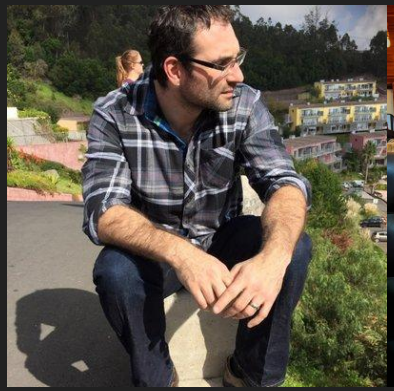
Constrained Decoding for Computationally Efficient Named Entity Recognition Taggers

Brian Lester, Daniel Pressel, Amy Hemmeter,
Sagnik Ray Choudhury, and Srinivas Bangalore

To appear in Findings of EMNLP 2020
Presented at SustaiNLP 2020

Collaborators





What are Taggers?

- A Sequence Transduction task

  $X = (x_1, x_2, \ldots, x_n)$

  $Y = (y_1, y_2, \ldots, y_n)$

- Additional constraint that the input and output sequences have the same length.

Tagger Tasks

- Token Level
  - Part of Speech Tagging
- Span Level
  - Named Entity Recognition
  - Slot Filling for Dialogue Systems

Why do we Want to Train Efficient Taggers?

We train a lot of taggers, in 3 months:

- CoNLL 2003: 342
- Ontonotes: 56
- Snips: 49
- WNUT: 34
- Internal: 788

Tagger Models

- Windowed Classifiers
- MEMMs
- BiLSTM-CRF
- Transformers

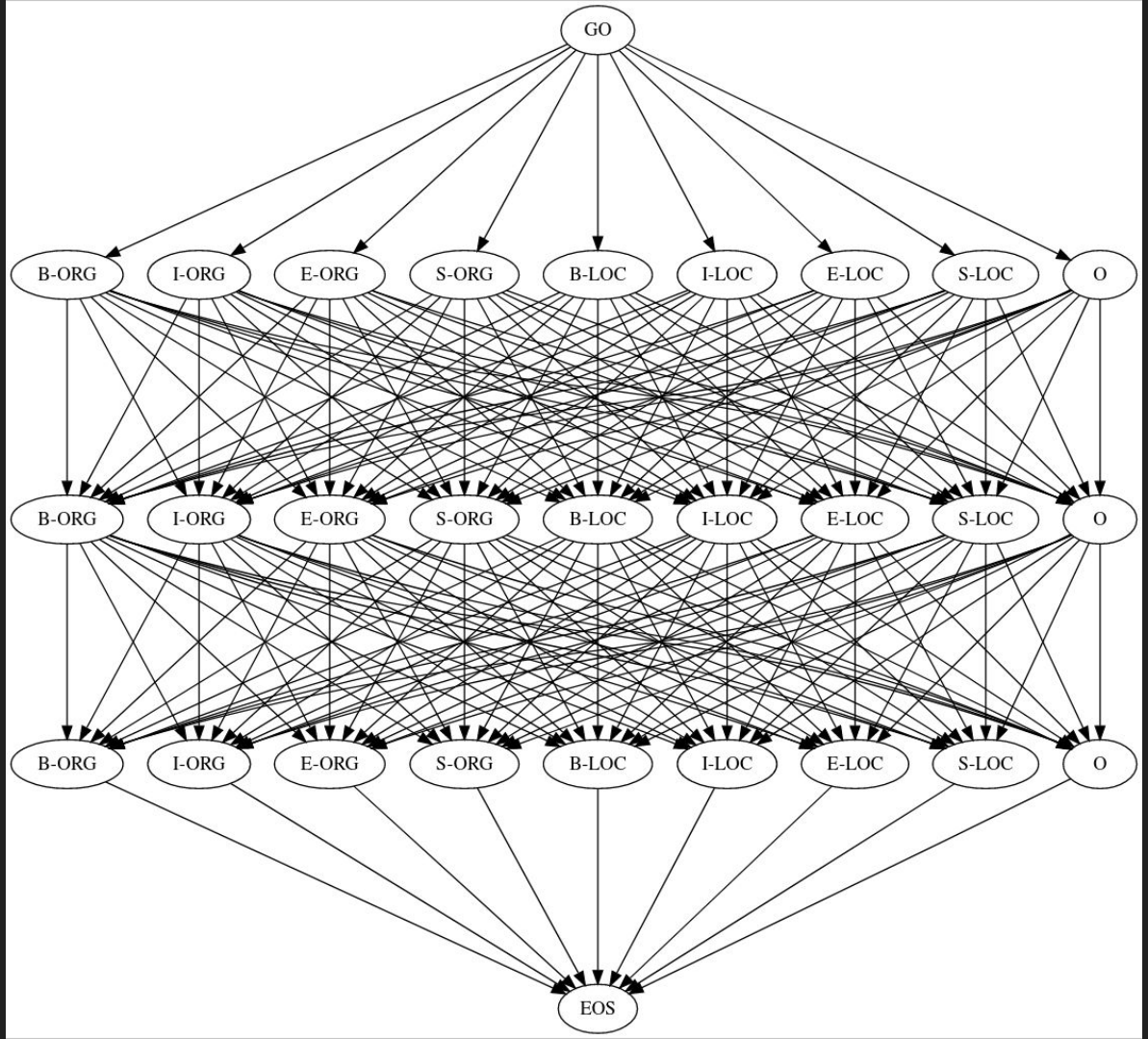
Greedy Taggers

- Make an independent decision at each timestep
- Your choice at t-1 doesn’t factor into your choice at t
- These taggers often have difficulty with global coherence
  - They change the types of entities in the middle of spans

Structured Tagger Inference

- We want to find the best over all tag sequence
- Not just the best tag for each token
- Enumerating and scoring each sequence would be intractable
- We use dynamic programming with the Viterbi Algorithm
- This generally involves emission scores, a distribution over labels for a given token, and transition scores, a distribution of transitions from one label to another.

Viterbi Decoding



Span Encoding

- For some tasks we need more than just a token label
- We want the whole phrase “Jack White” to be labeled as a single person, not each token to be labeled separately
- We keep the labels types like in tokens
- We add special prefixes to group tokens into spans

Span Encoding

B-PER

Each tag is made from two parts

- The second part is the type of entity it is. A person, location, etc.
- The first part is the function of this token in the span
  - B is the beginning of the span
  - I is inside of a span
  - E is the end of a span
  - S is a token that makes up the whole span
  - O is outside of a span

Span Encoding

```
Jack     B-PER
White    E-PER
was      O
born     O
in       O
Detroit  S-LOC
On       O
July     B-DATE
9th      I-DATE
1975     E-DATE
```

“Jack White” is a person

“Detroit” is a location

“July 9th 1975” is a date

Span Encoding Constraints

- The span encoding scheme imposes some rules
  - I and E must follow a token of the same type
  - B can only follow an O, E, or S
  - S cannot follow B or I

Constraints as Transition Parameters

What if instead of learning these transition scores we use these constraints to stop illegal moves?

Constraints as Transition Parameters

Our Method

- Train a Tagger with Cross Entropy Loss
- Create a mask based on the transition rules
  - A mapping from one label to another
  - Zero if the transition is legal
  - Negative infinity if it is illegal
- Use this mask as transition parameters in our CRF implementation

Results

Dataset	CRF Score	CD Score	Difference
CoNLL 2003	91.61	91.44	-0.03
WNUT-17	40.33	40.59	0.65
Snips	96.04	96.07	0.03
Ontonotes	87.43	86.13	-1.48
Internal Customer Service			0.21
Internal Automotive			-0.68
Internal Cyber Security			0.84
Internal NER			0.80

Analysis

Why did we only see this drop in Ontonotes?

Strictly Dominated Tokens

- Within a dataset a type can often have different labels assigned to different tokens
  - Kurdistan can be a `B-ORG`, `E-ORG`, or `E-LOC`

Strictly Dominated Tokens

- Within a dataset a type can often have different labels assigned to different tokens
  - Kurdistan can be a `B-ORG`, `E-ORG`, or `E-LOC`
- Previous tokens (and their labels) and our transition rules can help use make a decision

Strictly Dominated Tokens

- Within a dataset a type can often have different labels assigned to different tokens
  - Kurdistan can be a `B-ORG`, `E-ORG`, or `E-LOC`
- Previous tokens (and their labels) and our transition rules can help use make a decision
- What if the last token was `I-ORG`?

Strictly Dominated Tokens

- Within a dataset a type can often have different labels assigned to different tokens
  - Kurdistan can be a `B-ORG`, `E-ORG`, or `E-LOC`
- Previous tokens (and their labels) and our transition rules can help use make a decision
- What if the last token was `I-ORG`?
- What if the last token was `B-LOC`?

Easy First and Easy Last

- Once you make a decision your search space is vastly reduced.

Easy First and Easy Last

- Once you make a decision your search space is vastly reduced.
- Once you decide a token in `B-ORG` you know your next token is either `I-ORG` or `E-ORG`.

Easy First and Easy Last

- Once you make a decision your search space is vastly reduced.
- Once you decide a token in `B-ORG` you know your next token is either `I-ORG` or `E-ORG`.
- What about entities where the first token is always a `B-ORG`?

Easy First and Easy Last

- Once you make a decision your search space is vastly reduced.
- Once you decide a token in `B-ORG` you know your next token is either `I-ORG` or `E-ORG`.
- What about entities where the first token is always a `B-ORG`?
- Because of how Viterbi works the same ideas apply to entities where the last token always has a single label.

Efficiency

- Faster Training (51.2% of the time)

Efficiency

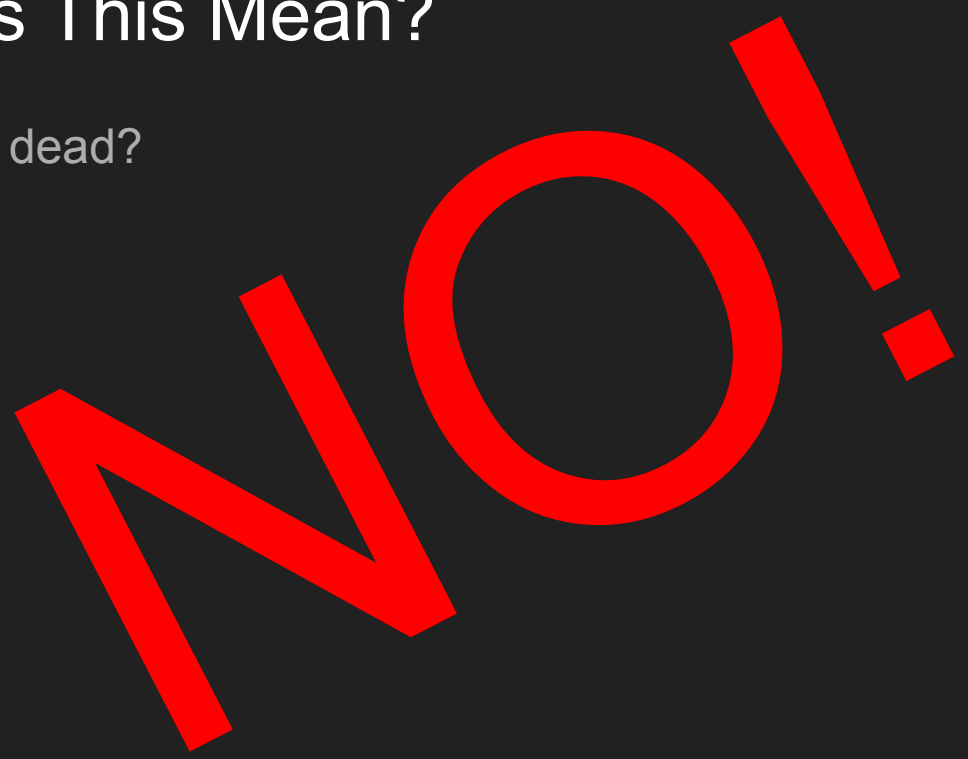
- Faster Training (51.2% of the time)
- 65% of the Carbon Emissions during training
    - It does draw 1.3 times the power

What Does This Mean?

- Structure is dead?

What Does This Mean?

- Structure is dead?



What Does This Mean?

- Structure is dead?
- Structure needs to evolve

Contact

- The work: https://github.com/blester125/constrained-decoding
- Me
  - Twitter: https://twitter.com/blester125
  - Web: https://blester125.com/