodes)!
  - TSP 0.34% to 4.53% (greedy)
  - TSP 0.08% to 2.26% (best of 1280 samples)

The end!

- Learning algorithms for optimization problems is promising

- Especially for less well-studied problems

- Can deal with uncertainty, specialize to data distribution, etc.

Thank you for your attention!