Sequence labelling with global search

Marco Kuhlmann
Department of Computer and Information Science

High-level approaches

- **Local search**

  Cast sequence labelling as a sequence of classification problems: predict the label for each position in the input sequence.

- **Global search**

  Cast sequence labelling as a combinatorial optimisation problem over the full set of candidate sequences.

  requires specialised algorithms

Sequence labelling tasks

sequence of labels

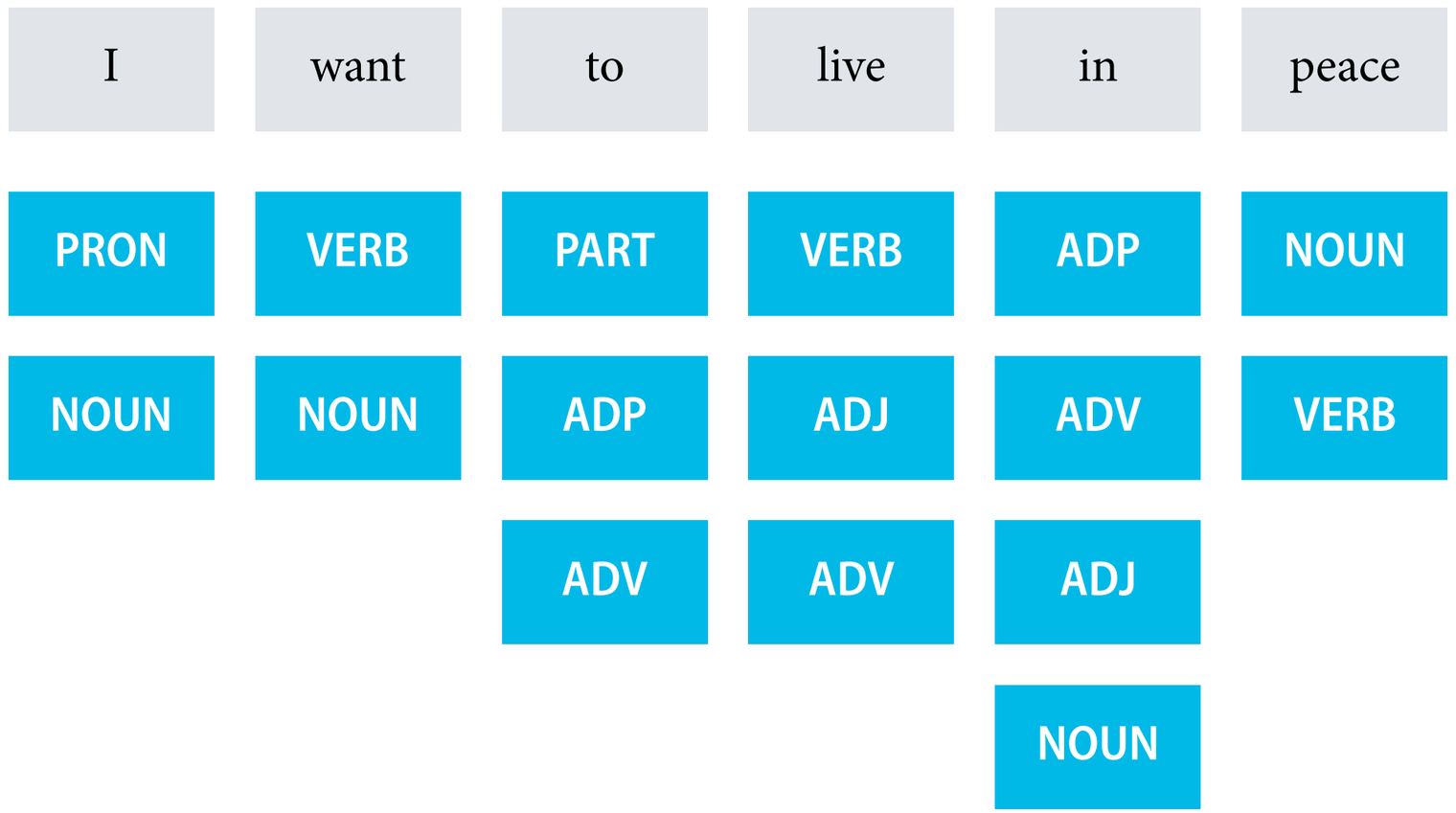
sequence of words

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\ \operatorname{score}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$$

candidate label sequences

model parameters

Ambiguity causes combinatorial explosion



288 possible tag sequences

Dealing with combinatorial explosion

- The number of candidate sequences is exponential in the length of the input sequence, so naive optimisation is doomed to fail.
- To make the search problem tractable, we will make assumptions about the scoring function.
- Then, in some cases, we can design special algorithms to solve the optimisation problem in polynomial time.

First-order factorized scoring function

candidate output sequence

sequence length

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \sum_{i=1}^{L} \text{score}^{e}(\boldsymbol{x}, y_i; \boldsymbol{\theta}) + \sum_{i=1}^{L} \text{score}^{t}(\boldsymbol{x}, y_{i-1}, y_i; \boldsymbol{\theta})$$

input sequence

score for a single label

emission score

score for a pair of labels

transition score

Maximum Entropy Markov Model (MEMM)

sequence of labels

sequence of words

$$P(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{L} \frac{\exp(\text{score}(\boldsymbol{x}, y_{i-1}, y_i))}{\sum_{y'} \exp(\text{score}(\boldsymbol{x}, y_{i-1}, y'))}$$

candidate label

Algorithmic problems

- **Decoding:** For a trained model, we want to find the most probable label sequence $\boldsymbol{y}$, given the input sequence $\boldsymbol{x}$.

  involves the search over exponentially many candidate sequences → Viterbi

- **Training:** To train a model, we want to minimise the negative log likelihood on gold-standard examples $((\boldsymbol{x}, y_{i-1}), y_i)$.

  standard softmax regression problem

- Search during decoding is global, search during training is local.

The label bias problem

Zhang and Teng (2021)

Sequence id	Sequence
1	ACA
2	ACA
3	AA
4	AAB
5	ABB
6	CCC

Global probability

$$\frac{2}{6}$$

$$\frac{1}{6}$$

First-order probability

How often is A followed by C?

How often is C followed by A?

$$\frac{5}{6} \cdot \frac{2}{6} \cdot \frac{2}{4} = \frac{5}{36}$$

How often does A start a sequence?

$$\frac{5}{6} \cdot \frac{2}{6} \cdot \frac{1}{1} = \frac{5}{18}$$

How often is A followed by B?

How often is B followed by B?

Conditional random field

Lafferty et al. (2001)

sequence of labels

sequence of words

$$P(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp(\text{score}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}'} \exp(\text{score}(\boldsymbol{x}, \boldsymbol{y}'))}$$

partition function

candidate label sequence

Algorithmic problems

- **Decoding:** For a trained model, we want to find the most probable label sequence $\boldsymbol{y}$, given the input sequence $\boldsymbol{x}$.

  involves the search over exponentially many candidate sequences → Viterbi

- **Training:** To train a model, we want to minimise the negative log likelihood on gold-standard examples ($\boldsymbol{x}$, $\boldsymbol{y}$).

  involves the search over exponentially many candidate sequences → Viterbi

- Search is global during both decoding and training.

Finding the highest-probability sequence

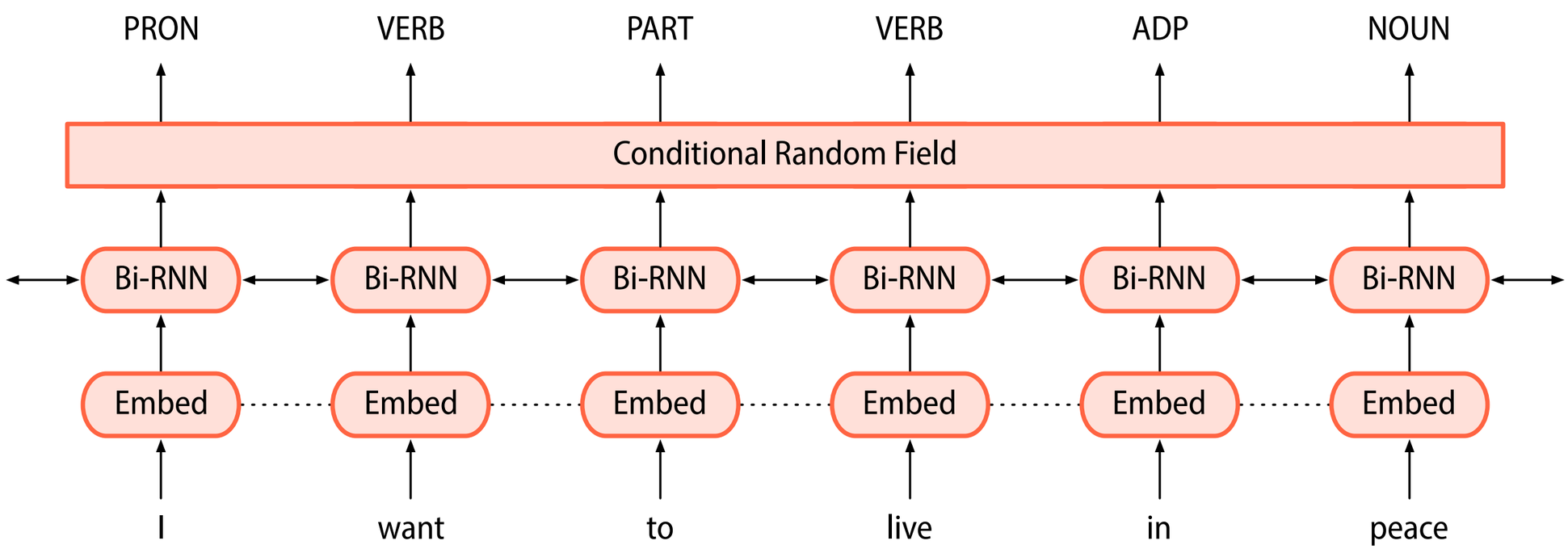
$$\arg\max_{\boldsymbol{y}} P(\boldsymbol{y} \mid \boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \frac{\exp\big(\text{score}(\boldsymbol{x}, \boldsymbol{y})\big)}{\sum_{\boldsymbol{y}'} \exp\big(\text{score}(\boldsymbol{x}, \boldsymbol{y}')\big)}$$

$$= \arg\max_{\boldsymbol{y}} \exp\big(\text{score}(\boldsymbol{x}, \boldsymbol{y})\big)$$

$$= \arg\max_{\boldsymbol{y}} \text{score}(\boldsymbol{x}, \boldsymbol{y})$$

We do not need the partition function to find the highest-scoring output.

Adding a CRF to a Bi-RNN model



PyTorch-compatible implementation available in AllenNLP