Named-Entity Recognition from Scratch with SpaCy

Benjamin Batorsky

About me

Data Science Consultant

PhD, Policy Analysis

https://benbatorsky.com/

This work:

MIT MANAGEMENT SLOAN SCHOOL

Food Supply Chain Analytics and Sensing Group

Global pilots of risk-based food safety testing technology

Overview of this presentation

1. Introduction to NER
    a. Why it is difficult and why it is important
2. Introduction to our use-case at MIT
3. Approaches to NER
4. NER in SpaCy
5. Using SpaCy's NER model architecture and results
6. Conclusion/Next steps

What is Named-Entity Recognition (NER)?

- Named-entity: A real-world named object (e.g. person, place, organization)
  - New York City is different than just an assembly of three words “new”, “york” and “city”
- Essentially two tasks:
  - Identify where in the text the entity occurs
  - Identify what type of entity it is
- Often accomplished through inventories or rule-based methods

On September 3, 2020, I spoke at PyData Edinburgh.

What are the named-entities here?

What is Named-Entity Recognition (NER)?

- Named-entity: A real-world named object (e.g. person, place, organization)
  - New York City is different than just an assembly of three words “new”, “york” and “city”
- Essentially two tasks:
  - Identify where in the text the entity occurs
  - Identify what type of entity it is
- Often accomplished through inventories or rule-based methods

On September 3, 2020 [DATE], I spoke at PyData [ORG] Edinburgh [LOC].

Also possibly PyData Edinburgh as an organization or an event

Why is NER difficult?

- Ambiguity about entity boundaries
- Good performance typically requires a lot of quality labelled data
  - CoNLL 2003 task: ~21k labelled sentences
  - OntoNotes 5.0: 1.4 M articles
- Task complexity
  - Each token needs one tag and one entity type, decision between X tags * Y types
- Performance evaluation
  - What if the method achieves partial match?
  - Are we interested in performance on new data? Out-of-inventory entities?

PyData [U-ORG (or B-ORG)]

Meetup [B-ORG (or I-ORG)]

Group [L-ORG]

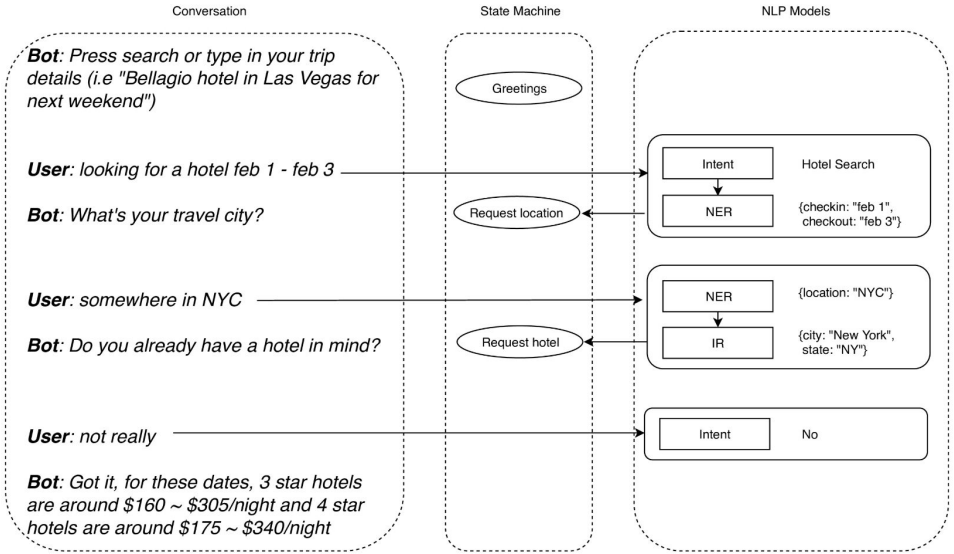
in [O]

Edinburgh [U-LOC]

CoNLL 2003: https://www.aclweb.org/anthology/W03-0419/
OntoNotes 5.0: https://catalog.ldc.upenn.edu/LDC2013T19

Why is NER important?

- Named-entities are fundamentally different from other tokens
  - “New York City” != “new” + “york” + “city”
- Having a “complete” inventory is extremely rare
- NER is part of many Machine Learning applications
  - Content recommendation/Search
  - Chatbots
  - Translation



https://www.groundai.com/project/real-world-conversational-ai-for-hotel-bookings/1

NER at Sloan's Food Supply Chain Analytics group

- Dataset: Full-text for 22k court cases
- Current NER approach:
  - Regular expressions using inventories
    - Inventories likely to be incomplete and spellings/mentions likely to vary
  - Manual review and hand-labelling
    - Extremely time-consuming
    - Needs to be repeated for new data/new questions
- Proposed approach:
  - Model-based NER
  - Compare versus current approach
    - Performance on new data
    - Performance on entities out-of-inventory

What are the regulatory agencies involved in food safety enforcement?

"China FDA brought a suit against…"

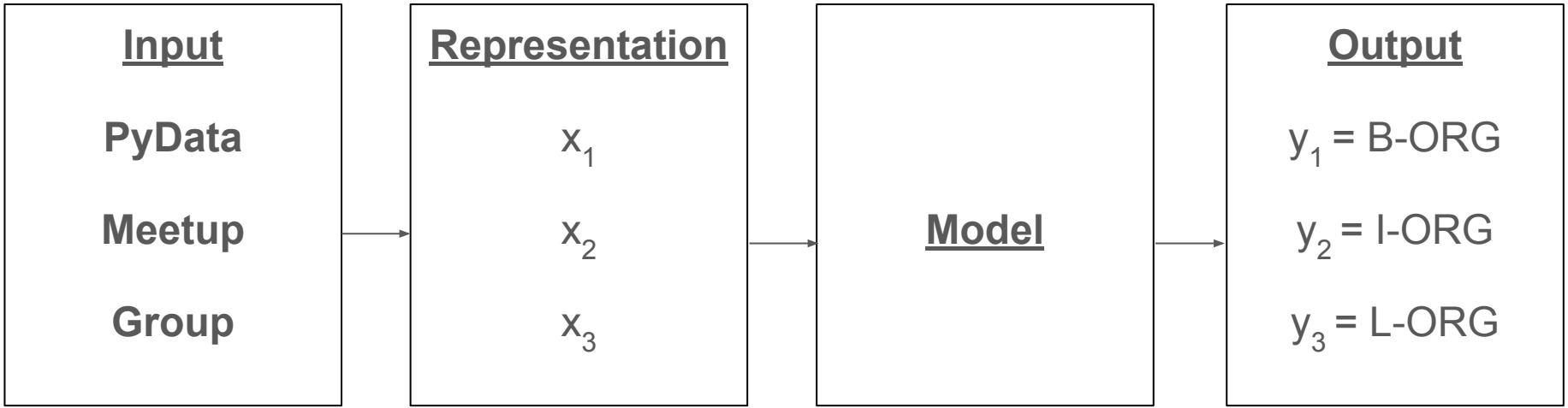
What types of products do they oversee?

"...for selling tainted pork products…"

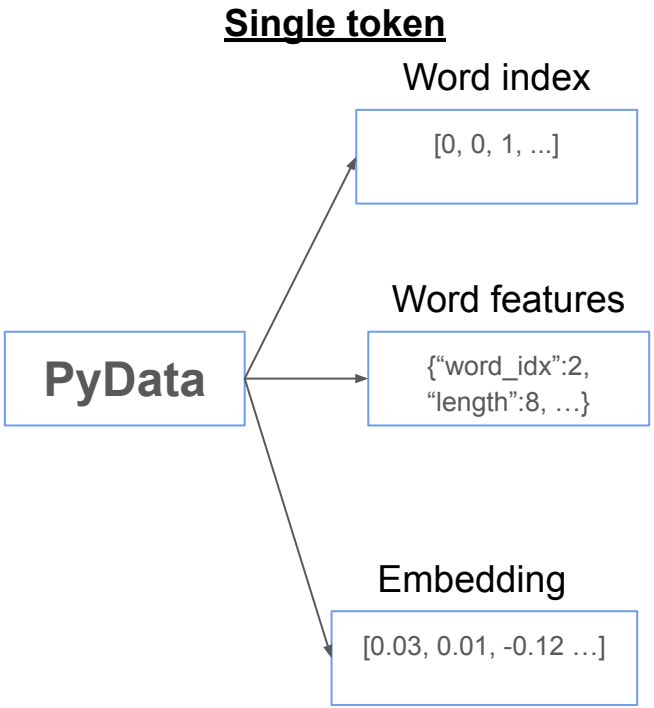
What is their jurisdiction?

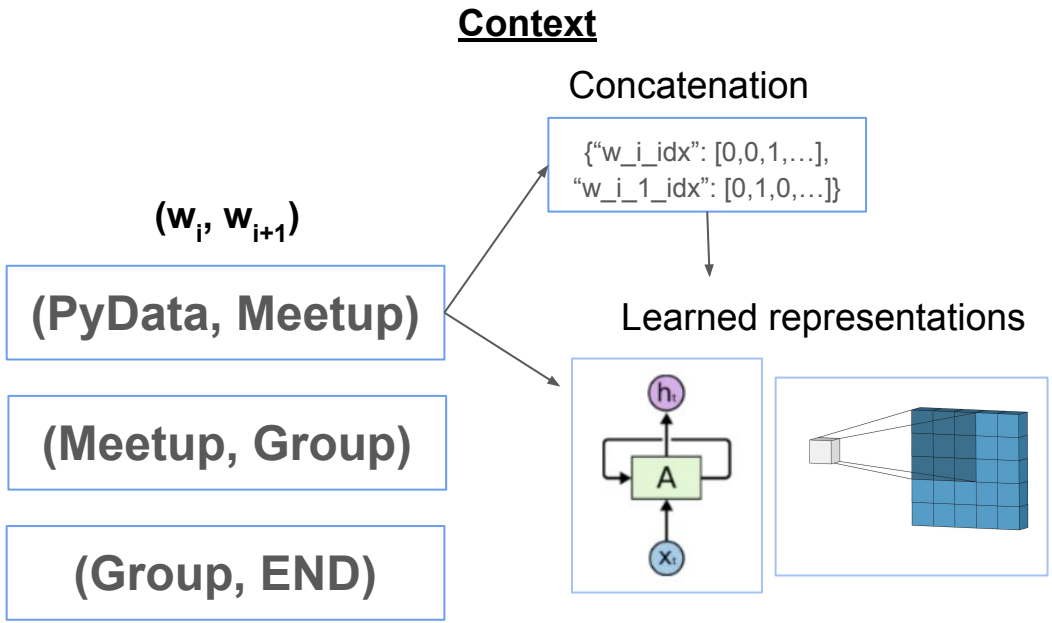
"...in Hangzhou Province."

Overview of model-based NER



Representation

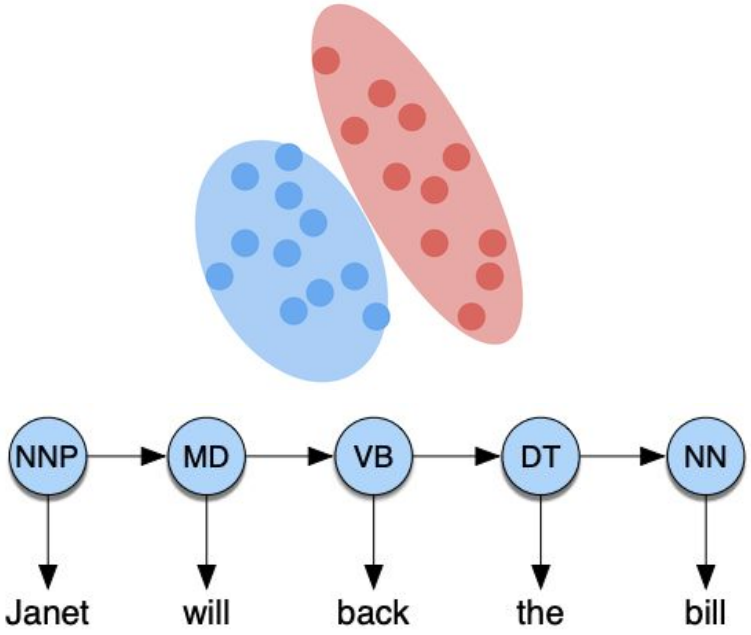




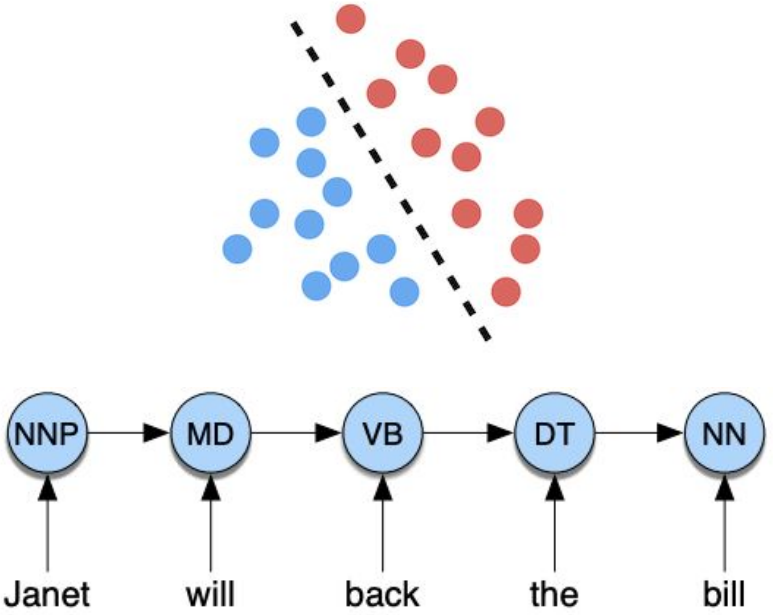
https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e
https://towardsdatascience.com/classifying-skin-lesions-with-convolutional-neural-networks-fc1302c60d54

Model

Generative (Naive Bayes, HMMs)



Discriminative (MEMMs, CRFs, Perceptron)

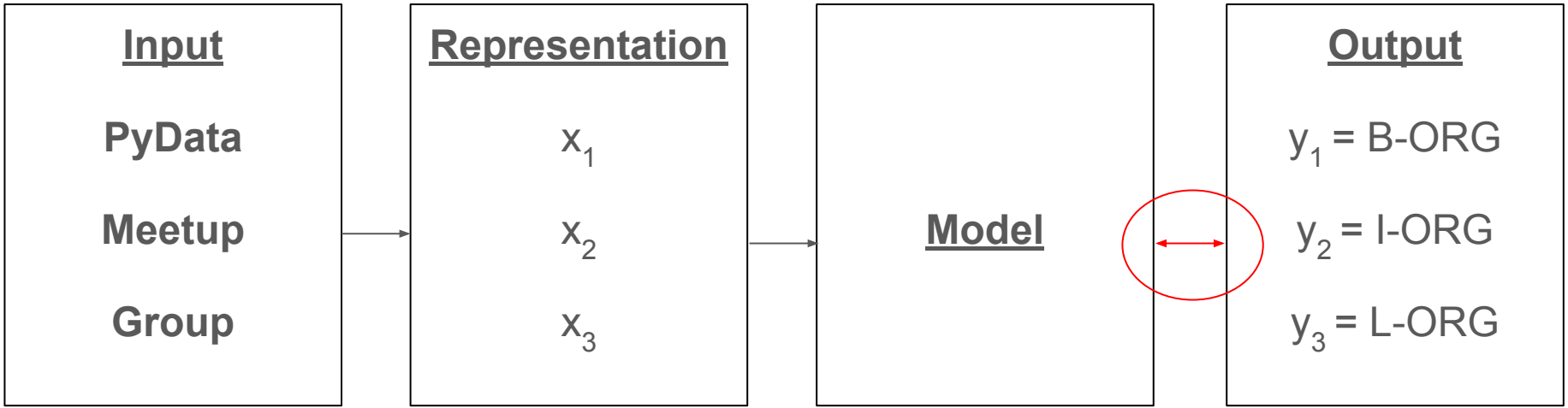


L Ratinov & D Roth (2009) Design Challenges and Misconceptions in Named Entity Recognition: https://www.aclweb.org/anthology/W09-1119.pdf

Caltech CS/CNS/EE 155 lecture HMM, MMEM and CRFs; https://www.youtube.com/watch?v=B1nl8fLgKMk

Images from: "Speech and Language Processing" Daniel Jurafsky & James H. Martin (https://web.stanford.edu/~jurafsky/slp3/8.pdf)

Overview of model-based NER



SpaCy: All-purpose NLP library in Python

- Uses “Language models” with tokenization, vocabulary, other pipeline elements
  - Documents->Spans->Tokens
- Models: Pipeline components either pre-trained or trainable via API
  - Part-of-speech
  - Categorization
  - NER
- No Chinese language model with NER available for SpaCy (core)

```
from spacy.lang.zh import Chinese
nlp = Chinese()
[(x, type(x)) for x in nlp('中国菜')]
```

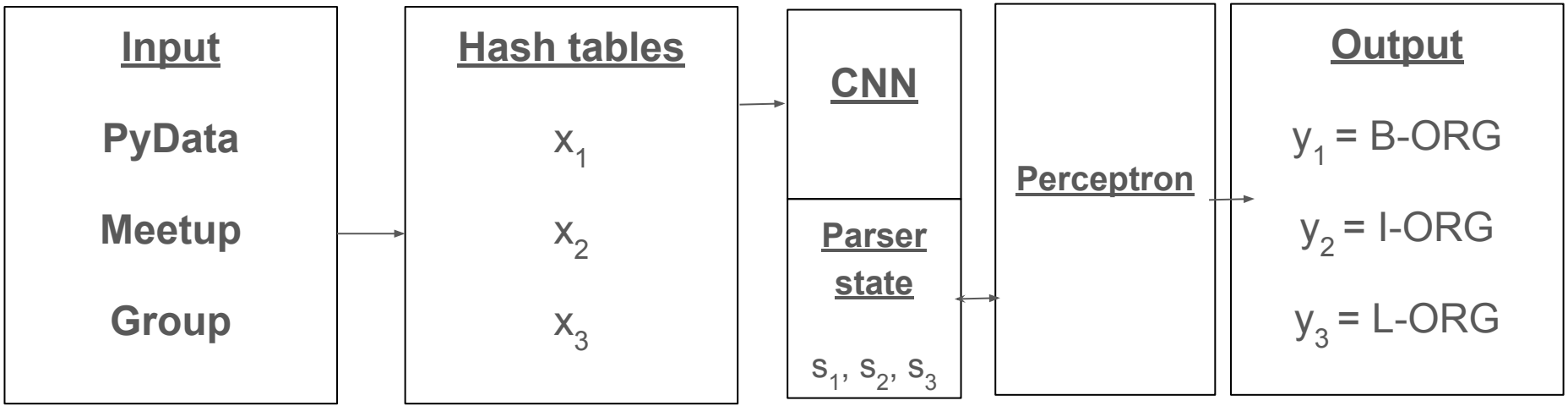
```
[(中国, spacy.tokens.token.Token), (菜, spacy.tokens.token.Token)]
```

```
import spacy
nlp = spacy.load('en_core_web_sm')
print(nlp.pipe_names)
doc = nlp("I'm travelling to San Jose")
[(e.text, e.label_) for e in doc.ents]
```

```
['tagger', 'parser', 'ner']
```

```
[('San Jose', 'GPE')]
```

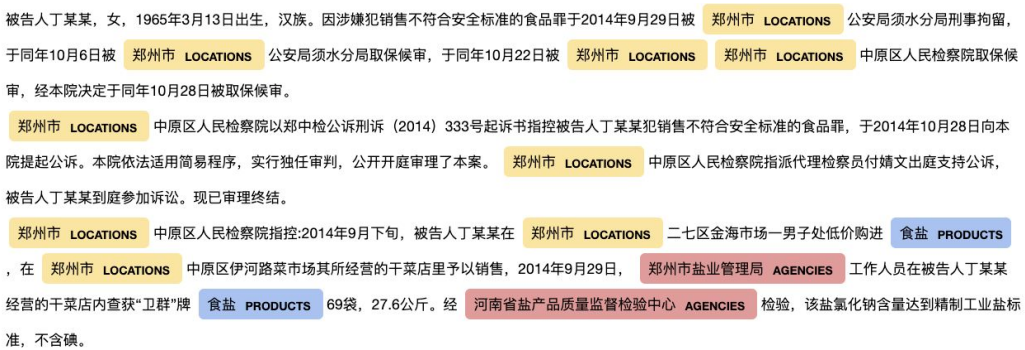
SpaCy's NER model



Matthew Honnibal on spaCy's NER model: https://spacy.io/universe/project/video-spacys-ner-model

Our data and inventories

- Dataset
  - Full-text for 22k court cases prosecuted by government agencies
- Inventories
  - Products
    - Sourced from our database of food inspection results
  - Agencies
    - Curated as part of the previous manual process
  - Locations
    - A selection of cities and all prefectures and provinces in China



Using SpaCy's PhraseMatcher

- PhraseMatcher
  - Given a pattern, extracts matches and can pass to callback function
    - (Match id, start token, end token)
- Custom callback
  - Default entity attribute can't handle overlap
  - Used custom attribute for Doc objects
    - Doc._.entities
  - For overlapping entities
    - If different types, choose higher priority one
    - If same types, go with longer entity

```
# intialize model
nlp = Chinese()
# initialize the matcher with model vocab
matcher = PhraseMatcher(nlp.vocab)
# add entity inventory as Doc objects from model
or i, c in enumerate(ENTITY_TYPES):
    matcher.add(c, add_entity, *parsed_ents[i])
```

Callback function

Constructing training and test datasets

- Year-split
  - How well will the model perform on future court case data?
  - Train on cases before 2017, test on after
  - Baseline: Inventory with only entities before 2017
- Excluded entities
  - How well does the model identify entities it hasn't seen?
  - Train on cases with 30% of entities from each inventory removed
  - Baseline: Inventory with remaining 70%

Method	Train		Test	
	Docs	Unique entities	Docs	Unique entities
Year-split	13,501	6,707	7,722	4,378
Excluded entities	14,859	8,548	6,364	2,564

Training the model

- Available hyperparameters
  - Dropout
  - Batch size
  - Optimizer
- Outputs loss with each epoch
  - Based on the model predicted tags (e.g. entity/non-entity)*
- Important to note
  - Plateauing: I haven't seen much movement in loss after 15-20 epochs
  - If updating: Not enough to just put new examples, model likely to forget what it's learned

```
nlp_model = Chinese()
ner = nlp_model.create_pipe('ner')
nlp_model.add_pipe(ner)
for l in labels:
    ner.add_label(l)
optimizer = nlp_model.begin_training()
sizes = compounding(1.0, 4.0, 1.001)
epoch = 15
for itn in range(epoch):
    random.shuffle(train_data)
    batches = minibatch(train_data, size=sizes)
    for batch in batches:
        texts, annotations = zip(*batch)
        nlp_model.update(texts, annotations, sgd=optimizer, drop=0.35, losses=losses)
    print("Losses", losses)
```

Losses {'ner': 169404.9403350675}

Losses {'ner': 79686.03164099755}

*More details on NER model loss: https://github.com/explosion/spaCy/issues/3360

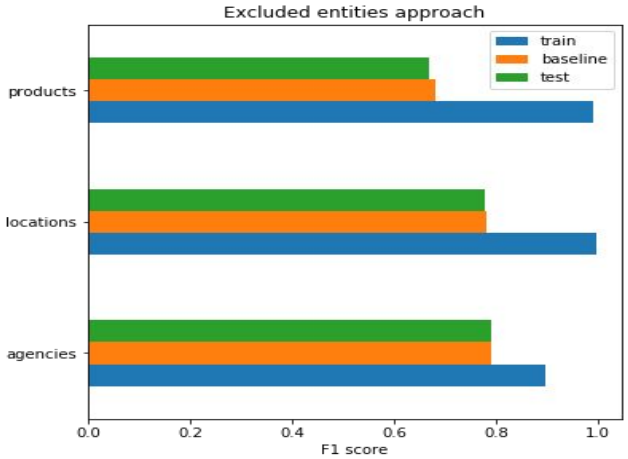
Scoring the result

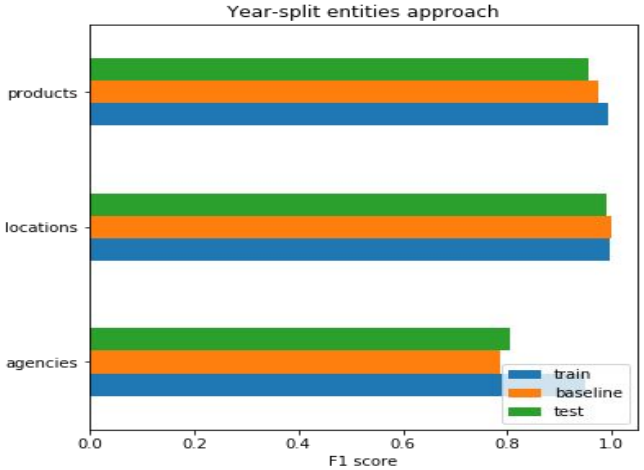
- Built-in SpaCy Scorer
  - Compare model parsing result to “gold standard”
  - Provides entity-level precision (“p”), recall (“r”) and F1-score (“f”)
  - Adapted to get baseline performance
- How does this model perform on data it has seen?
  - F1-score on training data
- How does this model perform on unseen data?
  - F1-score on test data
- How well does this model identify entities it hasn’t seen?
  - Recall of excluded entities

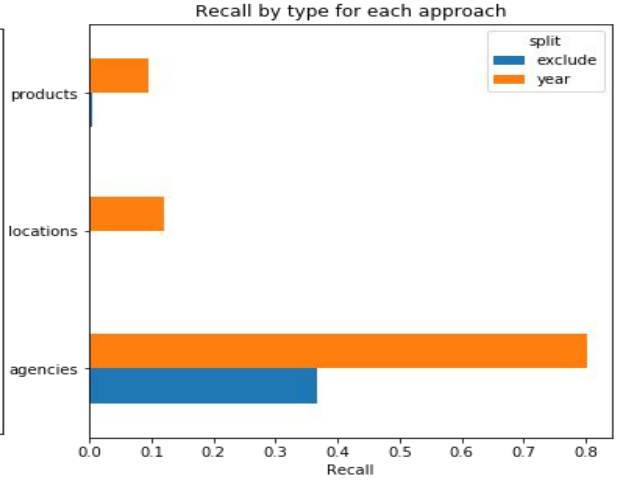
```
scorer = Scorer()
for doc, annot in test_data:
    doc_to_test = full_model(doc)
    gold_text = nlp(doc)
    gold = GoldParse(gold_text, entities=annot.get("entities"))
    scorer.score(doc_to_test, gold)
```

```
{'uas': 0.0,
 'las': 0.0,
 'las_per_type': {'': {'p': 0.0, 'r': 0.0, 'f': 0.0}},
 'ents_p': 97.69166443143628,
 'ents_r': 55.78143651884051,
 'ents_f': 71.0141561506281,
 'ents_per_type': {'agencies': {'p': 88.99794567450354,
   'r': 70.3917674670518,
   'f': 78.60887096774192},
  'locations': {'p': 99.61005302327793,
   'r': 63.799037524366476,
   'f': 77.7805627500673},
  'products': {'p': 97.63231014366795,
   'r': 51.04266349059916,
   'f': 67.03768671561359}},
 'tags_acc': 0.0,
 'token_acc': 100.0,
 'textcat_score': 0.0,
 'textcats_per_cat': {}}
```

Model-based NER has better performance on agencies than other entity types







What is our model missing?

Excluded entities

Agencies

'秭归县社区矫正工作管理局', 'Zigui County Community Corrections Administration'

'食品药品监督管理局', 'Food and Drug Administration'

Locations

'淄博市', 'Zibo'

'河南省', 'Henan Province'

Products

'大包', 'Big bag'

'食品', 'food'

Year split

Agencies

'食品药品监督管理局', 'Food and Drug Administration'

'动物卫生监督所', 'Animal Health Authority'

Locations

'漳州市', 'Zhangzhou City'

'新疆', 'Xinjiang'

Products

'减肥胶囊', 'Slimming Capsule'

'黄白', 'Yellow and white'

Next steps

- Sanitizing the inventory
  - Agencies: Sanitized inventory, reasonable model performance
  - Location: Pretty comprehensive inventory, may not benefit from model
  - Product: Noisy inventory, requires cleaning
- Improving the featureset
  - Using out-of-the-box model, may be improved by additional tweaking
  - Pretrained language models available for a variety of languages
- Reformulating the problem
  - Is NER necessary for our goal?

SOTA performance on CoNLL English NER task

RANK	METHOD	F1	EXTRA TRAINING DATA	PAPER TITLE
1	CNN Large + fine-tune	93.5	✓	Cloze-driven Pretraining of Self-attention Networks
2	GCDT + BERT-L	93.47	✓	GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling
3	I-DARTS + Flair	93.47	✓	Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition
4	LSTM-CRF+ELMo+BERT+Flair	93.38	✓	Neural Architectures for Nested NER through Linearization
5	Hierarchical + BERT	93.37	×	Hierarchical Contextualized Representation for Named Entity Recognition

https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

Thank you!

Github repo for code *not yet published*

https://github.com/bpben/ner_chinese_spacy

(Contains only sample data)

Thanks to:

SpaCy team: https://spacy.io/

FSAS team at MIT Sloan

Additional resources:

@honnibal's talk on NER with SpaCy: https://spacy.io/universe/project/video-spacys-ner-model

Ratinov 2009 paper on design considerations of NER: https://www.aclweb.org/anthology/W09-1119.pdf

Collobert 2011 paper on Bloom Embeddings: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf

Lample 2016 paper on Stack-LSTM: https://arxiv.org/pdf/1603.01360.pdf

Caltech CS/CNS/EE 155 lecture HMM, MMEM and CRFs; https://www.youtube.com/watch?v=B1nl8fLgKMk