Deep Learning Applications in Natural Language Processing

Jindřich Libovický

April 9, 2025

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Today's Learning Outcomes

After this lecture you should be able to ...

1. Tell how various NLP tasks can be **formulated** as a **sequence labeling** problem
2. Reason about **problems with benchmarking** in NLP (using Visual Question Answering and Answer Span Selection as an Example)
3. Describe how **Named Entity Recognition** and **Answer Span Selection** solved using pre-trained language representations

Outline

Named Entity Recognition

Answer Span Selection

Named Entity Recognition

Information Extraction

Information Extraction = Subfield of NLP

$\boxed{\text{Find } \textbf{who} \text{ did } \textbf{what} \text{ to } \textbf{whom}.}$

Named entity recognition (NER) is one of the tasks.

Named Entity Recognition: Example

The Mona Lisa Mona Lisa WORK OF ART is a 16th century 16th century DATE oil painting oil painting CONCEPT

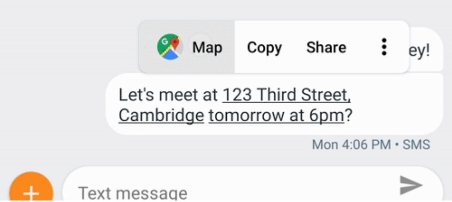
created by Leonardo Leonardo PERSON. It's held at the Louvre Louvre INSTITUTION

in Paris Paris LOCATION.

NER: application areas

- Part of information extraction pipeline
  - Entity linking (e.g., matching Wikipedia articles)
  - Coreference resolution

    Whom does pronoun “they” refer to?
    Who is “the president” in a text?
- Indexing text for search
- Direct use in smart devices



NER used to create links in text to different apps.

Image source: Google AI Blog. https://ai.googleblog.com/2018/08/the-machine-learning-behind-android.html

Entity Types

Different entity recognizers use different sets. Example:

`Person`	Names of people.
`PersonType`	Job types or roles held by a person.
`Location`	Natural and human-made landmarks, structures, geographical features, and geopolitical entities
`Organization`	Companies, political groups, musical bands, sport clubs, government bodies, and public organizations.
`Event`	Historical, social, and naturally occurring events.
`Product`	Physical objects of various categories.
`Skill`	A capability, skill, or expertise.
`Address`	Full mailing addresses.
`Phone`	number Phone numbers.
`Email`	Email addresses.
`URL`	URLs to websites.
`IP`	Network IP addresses.
`DateTime`	Dates and times of day.
`Quantity`	Numerical measurements and units.

List from Microsoft Text Analytics API (`https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/named-entity-types`)

NER as Sequence Labeling

- Assign each token with a tag saying what entity it belongs to
- Different tagging schemes

IOB Scheme

`I` — Token is inside an entity.
`O` — Token is outside an entity.
`B` — Token is the beginning of an entity.

Lance A. Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In David Yarowsky and Kenneth Church, editors, *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995*, 1995. URL `https://www.aclweb.org/anthology/W95-0107/`

BILUO Scheme ← usually better

`B` — Token is the beginning of a multi-token entity.
`I` — Token is inside a multi-token entity.
`L` — Token is the last token of a multi-token entity.
`U` — Token is a single-token unit entity.
`O` — Token is outside an entity.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W09-1119`

IOB Example

A sentence with 2 named entities:

		Quantity				Person			
There	are	over	1000	compositions	by	Johan	Sebastian	Bach	.
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
O	O	B-QUANT	I-QUANT	O	O	B-PERSON	I-PERSON	I-PERSON	O

Special B and I tags for each of the entity types.

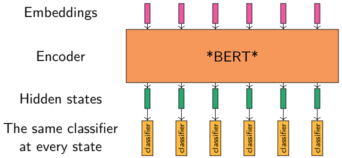
Deep learning solution

As usual ...

1. Embed input tokens as vectors
2. Contextualize the input embeddings using RNN/Transformer
3. Apply a classifier over each of the hidden states to predict the label

Current best solution:

Pre-trained Transformer (BERT) + finetuning for sequence labeling



Evaluation

Precision

$$P = \frac{\#\ \text{words correctly assigned to entities}}{\#\ \text{words in all } \textbf{detected} \text{ entities}}$$

Interpretation: How correct the system output is.

Recall

$$R = \frac{\#\ \text{words correctly assigned to entities}}{\#\ \text{words in all } \textbf{ground-truth} \text{ entities}}$$

Interpretation: How well the are the "real" entities covered.

F-Score

Harmonic mean of the previous two:

$$F_1 = \frac{2PR}{P + R}$$

Reasonable numbers are >90% on standard datasets.

Adding Conditional Random Field

- Standard tagging: conditional independence assumption
  i.e., given the hidden states all predictions are independent
- Tags have their internal grammar
  e.g., `I-PERSON` can only follow `I-PERSON` or `B-PERSON`
- If the model is unsure about the tag, it can lead to inconsistent predictions

$\Rightarrow$ **Conditional Random Fields** (CRF) can help restrict the models to produce more consistent outputs.

Original model: John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proc. of the 18th ICML.*, pages 282–289. Morgan Kaufmann, 2001

Neural CRF: Trinh Minh Tri Do and Thierry Artières. Neural conditional random fields. In Yee Whye Teh and D. Mike Titterington, editors, *Proc. of the 13th International Conference on Artificial Intelligence and Statistics*, volume 9 of *JMLR Proceedings*, pages 177–184. JMLR.org, 2010

Neural CRF for NER: Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proc. NAACL-HLT*, pages 260–270. ACL, June 2016

Formal definition of the CRF

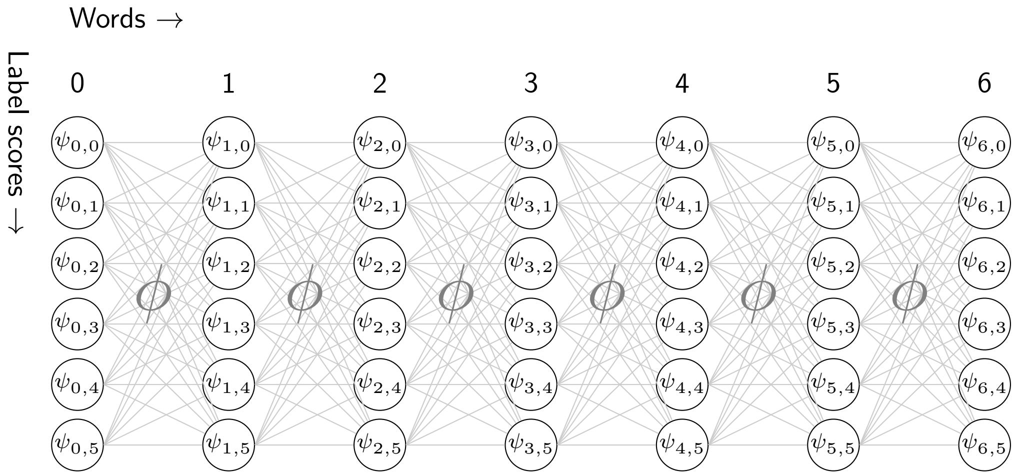
$$P(\mathbf{y}|\mathbf{h}) = \frac{1}{Z} \exp \left(\sum_{i=1}^{n} \psi(y_i|h_i) + \sum_{i=1}^{n-1} \phi(y_i, y_{i+1}) \right)$$

$\mathbf{y} = (y_1, \ldots, y_n) \sim$ output tags, $\mathbf{h} = (h_1, \ldots, h_n) \sim$ encoder states, Z is a normalizer, such that the probabilities sum up to 1

ψ	A linear projection of h_i, the same as in stantard labeling. No softmax here.	ϕ	A table of transitions scores between the tags $\approx$ grammar of the tags.

$P(\mathbf{y}|\mathbf{h})$ is a probability distribution over the space of **all possible tag sequences**, not single tags.

CRF as a Trellis



Best labeling = finding best path in the graph

Making CRF tractable

☺ Good news: Everything is differentiable

☹ Bad news: There are exponentially many possible tag sequence **y**.

1. Inference

Easy: we are only interested in the maximum $\Rightarrow$ throw away Z, throw away exponentiating, find maximum path in a trellis

2. Traning

Tricky, we need the normalizer:

$$Z = \sum_{\mathbf{y}} \exp\left(\sum_{i=1}^{n} \psi(y_i|h_i) + \sum_{i=1}^{n-1} \phi(y_i, y_{i+1})\right)$$

Simple algebraic tricks allow dynamic programming algorithm.

CRF Inference: Pseudocode

`psi` is a 2D array with labels scores, length $\times$ labels of length `T` with `n_labels` labels

`phi` label transition scores, shape: `n_labels` $\times$ `n_labels`

```
# 1. Search for the max-scoring path
scores = psi[0]
prev_pointers = []

for t range(T):
  prev = []; new_scores = []
  for i in range(n_labels):
    cost_to_i = scores + phi[:, i] + psi[t, i]
    prev.append(cost_to_i.argmax())
    new_scores.append(cost_to_i.max())
  prev_pointers.append(prev)
  scores = new_scores
```

```
# 2. Reverse-decode the path
#    indices
best_path = [scores.armax()]
for prev in
    reversed(prev_pointers):
  best_path.append(
    prev[best_path[-1]])
return reversed(best_path)
```

CRF Training: Compute the normalizer

Factor out the last step to get a recurrent equation:

$$
\begin{aligned}
\alpha_t(k) &= \psi_k + \log \sum_i \exp\left(\alpha_{t-1}(j) + \phi_{i,k}\right) \\
\alpha_0(k) &= \psi_0(k) \\
Z &= \log \sum_k \exp \alpha_T(k)
\end{aligned}
$$

There is an efficient implentation of log-sum-exp.

```
alphas = psi[0]
for t in range(1, T):
  new_alphas = []
  for i in ranage(n_labels):
    new_alpha.append(
      psi[t, i] +
      logsumexp(alpha[j] + phi[j, i]
          for j in range(n_lables)))
  alphas = new_alphas
return logsumexp(alphas)
```

Implementation in PyTorch

Package `pytorch-crf`

```
pip install pytorch-crf
```

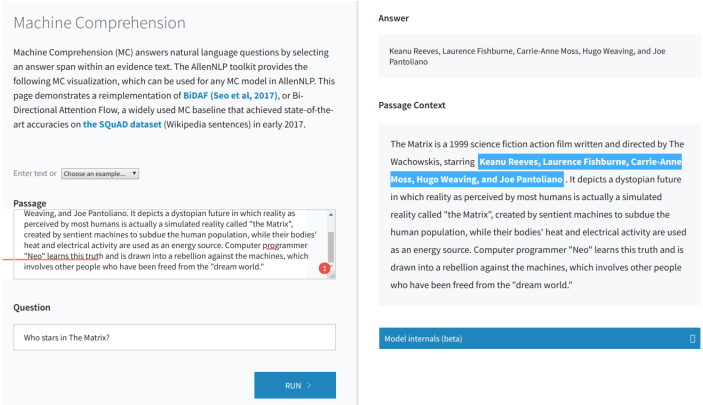
Initialize the model:

```
import torch
from torchcrf import CRF
num_tags = 5 # number of tags is 5
model = CRF(num_tags)
```

The modul expects the unnormalized tag scores as the input

Answer Span Selection

Task: Find an answer for a question given question in a coherent text.



`http://demo.allennlp.org/machine-comprehension`

Standard Dataset: SQuAD

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

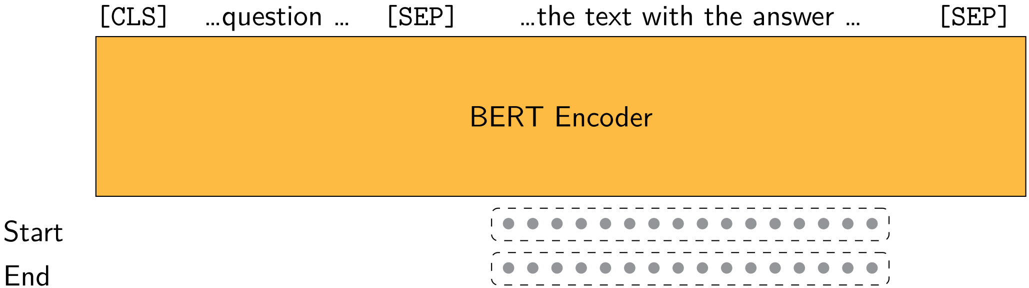
- best articles from Wikipedia, of reasonable size (23k paragraphs, 500 articles)
- crowd-sourced more than 100k question-answer pairs
- complex quality testing (which got estimate of single human doing the task)

`https://rajpurkar.github.io/SQuAD-explorer/explore/1.1/dev/`

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1264

The same thing with BERT

Just throw **everything into BERT**: both the text and the question.



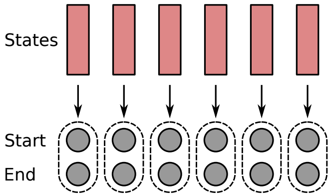
Assign start the start and end labels.

Output Layer

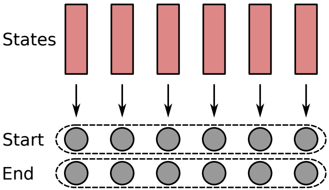
1. Start-token probabilities: project each state to scalar $\rightarrow$ apply softmax over the context
2. End-token the same
3. At the end select the most probable span

A difference from standard labeling: scores get normalized:

Standard: **per label**



Here: **over the entire text**



SQuAD Leaderboard

method	Exact Match	F1 Score
Human performacne	82.304	91.221
BiDAF trained from scratch	73.744	81.525
BERT	87.433	93.160
FPNet (Best in leaderboard; Feb 2021)	90.871	93.183

⚠ Any time an NLP model is **better than humans**, something is **wrong**.
Probably overfitting to specifics of the dataset. ⚠

Today's Learning Outcomes

After this lecture you should be able to ...

1. Tell how various NLP tasks can be **formulated** as a **sequence labeling** problem
2. Reason about **problems with benchmarking** in NLP (using Visual Question Answering and Answer Span Selection as an Example)
3. Describe how **Named Entity Recognition** and **Answer Span Selection** solved using pre-trained language representations

Summary

1. **Named Entity Recognition:** a labeling problem with more clever training objective
2. **Answer Span Selection:** showcasing the strength of Transformers, in the end labeling problem too