

# **Advanced NLP tasks**

# Contents

1. Named Entity Recognition (NER)
  - a. Part-of-Speech Tagging (POS)
  - b. Conditional Random Field (CRF)
  - c. Weakly Supervised NER
2. Sentiment Analysis
3. Natural Language Inference (NLI)
4. QuestionAnswering (QA)
  - a. Going further: LM as knowledge graphs
5. Exploit LLMs capacities: Chain-of-thoughts & In context learning

# Named Entity Recognition (NER)

# NER

Named entity recognition (NER), aims at identifying real-world entity mentions from texts, and classifying them into predefined types.

## Gold Dataset

Suxamethonium infusion rate and observed fasciculations.

Suxamethonium chloride (Sch) was administered i.v.

# NER

We wish to predict an output vector  $\mathbf{y} = (y_1, y_1, \dots, y_L)$ , of random variables, given an observed characteristic vector

$$\mathbf{x} = (x_1, x_2, \dots, x_L)$$

$\mathbf{y}$  takes its value from a list of  $N$  possible values.

# Part-of-Speech Tagging (POS)

POS is the process of mapping words in a text with a label corresponding to their grammatical class.

("He", "likes", "to", "drink", "tea"), → ("PERSONAL PRONOUN", "VERB", "TO", "VERB", "NOUN").

# Part-of-Speech Tagging (POS)

There are several levels of granularity.: using [the tag set for english](#)

("He", "likes", "to", "drink", "tea"), → ("PRP", "VBP", "TO", "VB", "NN").

# Conditional Random Field (CRF)

Knowing that language models are good at generating vector spaces to better represent words:

for each token in a sentence at position  $l$  we want to compute a probability  $p$  to belong to a class  $n$ .

$$p : f(\mathbf{x}, \theta)_l \mapsto ?$$

with  $p \in [0, 1]$



# Conditional Random Field (CRF)

Using the softmax function?

$$p : f(\mathbf{x}, \theta)_l \mapsto \frac{e^{f(\mathbf{x}, \theta)_l^{(n)}}}{\sum_{n'=1}^N e^{f(\mathbf{x}, \theta)_l^{(n')}}}$$

The probability given by the softmax function will not encode non-local dependencies!

# Conditional Random Field (CRF)

We need to take sequential decisions: what if we add transition scores into our softmax?

$$p : f(\mathbf{x}, \theta)_l \mapsto \frac{e^{f(\mathbf{x}, \theta)_l^{(n)} + t(y_l^{(n)}, y_{l-1})}}{\sum_{n'=1}^N e^{f(\mathbf{x}, \theta)_l^{(n')} + t(y_l^{(n')}, y_{l-1})}}$$

But this is the probability for one token to belong to a class, we want to compute the probability of a whole sequence of label at once...

# Conditional Random Field (CRF)

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}) &= \prod_{l=2}^L p(\mathbf{y} | f(\mathbf{x}, \theta)_l) \\ &= \prod_{l=2}^L \frac{e^{f(\mathbf{x}, \theta)_l^{(n)} + t(y_l^{(n)}, y_{l-1})}}{\sum_{n'=1}^N e^{f(\mathbf{x}, \theta)_l^{(n')} + t(y_l^{(n')}, y_{l-1})}} \end{aligned}$$

$$\begin{aligned}
P(\mathbf{y}|\mathbf{x}) &= \frac{\exp[\sum_{l=2}^L (f(\mathbf{x}, \theta)_l^{(n)} + t(y_l^{(n)}, y_{l-1}))]}{\sum_{n'=1}^N \exp[\sum_{l=2}^L (f(\mathbf{x}, \theta)_l^{(n')} + t(y_l^{(n')}, y_{l-1}))]} \\
&= \frac{\exp[\sum_{l=2}^L (U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]}{\sum_{n'=1}^N \exp[\sum_{l=2}^L (U(\mathbf{x}, y_l^{(n')}) + T(y_l^{(n')}, y_{l-1}))]} \\
&= \frac{\exp[\sum_{l=2}^L (U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]}{Z(\mathbf{x})}
\end{aligned}$$

# Conditional Random Field (CRF)

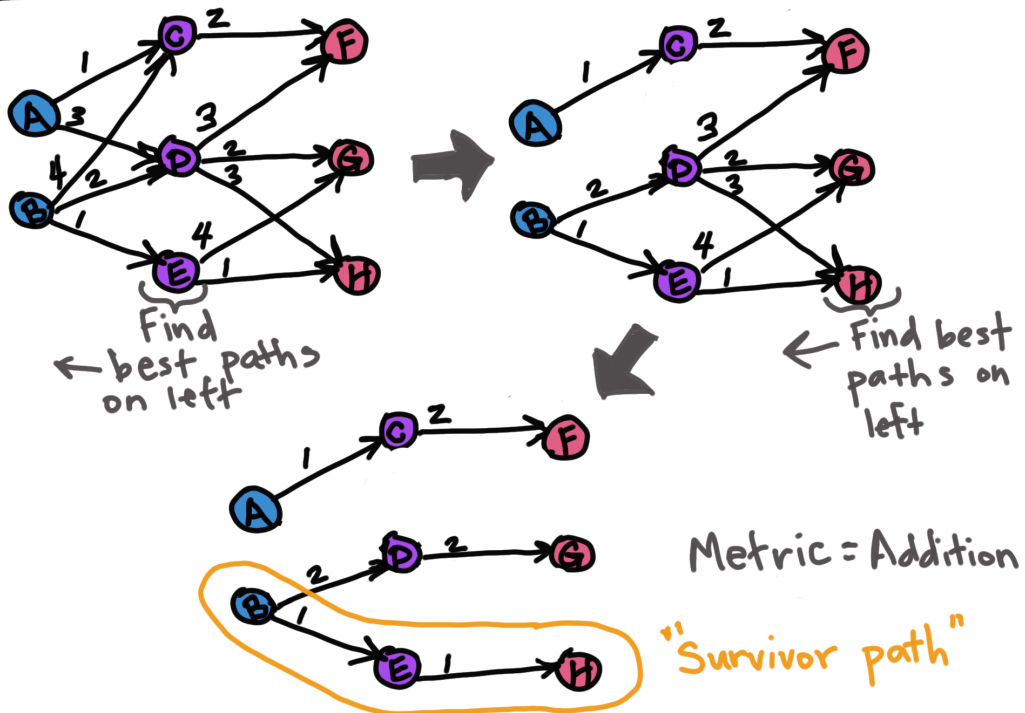
$Z(\mathbf{x})$  is commonly referred as the partition function. However, its not trivial to compute: we'll end up with a complexity of  $\mathcal{O}(N^L)$ .

Where  $N$  is the number of possible labels and  $L$  the sequence length.

How do we proceed?

# Conditional Random Field (CRF)

## Viterbi Algorithm



# Conditional Random Field (CRF)

## NER Transition Matrix

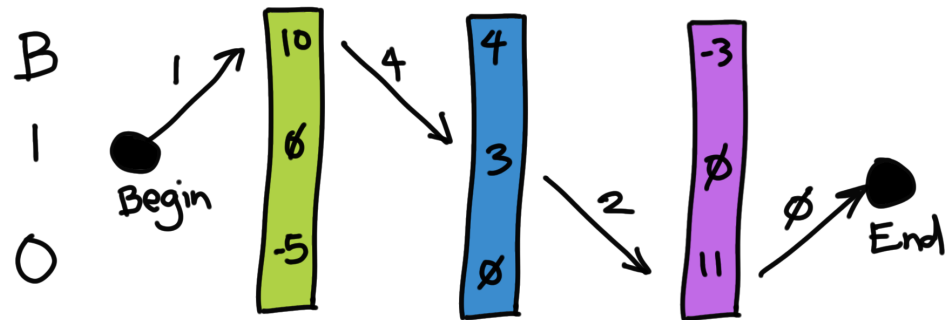
	B	I	O
B	$C(B \rightarrow B)$	$C(B \rightarrow I)$	$C(B \rightarrow O)$
I	$C(I \rightarrow B)$	$C(I \rightarrow I)$	$C(I \rightarrow O)$
O	$C(O \rightarrow B)$	$\infty$	$C(O \rightarrow O)$

$C$  = cost function

$\infty$  = wouldn't happen

# Conditional Random Field (CRF)

Linear-Chain CRF Decoded



Python comments help

Best path: B  $\rightarrow$  1  $\rightarrow$  0

Best score:  $1 + 10 + 4 + 3 + 2 + 11 + 0 = 31$



# Conditional Random Field (CRF)

Negative log-likelihood:

$$\begin{aligned}\mathcal{L} &= -\log(P(\mathbf{y}|\mathbf{x})) \\ &= -\log\left(\frac{\exp[\sum_{l=2}^L (U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]}{Z(\mathbf{x})}\right) \\ &= -[\log(\exp[\sum_{l=2}^L (U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]) - \log(Z(\mathbf{x}))] \\ &= \log(Z(\mathbf{x})) - \sum_{l=2}^L (U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))\end{aligned}$$

# Conditional Random Field (CRF)

There is an effective way to compute  $\log(Z(\mathbf{x}))$  with a complexity of  $\mathcal{O}(L)$  using [the Log-Sum-Exp trick](#).

$$\sum_{n'=1}^N \exp\left[\sum_{l=2}^L (U(\mathbf{x}, y_l^{(n')}) + T(y_l^{(n')}, y_{l-1}))\right]$$

# Weakly Supervised NER

What if we don't have annotated data?

1. Open-domain -> use Wikipedia or any anthology-based database.

# Questions?

# References

- [1] He, H. (2023, July 9). Robust Natural Language Understanding.
- [2] Singla, S., & Feizi, S. (2021). Causal imagenet: How to discover spurious features in deep learning. arXiv preprint arXiv:2110.04301, 23.
- [3] Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., & Liang, P. S. (2019). Unlabeled data improves adversarial robustness. Advances in neural information processing systems, 32.

[4] [Pretrained Transformers Improve Out-of-Distribution Robustness](#)

(Hendrycks et al., ACL 2020)

[5] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners.

Advances in neural information processing systems, 33, 1877-1901.

[6] Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021, July).

Calibrate before use: Improving few-shot performance of language models. In International Conference on Machine Learning (pp. 12697-12706). PMLR.