

# Introduction to natural language processing

Marco Kuhlmann

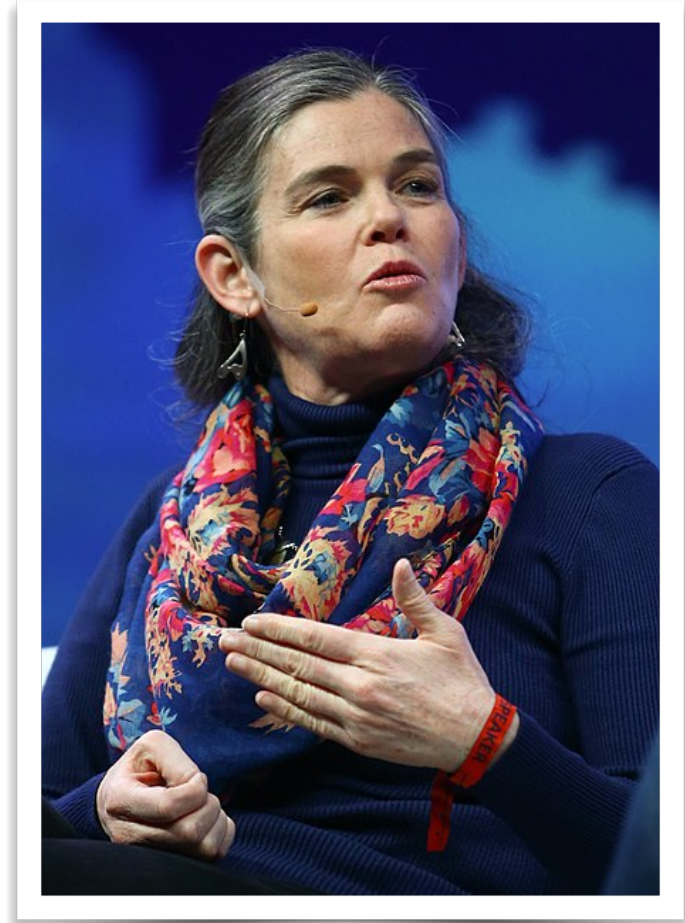
Department of Computer and Information Science

# What is natural language processing?

- **Natural language processing** develops methods for making human language accessible to computers.
- Some well-known example applications are intelligent search engines, machine translation, and dialogue systems.
- These diverse applications are based on a common set of ideas from algorithms, machine learning, and other disciplines.

# JEOPARDY!

This Stanford University alumna co-founded educational technology company Coursera.

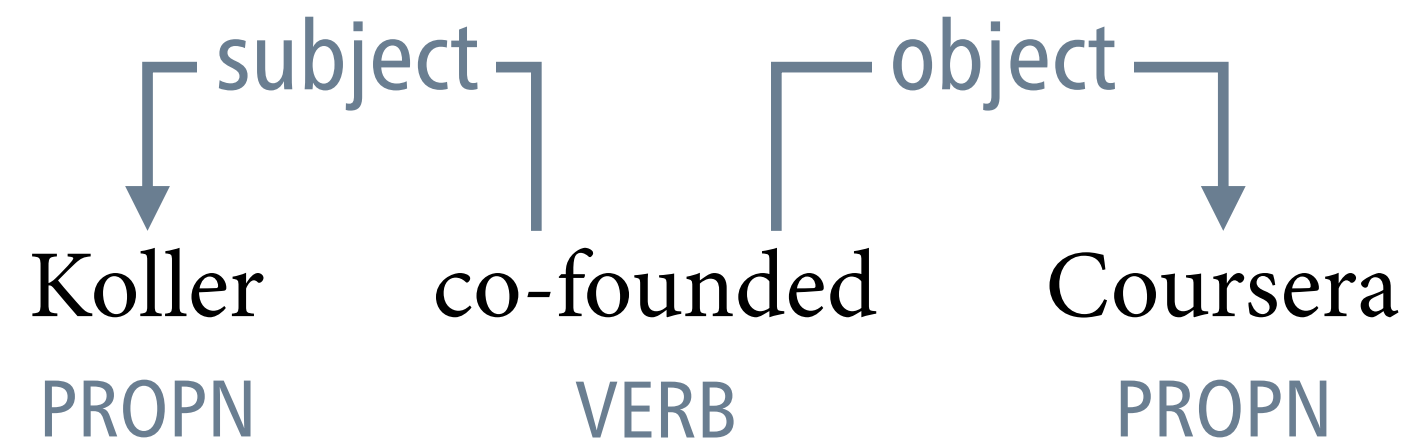


Collision Conf, CC BY 2.0, via Wikimedia Commons

[SPARQL query against DBPedia](#)

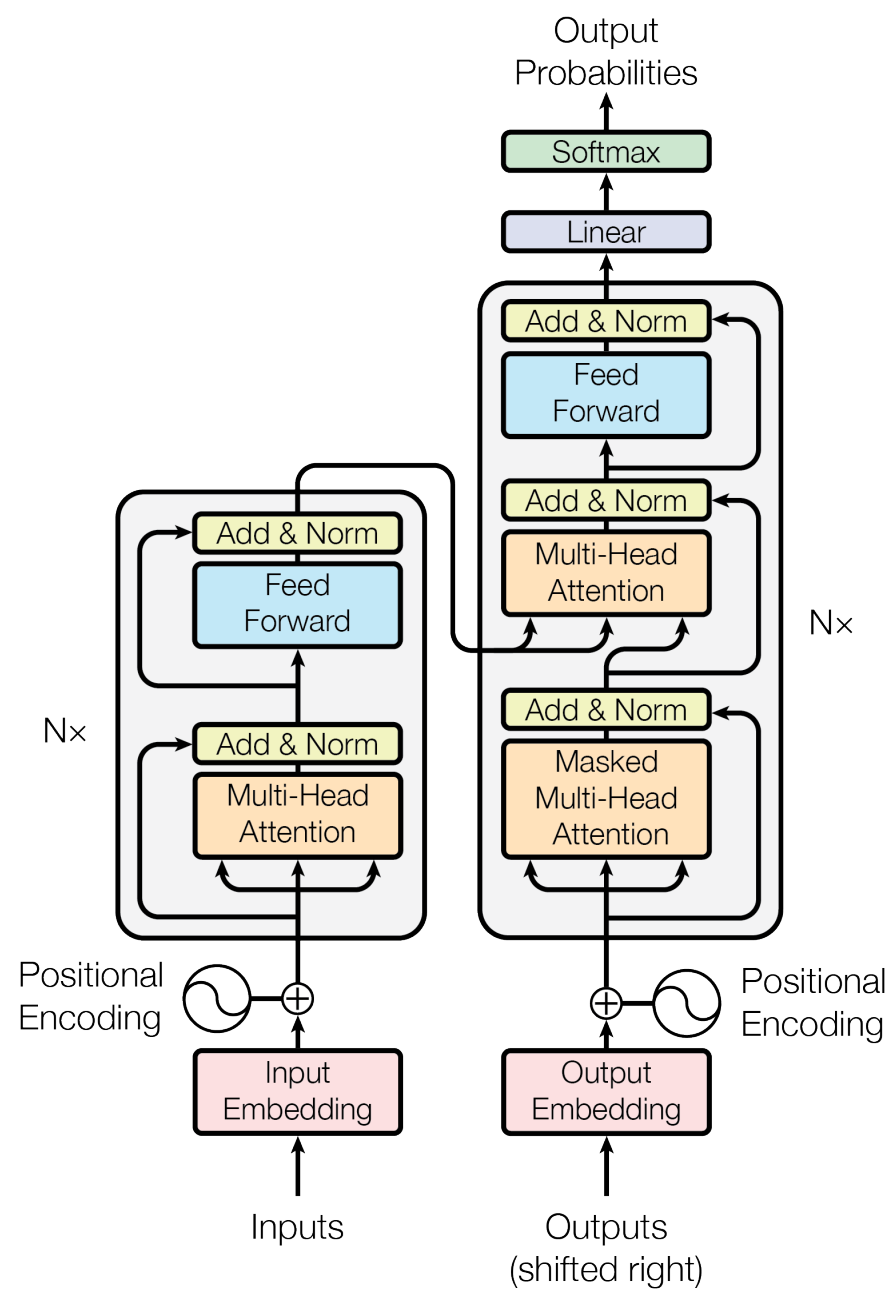
```
SELECT DISTINCT ?x WHERE {  
  ?x dbp:education dbr:Stanford_University.  
  dbr:Coursera dbp:founder ?x.  
}
```

# General-purpose linguistic representations

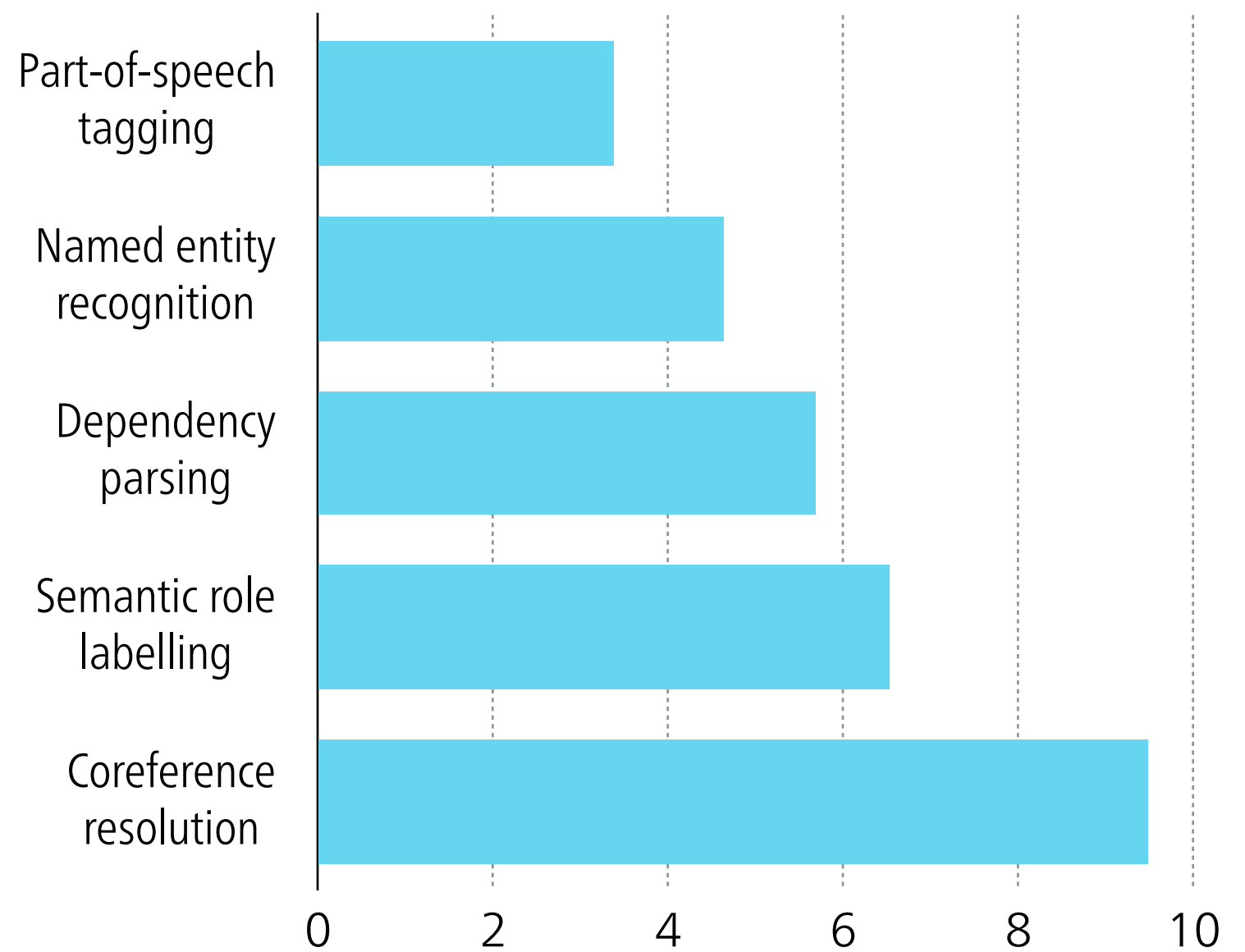


dbr:Coursera   dbr:founder   dbr:Daphne\_Koller

# 'Natural language processing from scratch'



[Vaswani et al. \(2017\)](#)



[Tenney et al. \(2019\)](#)

# Two paradigms

- **Linguistic knowledge**

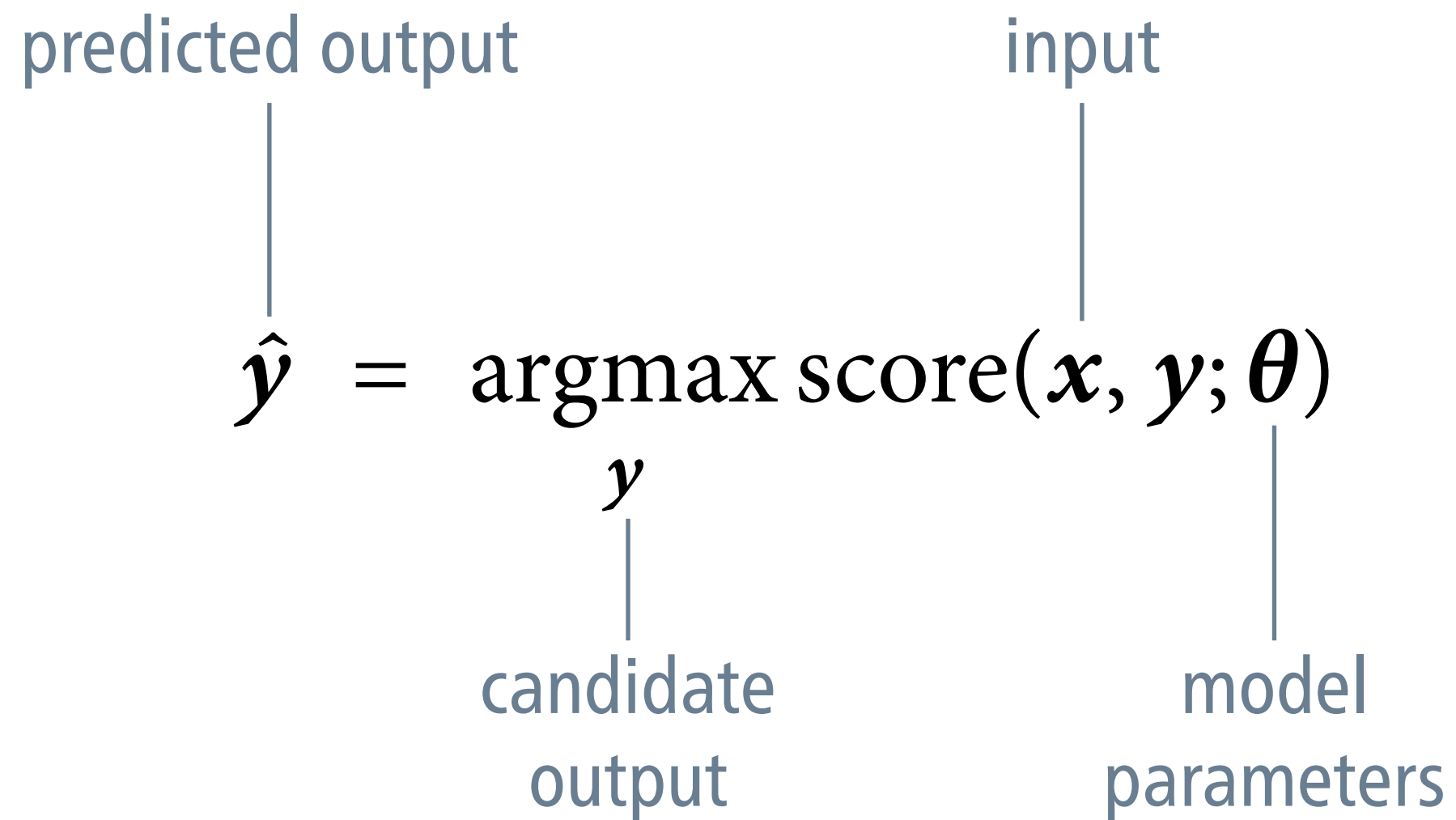
Build pipelines of modular components that produce general-purpose representations grounded in linguistic knowledge.

parts-of-speech, dependency trees, ...

- **Deep learning**

Train end-to-end neural networks that directly transmute raw text into whatever structure the desired application requires.

# Search and learning



# Search and learning

- **Search module**

The search module is responsible for finding the candidate output  $y$  with the highest score relative to the input  $x$ .

requires efficient algorithms

- **Learning module**

The learning module is responsible for finding the model parameters  $\theta$  that maximise the predictive performance.

for example, using supervised machine learning



# Language is special

- Unlike images or audio, text data is fundamentally discrete, with meaning created by combinatorial arrangement.
- Even though text appears as a sequence, machine learning methods must account for its implicit hierarchical structure.
- The distribution of linguistic elements follows a power law – algorithms must be robust to unobserved events.