Named-Entity Recognition with SpaCy

Benjamin Batorsky

About me

Data Scientist, focused on NLP

PhD, Policy Analysis

https://benbatorsky.com/

REAL WORLD DATA

Part of Ciox Health, health information management company representing 3/5ths of US hospitals

Focused on providing actionable health information and insights in the hands of researchers using AI + NLP technology

Disclaimer

- My work is leveraging spaCy’s capabilities, I haven’t built or contributed to their architecture
- NER is a hard problem, your mileage may vary from these configurations
- Much of the content in this presentation is from previous work, the Ciox pipelines are very different (though use the same theory)

Overview of this presentation

1. Introduction to NER
    a. Why it is difficult and why it is important
2. Approaches to NER
3. NER in SpaCy
4. Hands on example
5. Non-English research use-case
6. Recent advancements
7. Ciox RWD use-case

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 Feb 4, 2021, I spoke at PyData DC.

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 Feb 4, 2020 [DATE], I spoke at PyData [ORG] DC [LOC].

Also possibly PyData DC 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]

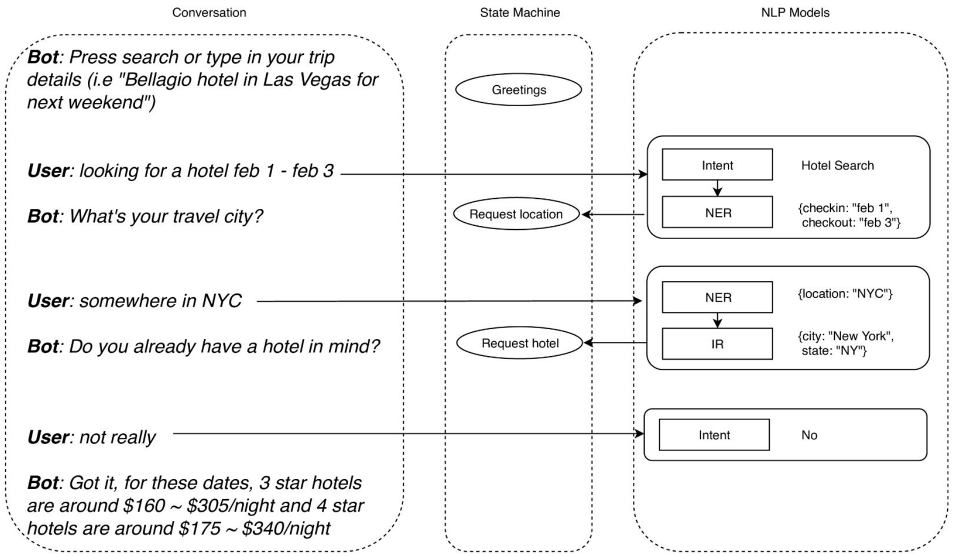
in [O]

DC [U-LOC]

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

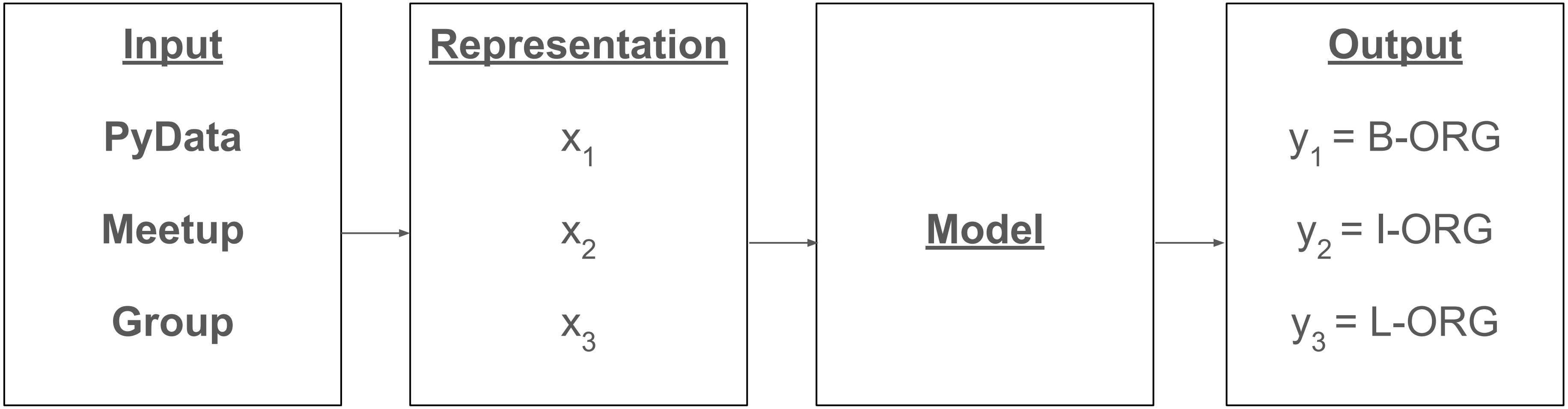
Why is NER important?

- 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

Overview of model-based NER



Representation

Single token

PyData

Word index

[0, 0, 1, ...]

Word features

{“word_idx”:2,
“length”:8, …}

Embedding

[0.03, 0.01, -0.12 …]

Context

(w_i, w_{i+1})

(PyData, Meetup)

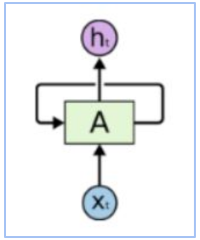
(Meetup, Group)

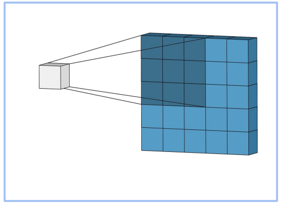
(Group, END)

Concatenation

{“w_i_idx”: [0,0,1,…],
“w_i_1_idx”: [0,1,0,…]}

Learned representations

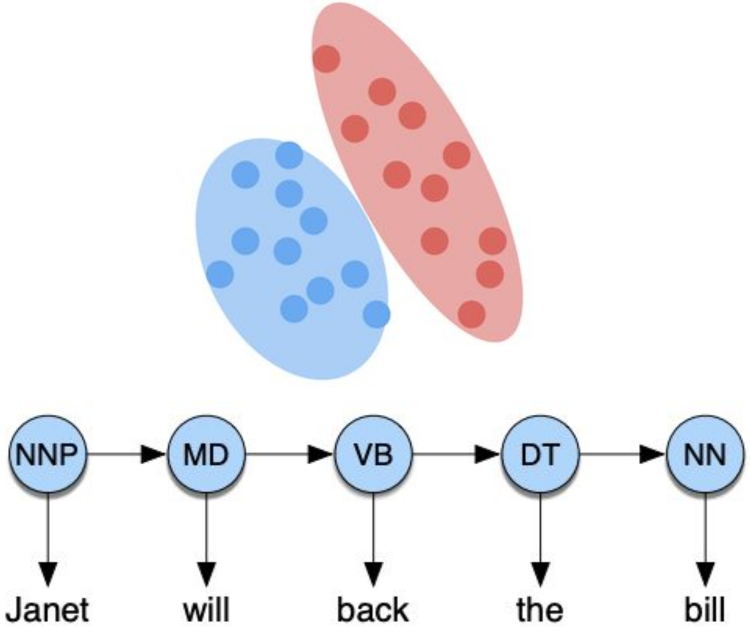




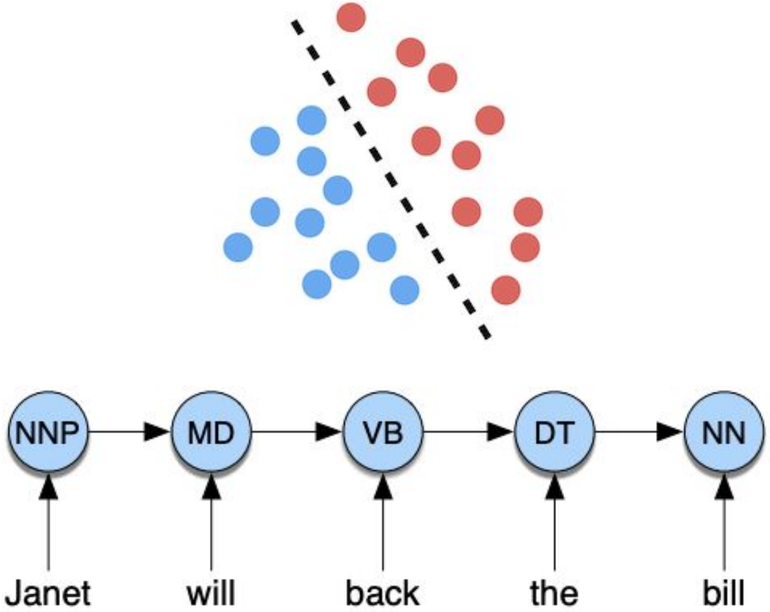
https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e
https://towardsdatascience.com/classifying-skin-lesions-with-convolutional-neural-networks-fc1302c60d54

Model

Generative (Naive Bayes, HMMs)



Discriminative (MEMMs, CRFs, Perceptron)

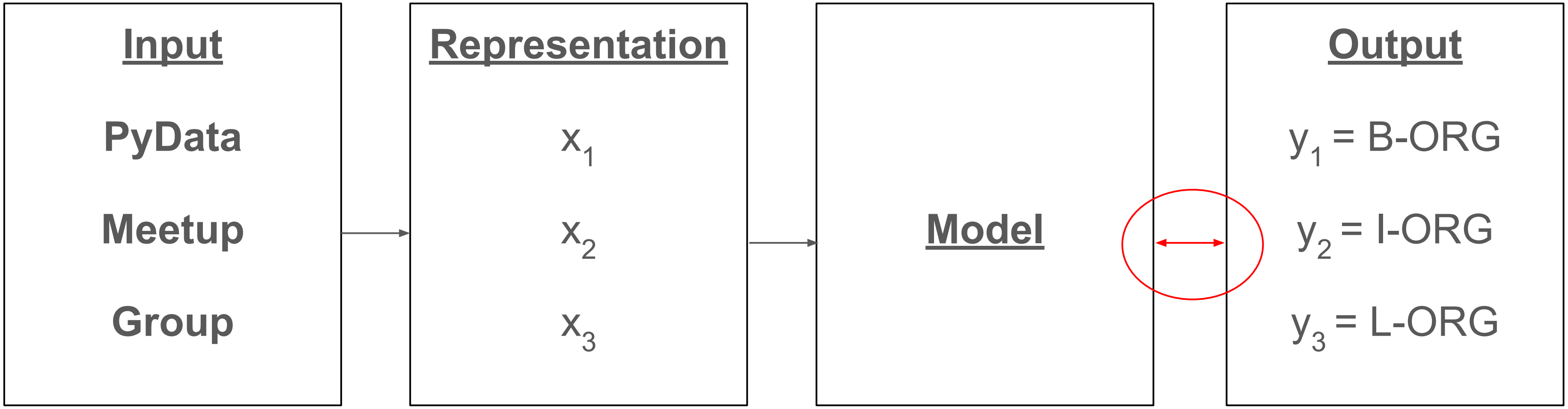


L Ratinov & D Roth (2009) Design Challenges and Misconceptions in Named Entity Recognition: https://www.aclweb.org/anthology/W09-1119.pdf

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

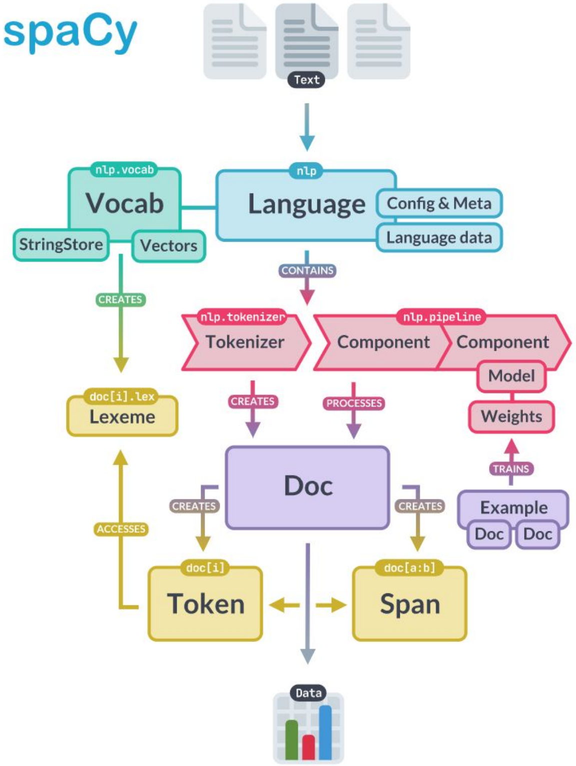
Images from: "Speech and Language Processing" Daniel Jurafsky & James H. Martin (https://web.stanford.edu/~jurafsky/slp3/8.pdf)

Overview of model-based NER



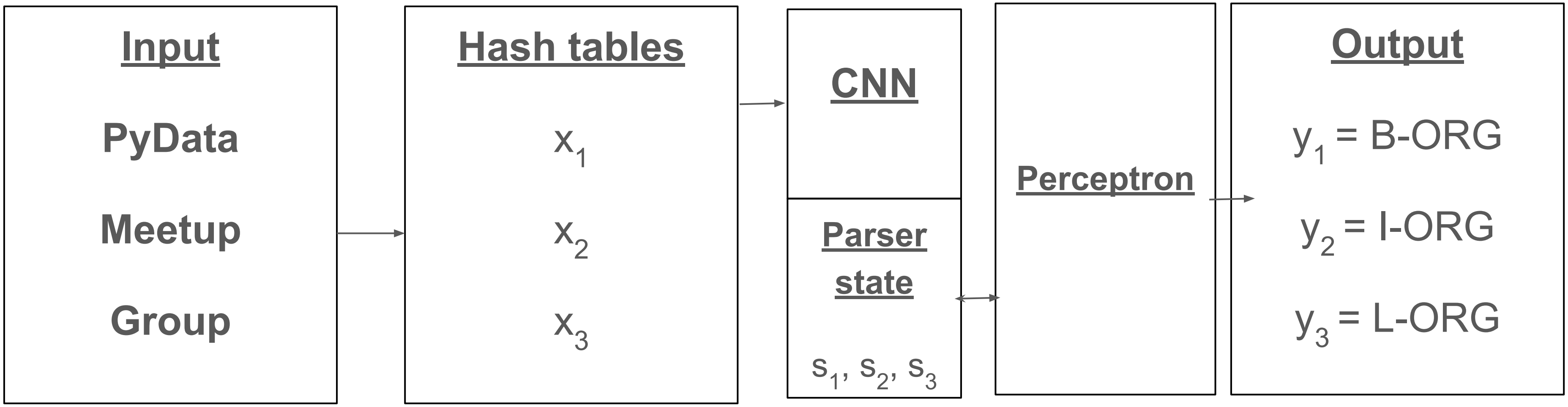
SpaCy: All-purpose NLP library in Python

- Uses “Language models” with tokenization, vocabulary, other pipeline elements
  - Documents->Spans->Tokens
- Components: Elements of pipeline processing docs, trainable
  - Part-of-speech
  - Categorization
  - NER
- Language model + components configurable, portable



https://nightly.spacy.io/api

SpaCy's NER model



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

Hands-on example notebook

https://github.com/bpben/pydata_dc_ner/blob/master/ner_usecase.ipynb

NER at MIT’s Food Supply Chain Analytics group

- Dataset: Full-text for 22k court cases
- Previous 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.”

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

被告人丁某某，女，1965年3月13日出生，汉族。因涉嫌犯销售不符合安全标准的食品罪于2014年9月29日被 郑州市 LOCATIONS 公安局须水分局刑事拘留，于同年10月6日被 郑州市 LOCATIONS 公安局须水分局取保候审，于同年10月22日被 郑州市 LOCATIONS 郑州市 LOCATIONS 中原区人民检察院取保候审，经本院决定于同年10月28日被取保候审。

郑州市 LOCATIONS 中原区人民检察院以郑中检公诉刑诉（2014）333号起诉书指控被告人丁某某犯销售不符合安全标准的食品罪，于2014年10月28日向本院提起公诉。本院依法适用简易程序，实行独任审判，公开开庭审理了本案。 郑州市 LOCATIONS 中原区人民检察院指派代理检察员付婧文出庭支持公诉，被告人丁某某到庭参加诉讼。现已审理终结。

郑州市 LOCATIONS 中原区人民检察院指控:2014年9月下旬，被告人丁某某在 郑州市 LOCATIONS 二七区金海市场一男子处低价购进 食盐 PRODUCTS ，在 郑州市 LOCATIONS 中原区伊河路菜市场其所经营的干菜店里予以销售，2014年9月29日， 郑州市盐业管理局 AGENCIES 工作人员在被告人丁某某经营的干菜店内查获“卫群”牌 食盐 PRODUCTS 69袋，27.6公斤。经 河南省盐产品质量监督检验中心 AGENCIES 检验，该盐氯化钠含量达到精制工业盐标准，不含碘。

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

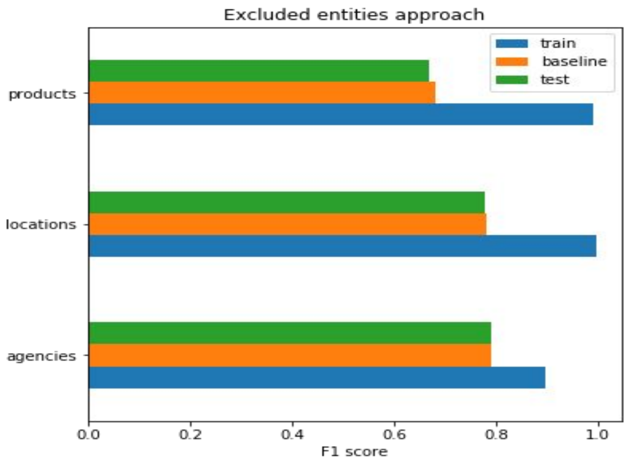
Scoring the result

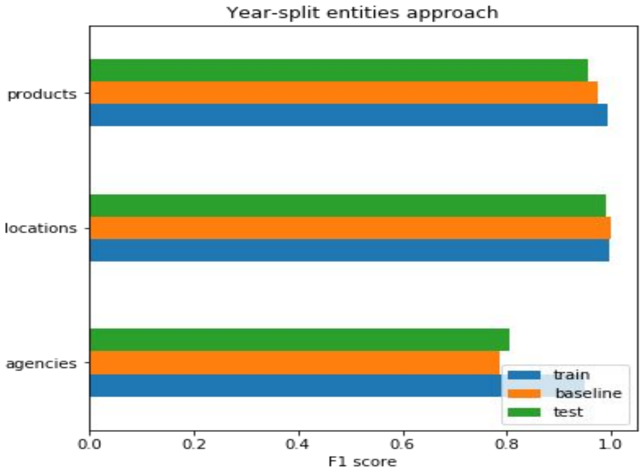
- Built-in SpaCy Scorer
  - Compare model parsing result to “gold standard”
  - Provides entity-level precision (“p”), recall (“r”) and F1-score (“f”)
  - Adapted to get baseline performance
- How does this model perform on data it has seen?
  - F1-score on training data
- How does this model perform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

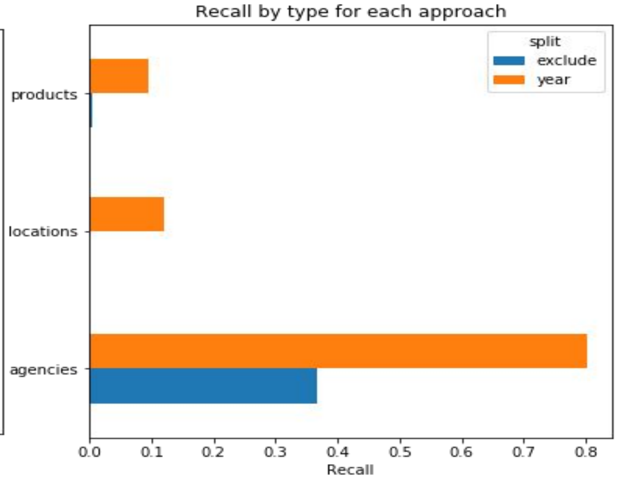
```
scorer = Scorer()
for doc, annot in test_data:
    doc_to_test = full_model(doc)
    gold_text = nlp(doc)
    gold = GoldParse(gold_text, entities=annot.get("entities"))
    scorer.score(doc_to_test, gold)
```

```
{'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







Recent advancements in SpaCy+NER

- Previous work made use of base NER model in SpaCy
- Better performance has been seen using transformer-based architectures
- Huggingface Transformers library puts transformers in reach for many applications
- Spacy-transformers integrates HF with SpaCy
- SpaCy v3.0, enables constructing complex pipelines for training and using models

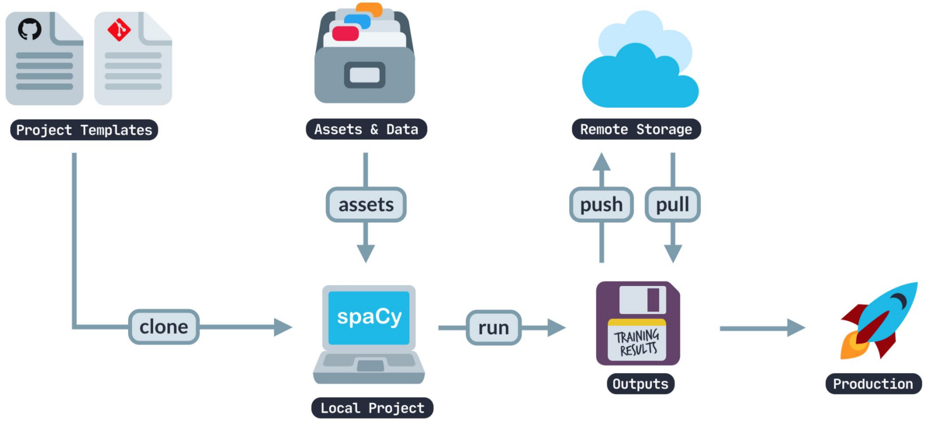
SOTA performance on CoNLL English NER task

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

SpaCy 3.0: Configurable, trainable NLP pipelines

- Streamlining chaining together different pipeline components
- project.yml
  - Connects elements of project (e.g. preprocessing, training)
- Training config
  - Connecting spaCy components (e.g. tagging model described in slides)
- Assets
  - Requirements (e.g. data) for the project (aside from scripts and config)
- Enables testing/tracking different configurations and data elements
- Blog announcement: https://explosion.ai/blog/spacy-v3



https://explosion.ai/blog/spacy-v3

[components.ner.model]

@architectures = "spacy.TransitionBasedParser.v1"

[components.tok2vec.model.embed]

@architectures = "spacy.MultiHashEmbed.v1"

[components.tok2vec.model.encode]

@architectures = "spacy.MaxoutWindowEncoder.v1"

Use-case at Ciox: i2b2 challenge for drug identification

- i2b2 challenges: Landmark NLP challenges on clinical records
- 2009: Medication challenges
  - Extract medications, dosages, frequency, etc from discharge summaries
- Ciox: Provide RWD for research, need to assess content of records
- I2b2 challenge provides clean baseline dataset for training and evaluation

Line no. text

63 well. Although left transmetatarsal amputation being considered ,
64 it was felt that she had a good chance of healing the wound
65 appropriately. She had a single temperature spike , although all
66 cultures remained negative. *She had continuation of her Heparin*
67 *while she was started on a course of Coumadin to reserve patency of*
68 *her graft. …*

Gold standard

m="heparin" 66:8 66:8||do="nm" ||mo="nm" ||f="nm" ||du="nm" ||r="nm"||ln="narrative"
m="coumadin" 67:8 67:8||do="nm" ||mo="nm" ||f="nm" ||du="nm" ||r="her graft." 68:0 68:1||ln="narrative"

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2995677/

Spacy 3 Drug NER from social media data

- Available here: https://github.com/explosion/projects/tree/v3/tutorials/ner_drugs
- Tutorial uses baseline NER model (described previously)
- With minor tweaks, can switch to transformer-based model

Config base (excerpt)

```
[components.tok2vec.model.embed]
@architectures = "spacy.MultiHashEmbed.v1"
width = ${components.tok2vec.model.encode.width}
[components.tok2vec.model.encode]
@architectures = "spacy.MaxoutWindowEncoder.v1"
```

Config transformer (excerpt)

```
[components.ner.model.tok2vec]
"spacy-transformers.Tok2VecTransformer.v1"
name = "distilbert-base-uncased"
```

	Base	Transformer
Precision	0.76	0.76
Recall	0.66	0.73
F1	0.71	0.75

Additional challenges

- i2b2 dataset = Clinical summaries
  - Patient records consists of many different sections
- Mentions of medications not always a "treatment"
  - Ciox models additionally include "allergies", very context-dependent
- 2010 i2b2 paper, rule-based systems did extremely well
  - Complex, based on curated medical inventories and expert annotation
  - Need to consider performance/complexity trade-offs

Group	Token-Level F-measure
USyd (Rules + CRF + SVM)	0.849
Vanderbilt (Rule-based)	0.823
NLM (Rule-based)	0.813
OpenU (Rule-based)	0.812
BME-Humboldt (Rule-based)	0.807
Manchester (Rule-based)	0.800

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2995677/

Thank you!

Github repo for code

https://github.com/bpben/pydata_dc_ner

Thanks to:

SpaCy team: https://spacy.io/

Paulo, Hussain, all the PyData DC community

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

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

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

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'