AdvNLP Week 5

Named Entity Recognition

Julie Weeds, Spring 2024

Previously

1. Distributional Semantics
   - Bootstrapping semantics from context
2. Word embeddings
   - Dimensionality reduction, neural language models, word2vec, GloVe
3. Probabilistic language models
   - n-gram modelling, perplexity and generalization
4. Neural language models
   - Word-based and character-based

Week 5 overview

1. Named Entity Recognition
   - What and why
   - Challenges
   - NE types
   - Sequence labelling
   - evaluation
   - features

2. Approaches to NER
   - Rule-based
   - Generative (e.g. HMM)
   - Discriminative (e.g. CRF)
   - Neural Models (e.g. CNN-BiLSTM-CRF, ACE)
   - Transformer LMs

3. Getting Data
   - Supervised
   - Semi-supervised
   - Unsupervised

Named Entity Recognition

- The detection and classification of named entities in a text
- A named entity is anything which can be referred to with a **proper name** e.g., "*Boris Johnson*", "*Pizza Hut*", "*Brighton*", but may also include dates, times, prices and more
- Often multi-word phrases
- Semantic structured prediction

Why Named Entity Recognition?

NER is the basis for downstream NLP tasks:

- Sentiment or intent regarding entities
- The relation between entities in fact/relation extraction
- Entity-linking (co-reference resolution) and Knowledge Base Population (KBP)
- Entity-informed search and question answering regarding an entity (Google/Bing Infobox, Google knowledge graph (GKG))
- Competitive/Product Intelligence, brand insights

Sample Text

1. Manchester United striker Wayne Rooney has agreed a new five-and-a-half year contract worth up to £300,000 a week.
2. The 28-year-old England international will extend his current contract by four years, tying him to the club until June 2019.
3. Rooney joined United from Everton in August 2004 and is only 42 goals shy of passing Bobby Charlton's record of 249 for United.
4. The former Everton forward has taken 430 games in all competitions to score his 208 goals for the Red Devils.
5. The club have yet to confirm the contract but BBC sports editor David Bond said the deal had been agreed.

Types of Named Entity

Type	Tag	Examples
People	PER	Wayne Rooney
Organization	ORG	Manchester United
Location	LOC	Manchester
Miscellaneous	MISC	£300,000

- The most popular NER benchmark, CoNLL-2003, includes only these 4 entity types
- Other benchmarks include more types, e.g. OntoNotes v5.0 includes 18 types
- Temporal expressions and numerical expressions are often included
- Some approaches aim to discover any entity regardless of type
- Which entity type set might be most useful?

Sample Text

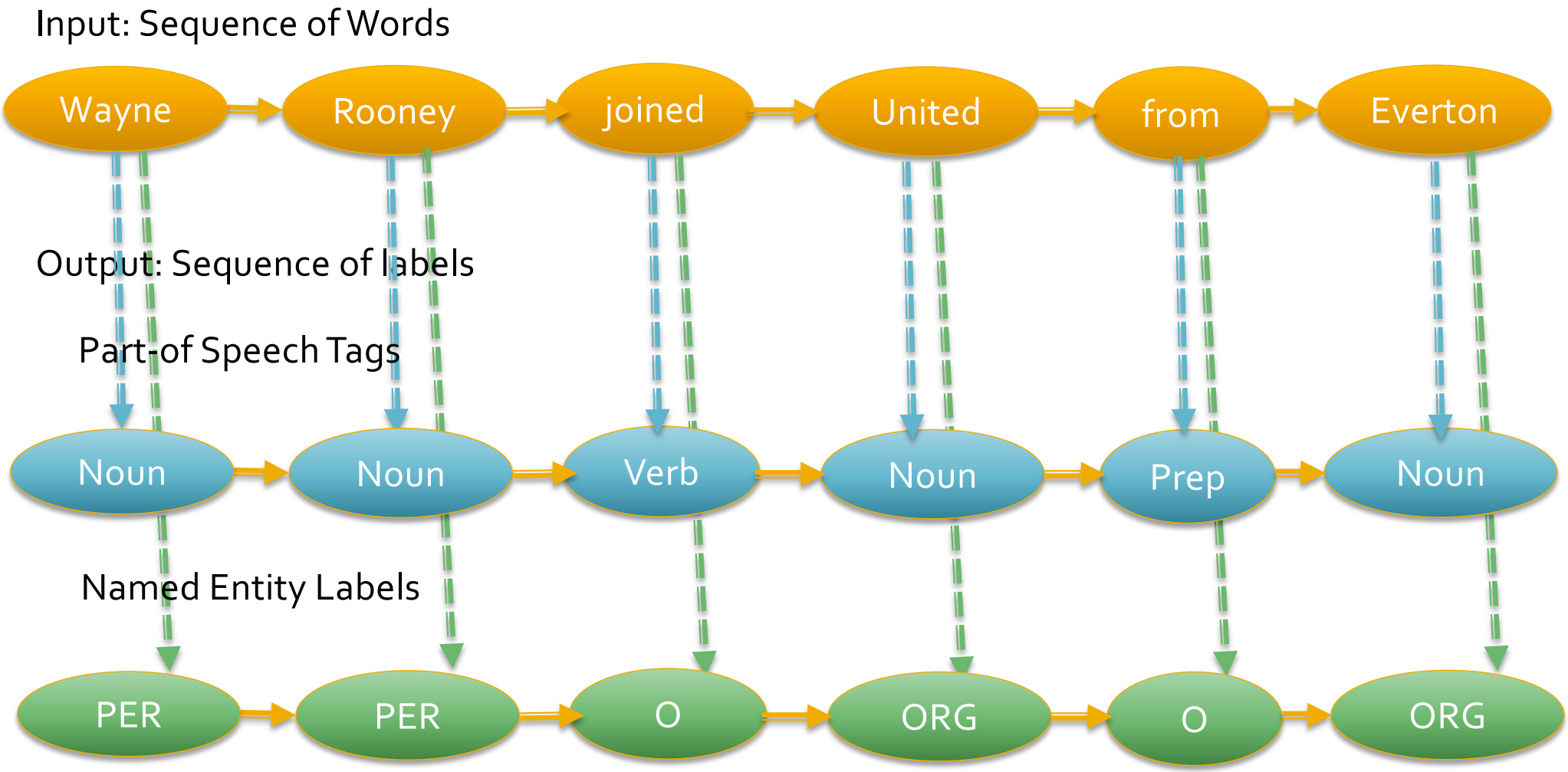
1. Manchester United striker Wayne Rooney has agreed a new five-and-a-half year contract worth up to £300,000 a week.
2. The 28-year-old England international will extend his current contract by four years, tying him to the club until June 2019.
3. Rooney joined United from Everton in August 2004 and is only 42 goals shy of passing Bobby Charlton's record of 249 for United.
4. The former Everton forward has taken 430 games in all competitions to score his 208 goals for the Red Devils.
5. The club have yet to confirm the contract but BBC sports editor David Bond said the deal had been agreed.

Types of Named Entity

Type	Tag	Examples
People	PER	Wayne Rooney
Organization	ORG	Manchester United
Location	LOC	Manchester
Miscellaneous	MISC	£300,000

- Difficulties:
  - Two types of ambiguity: boundary and type
  - Boundary detection (e.g. the New York Times)
  - Two types of type ambiguity: entity-noun, entity-entity
  - Variation, even with reliable text sources (e.g. United, Utd, Utd.)

Sequence Labelling



Sequence Labelling – IOB encoding

Words	Label
Manchester	B-ORG
United	I-ORG
striker	O
Wayne	B-PER
Rooney	I-PER
has	O
agreed	O
a	O
new	O
£300,000	B-MISC
contract	O
.	O

I → inside a chunk
O → outside a chunk
B → beginning a chunk

Sequence Labelling – IOB encoding

Words	Label
Manchester	B-ORG
United	I-ORG
striker	O
Wayne	B-PER
Rooney	I-PER
has	O

I → inside a chunk
O → outside a chunk
B → beginning a chunk

- Turn span identification into a sequence labelling problem using IOB encoding – 1 tag per token
- Find the correct spans of text that constitute an entity
- Typically entities should be identified with the correct type and exact position
- Sometimes multiple scores are included including partial overlaps
- Also known as BIO encoding

Evaluation

- $P = \frac{TP}{TP+FP}$
  - i.e., the proportion of named entities identified that are actually named entities
- $R = \frac{TP}{TP+FN}$
  - i.e., the proportion of named entities in the test data that were correctly identified
- $F_1 = \frac{2PR}{P+R}$
  - i.e., a harmonic mean of both

Features commonly used in NER

Entity Type	Tag	Examples
People	PER	Wayne Rooney
Organization	ORG	Manchester United
Location	LOC	Manchester
Miscellaneous	MISC	£300,000

Which rules or features might be useful in identifying these entity types?

Features commonly used in NER

Feature	example
lexical item	united
stemmed lexical item	unite
shape	initial capitalization
part-of-speech	N
syntactic chunk labels	part of NP
presence in gazetteer or name list	e.g., list of first names or place names
predictive words in context	Mr.
Bag of words/n-grams	preceding word(s) e.g., “striker”

What might the choice of features depend on?

Typical Supervised Approach

- Humans annotate training documents
- IOB encoding
- Feature extraction performed if necessary
- Classifier (e.g., HMM, SVM, MEMM, CRF, LSTM or a combination) trained
- Evaluation carried out on held out test data typically using precision, recall and the F-measure (F1)
- Paper is published (possibly)

Classification

- Sequence labelling task –
  - We want to assign the most likely sequence of labels given the observed tokens
  - NOT the most likely label for each token given all of the other tokens
  - So, this might rule out simple classifiers such as Naïve Bayes and Logistic Regression / MaxEnt

AdvNLP Week 5

Named Entity Recognition Part 2

Julie Weeds, Spring 2024

NER Approaches

- Approximate chronology of NER approaches
  - Rule-based
  - Generative (e.g. HMM)
  - Discriminative (e.g. MEMM, CRF)
  - RNN-based, usually LSTM
  - (Usually) Transformer/attention-based large language models, fine-tuned (e.g. BERT)
  - Hybrid

Rule-based

- Heuristics, entity lookup lists (gazeteers)
- Varying degrees of inclusion of supervised ML
- Bootstrapping using initial set of seeds
- Still used in non-academic scenarios

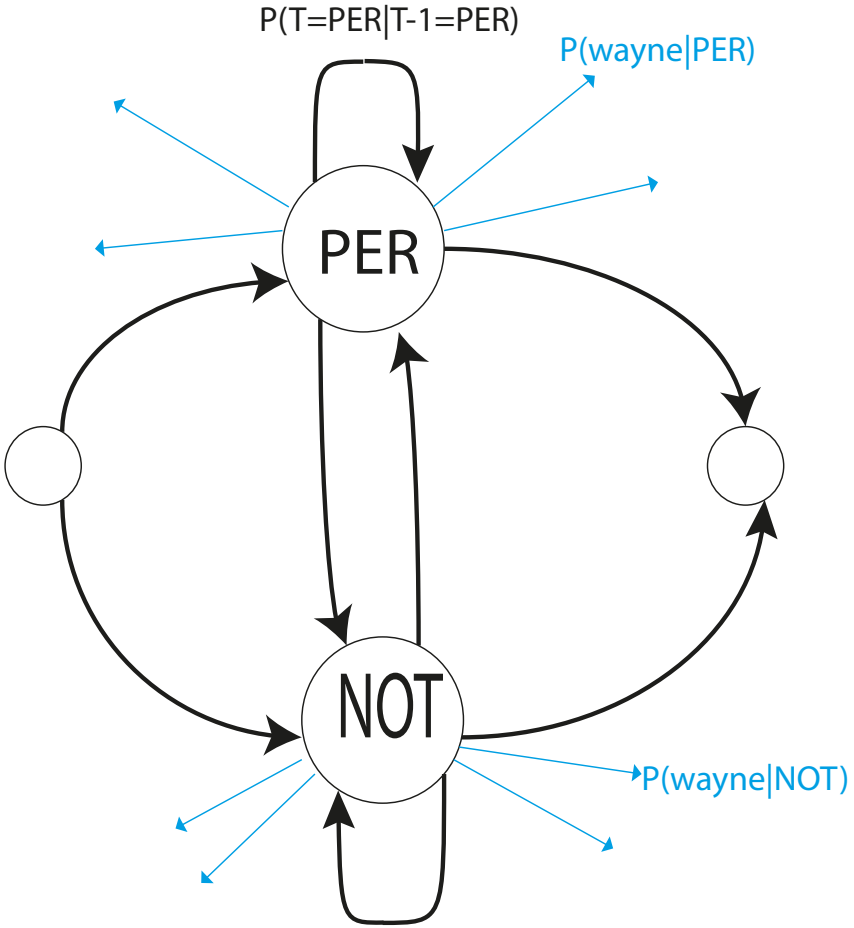
	Pros	Cons
Rule-based	• Declarative • Easy to comprehend • Easy to maintain • Easy to incorporate domain knowledge • Easy to trace and fix the cause of errors	• Heuristic • Requires tedious manual labor
ML-based	• Trainable • Adaptable • Reduces manual effort	• Requires labeled data • Requires retraining for domain adaptation • Requires ML expertise to use or maintain • Opaque

Chiticariu et al. (2013)

Generative models

- Model joint probability distributions from training data
- Can be used to generate new data from these distributions
- Use Bayes theorem to obtain a conditional probability to label unseen data
- Examples of generative models are Naïve Bayes and the Hidden Markov Model

Simple HMM



Defined by

- **state transition probabilities** (independence assumption that current state depend only on previous state) often given as a **state transition matrix**; and
- **emission probabilities** P(observed word | state)

These probabilities can be derived from labelled corpora using maximum likelihood estimation (MLE).

More on HMMs

- To maximise probability of tag sequence given word sequence, Bayes Rule is applied

$$\Pr(tags \mid words) = \frac{\Pr(words \mid tags) \times \Pr(tags)}{P(words)}$$

- The denominator is a constant so it can be ignored.
- The probability of the tag sequence comes from the transition probabilities
- The probability of the word sequence given the tag sequence comes from the emission probabilities
- Typically use the Viterbi Algorithm to find the tag sequence which optimises this probability

Hidden Markov Model (HMM)

- Drawbacks
  - Arise from the two simplifying assumptions
  - Bigram limits model history
  - Feature independence limits model richness
  - Difficult to add additional features

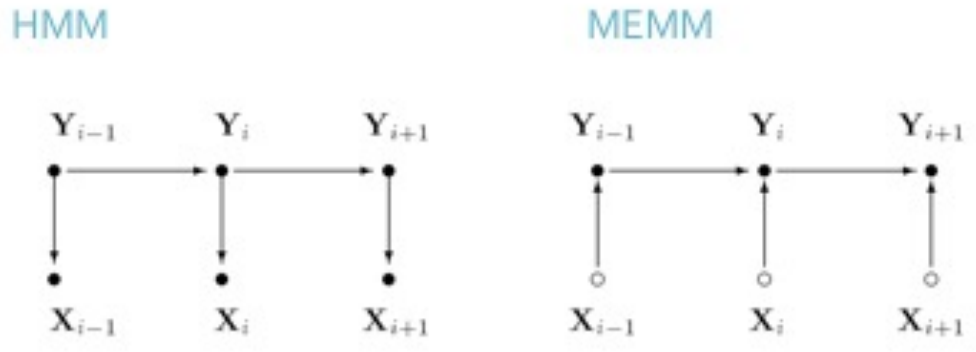
$$P(words|tags) = \prod_{j=1}^{n} P(word_j|tag_j)$$

ONLY VALID THROUGH INDEPENDENCE ASSUMPTION!

Discriminative models

- Simplify the problem by discriminating between classes (NE tag types) rather than modelling every data point
- Leverage multiple (potentially interdependent) features (e.g. word shape, presence in gazeteer, PoS, word embeddings) which can improve prediction and help with sparsity
- Examples of discriminative models are the Maximum Entropy Markov Model (MEMM) and the Conditional Random Field (CRF)

Model for MEMM



If the observations are given by X and the tag sequence is given by Y,

- In a HMM, we model P(X|Y)
- In a MEMM, we model P(Y|X)

$$P(y_1 \dots y_n | x_1 \dots x_n) = \prod_{i=1}^{n} P(y_i | x_i, y_{i-1})$$

$$P(y_i | x_i, y_{i-1}) = \frac{1}{Z} \exp\left(\sum_{j=1}^{k} \lambda_j f_j(x_i, y_i, y_{i-1}) \right)$$

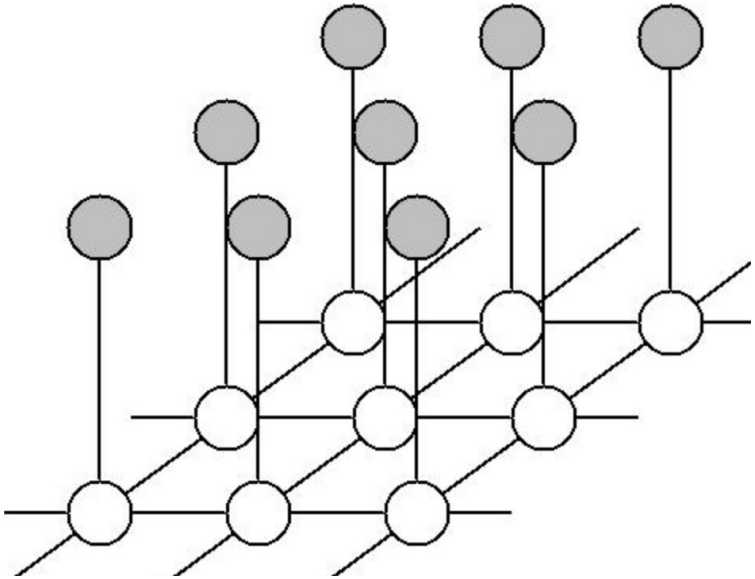
Label Bias Problem

- MEMMs use a per-state exponential model
  - Leads to label bias problem
  - Role of second (or subsequent) tokens in a chunk (multi-word NE) in distinguishing class can be lost
- Solution: Conditional Random Fields (CRFs):
  - CRFs have a single exponential model for the joint probability of the entire label sequence

So what is a CRF?

- A CRF is an undirected graphical model where the vertices can be divided exactly into two disjoint sets, X and Y, which are the observed and output variables respectively. We then model the conditional distribution p(Y|X).

A general CRF could look like this



Observed variables X

Hidden variables / labels / output variables Y

Linear Chain CRFs

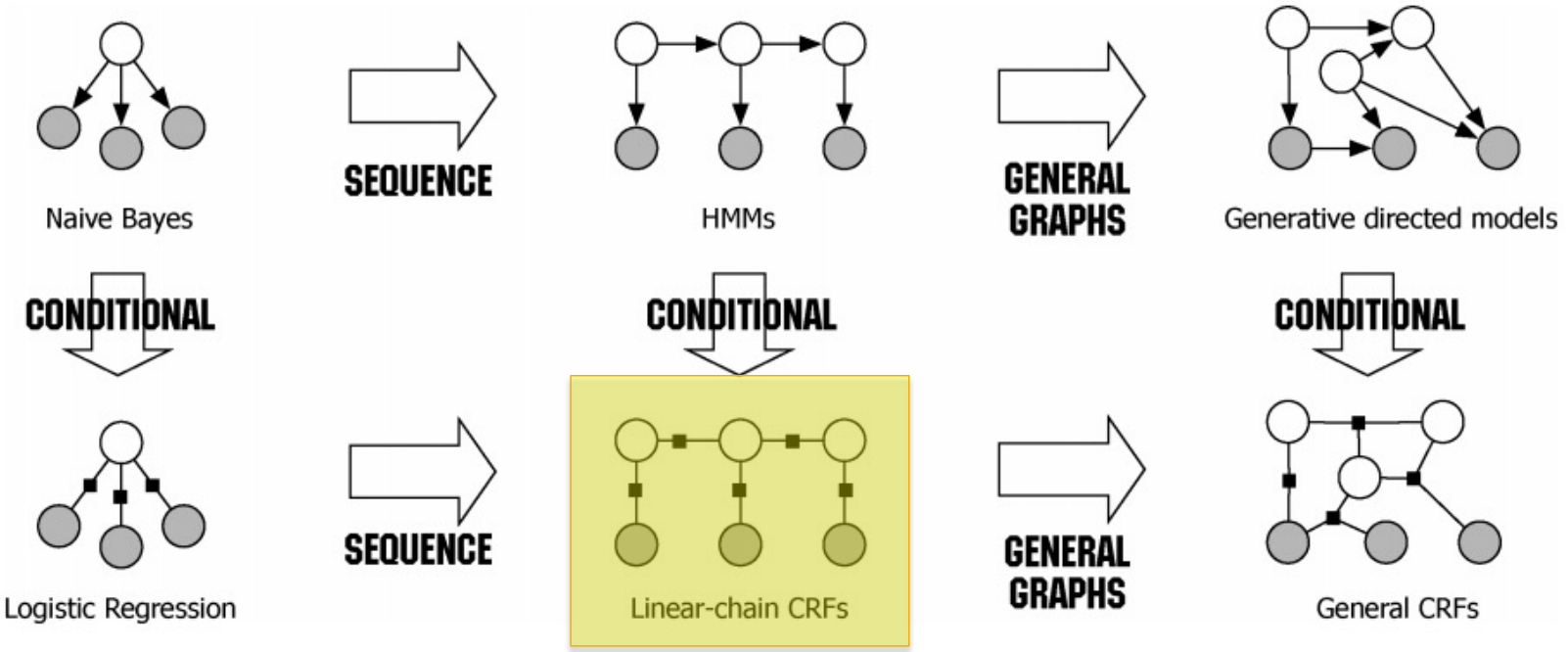


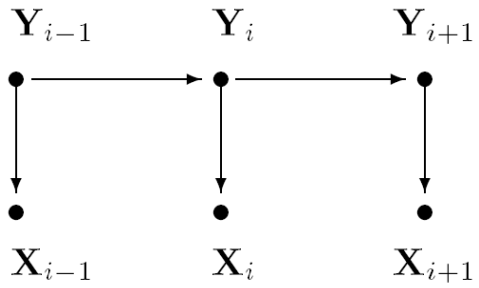
Fig. 2.4 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

[Sutton and McCallum, 2010]

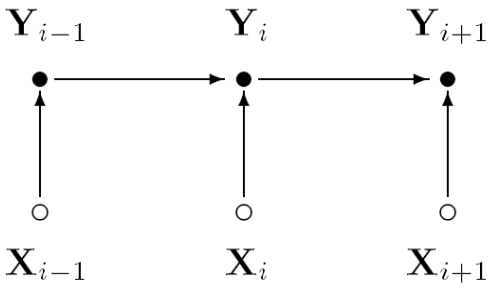
Graphical comparison among HMMs, MEMMs and CRFs

(from Lafferty et al.)

HMM



MEMM



CRF

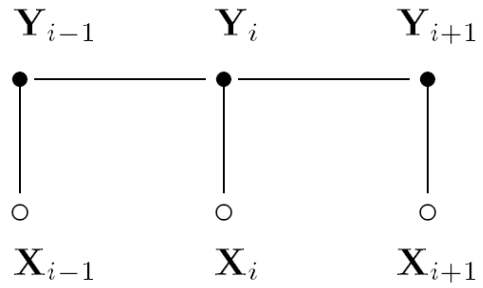


Figure 2. Graphical structures of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences. An open circle indicates that the variable is not generated by the model.

Linear Chain CRF

- If the G=(V,E) of Y is a chain, the joint distribution over the label sequence Y given X has the form

$$p(Y|X) \propto \exp\left(\sum_{e\in E,k} \lambda_k f_k\left(e, Y\big|_e, X\right) + \sum_{v\in V,k} \mu_k g_k\left(v, Y\big|_v, X\right)\right)$$

where x is a data sequence, y a label sequence and $y|_S$ is the set of components of y associated with the vertices in subgraph S

Parameter Estimation

- Need to determine the parameters $\theta=(\lambda_1 \ldots \lambda_n, \mu_1 \ldots \mu_n)$ from the training data D $\{(x^i,y^i)\}^N_{i=1}$ with empirical distribution *p(x,y)*
- Want to maximise the log-likelihood objective function

$$O(\theta) \propto \sum_{x,y} \mathbf{p}(x,y) \log p_\theta(y \mid x)$$

- This can be done with a gradient descent algorithm or an iterative scaling algorithm, e.g., improved iterative scaling (IIS) of Della Pietra et al. 1997

Drawbacks of using CRFs

- CRF performs well but is subject to drawbacks
- Very slow to train
- Lafferty et al.2001 report (on the same dataset):
  - MEMM+ trained to convergence from uniform distribution in 100 iterations
  - CRF did not converge after 2000 iterations from the uniform distribution
  - using the MEMM+ parameters to initialise the CRF, it converged in 1000 iterations
- Features need to be hand-crafted

Neural Models

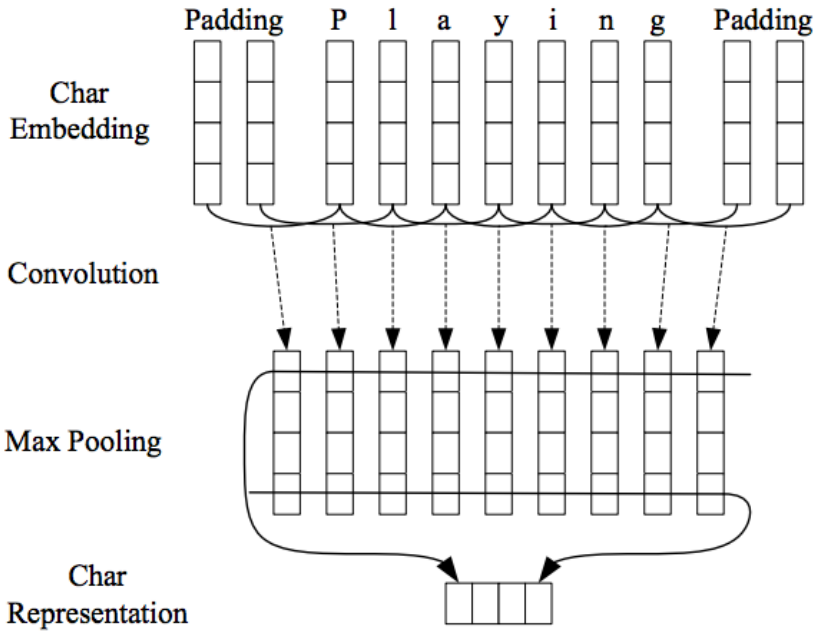
- Bidirectional-LSTM is most popular (non-transformer-based)
- Vanilla RNN and GRU have also been used
- NNs allow word, char or other embeddings to be used (or generated)
- Ma and Hovy (2016) is a good base Bi-LSTM architecture (CoNLL English SotA in 2016 and basis for Wang et al. SotA in 2021)

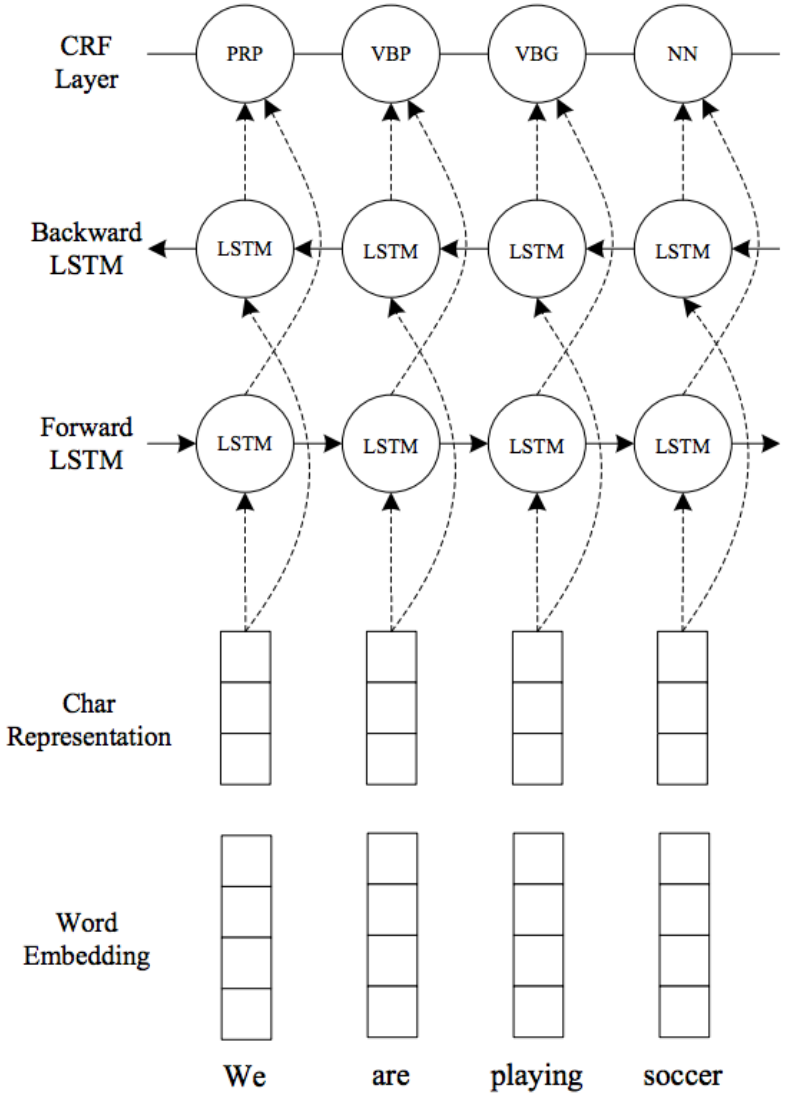
Ma and Hovy (2016)

- Also our seminar paper for this week
- A char representation for each word is generated from a CNN based on char embeddings
- Char rep is concatenated with GloVe word embedding
- Bi-directional LSTM is achieved by concatenating left-to-right and right-to-left LSTM hidden states
- CRF is used as output layer (rather than Softmax) to decode label sequences

Ma and Hovy (2016)

- Architecture diagrams from paper
- CRF shows tags from PoS task, NER task uses same config (except for initial LR)





Advantages of Neural Approach

- End-to-end sequence labelling
  - No feature-engineering
- Achieved F1 91.21 which was SotA in 2016

AdvNLP Week 5

Named Entity Recognition Part 3

Julie Weeds, Spring 2024

Supervised Learning

- For tasks like NER, supervised learning is often the best performing option
- Data requirements - as much as possible?
- Have some expectations of cost/time/acceptable performance
- Reduce data requirements using transfer learning e.g. pre-trained embeddings or deep LMs with fine-tuning

Sequence Labelling – IOB encoding

Words	Label
Manchester	B-ORG
United	I-ORG
striker	O
Wayne	B-PER
Rooney	I-PER
has	O
agreed	O
a	O
new	O
£300,000	B-MISC
contract	O
.	O

I → inside a chunk
O → outside a chunk
B → beginning a chunk

Existing Datasets

- Unless your task is completely novel, a dataset likely already exists
- Most papers eval on multiple datasets
- NER examples: CoNLL, OntoNotes, ACE, WNUT, various versions and languages
- There are many more
- See https://paperswithcode.com/task/named-entity-recognition-ner

Hand-labelling

- No existing dataset
- Novel task (e.g. NER with HTML or a low-resource language)
- Unlabelled data is abundant
- Labelling it by hand is costly and tedious
- Not best use of a researcher's time!
- Improve efficiency with:
  - UI support
  - Active learning
  - Crowdsourcing

Annotation User Interfaces

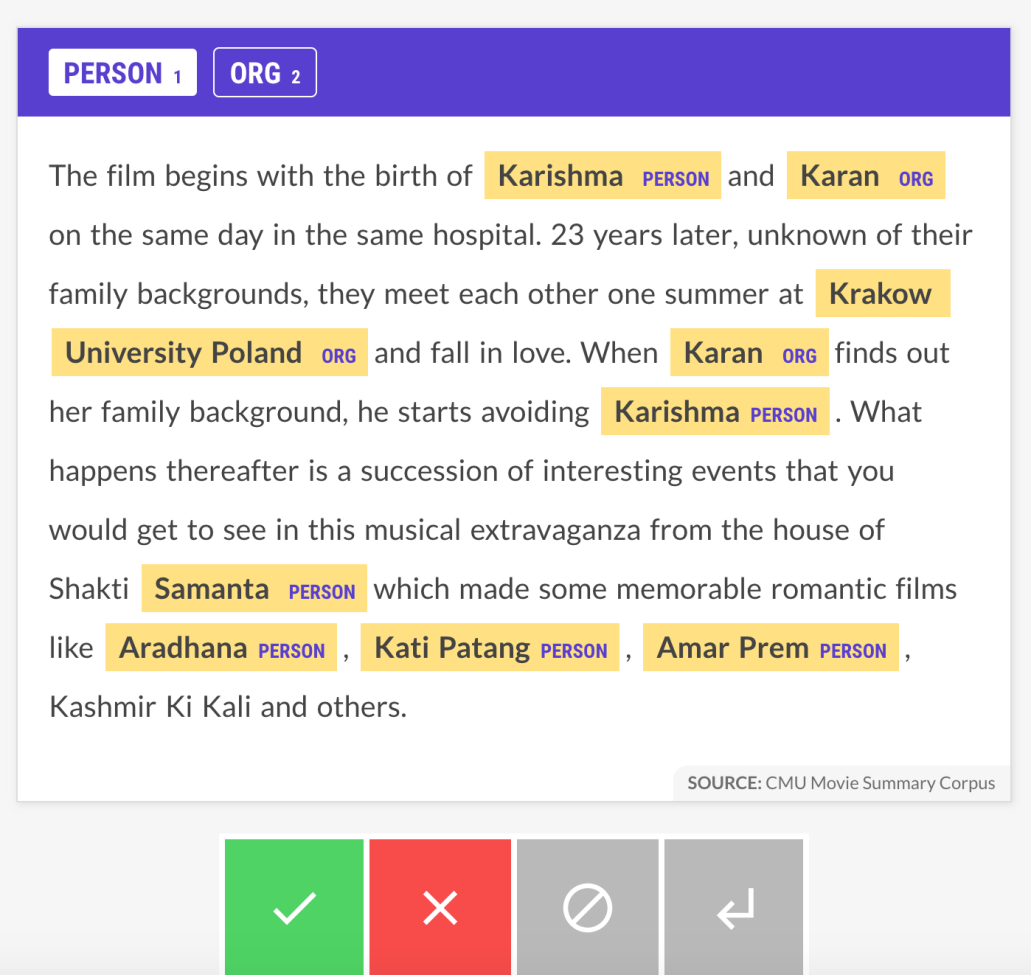
- As with most tasks, a good UI = efficiency
- A large unlabelled dataset can be labelled efficiently
- A trained model is required to suggest annotations which are then accepted/rejected/amended by user
- NER labelling applications:
  - prodigy (https://prodi.gy/)

    Paid, tightly integrated with spaCy for Active Learning
  - doccano (https://doccano.github.io/doccano/)

    Open source, labelling model integration through API
  - Label Studio (https://labelstud.io/)

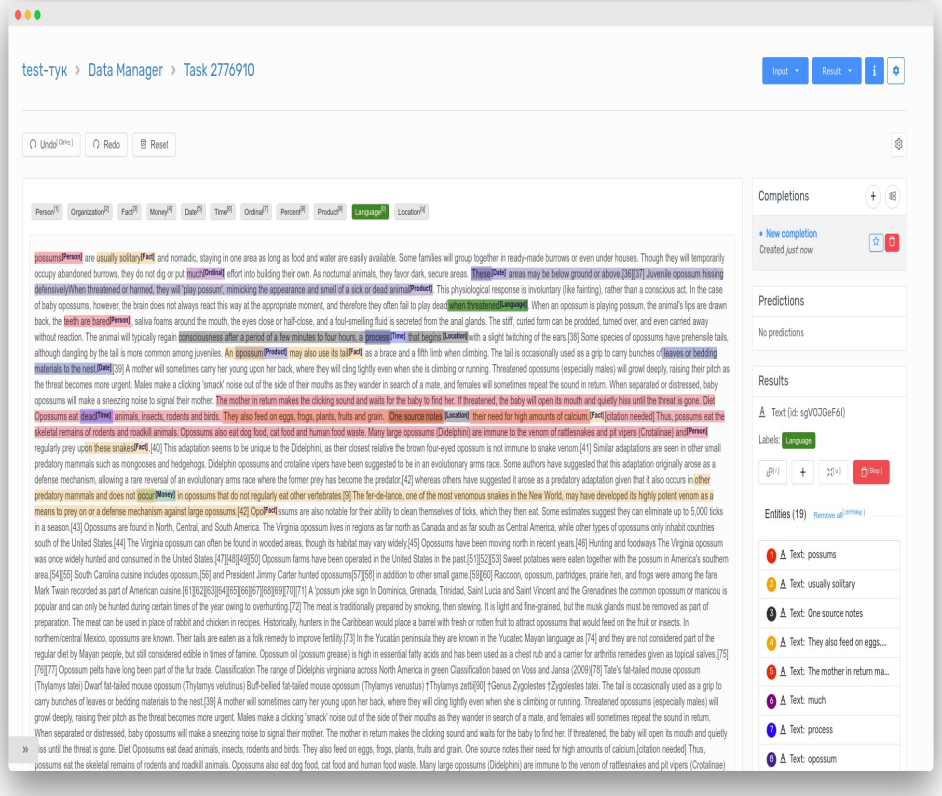
    Open source/paid, backend model integration for AL

prodigy



- Accept, reject or ignore all annotations
- Amend or delete (or undo) individual annotations
- Simple clear UI
- Binary training mode

Label Studio



- Similar UI to prodigy
- Integrates with your model (e.g. Torch)
- Larger entity set and longer text harder to label
- Takes time and concentration
- Design labelling task accordingly

Active Learning

- Don't label your whole dataset
- Actively label the best sentences while training
- Process:
  1. Select a model and unlabelled dataset
  2. Hand-label a small sample of data
  3. Train the model
  4. Select a batch of the most informative sentences
  5. Repeat from 2 until convergence (or some other constraint)

- What do we mean by "most informative"?

Semi-Supervised Learning

- Generate training data with minimal human interaction
- Unlabelled data is abundant
- Focus on labels i.e. providing rules or examples for identifying each entity type
- The main two semi-supervised techniques are bootstrapping and distant supervision, which share roughly equal popularity

Bootstrapping

- Using a classifier's own predictions to label more data
- General process:
  1. Train classifier (minimally trained initially) to convergence
  2. Label an unlabelled corpus with this classifier
  3. Add the most confident predictions to training data
  4. Iterate (25 iterations – Kozareva (2006))
- The most confident predictions? The reverse of Active Learning!

Disadvantages of Bootstrapping

- Problem 1: classifier errors are compounded with each iteration (semantic drift)
- Problem 2: the most confident classifications are not the most informative
- Solution: co-training (or even tri-training)
  - Use two (or more) classifiers
  - Use different classifier algorithms, biases, feature sets or differently sampled data
  - Use agreement between classifiers as the criteria for adding instances to training data
  - New instances can be swapped between classifiers for training
  - Disagreement (of one classifier) can be used in tri-training to prevent only easy data being supplied
- Can be expensive computationally (we have lots of unlabelled data to iterate over)

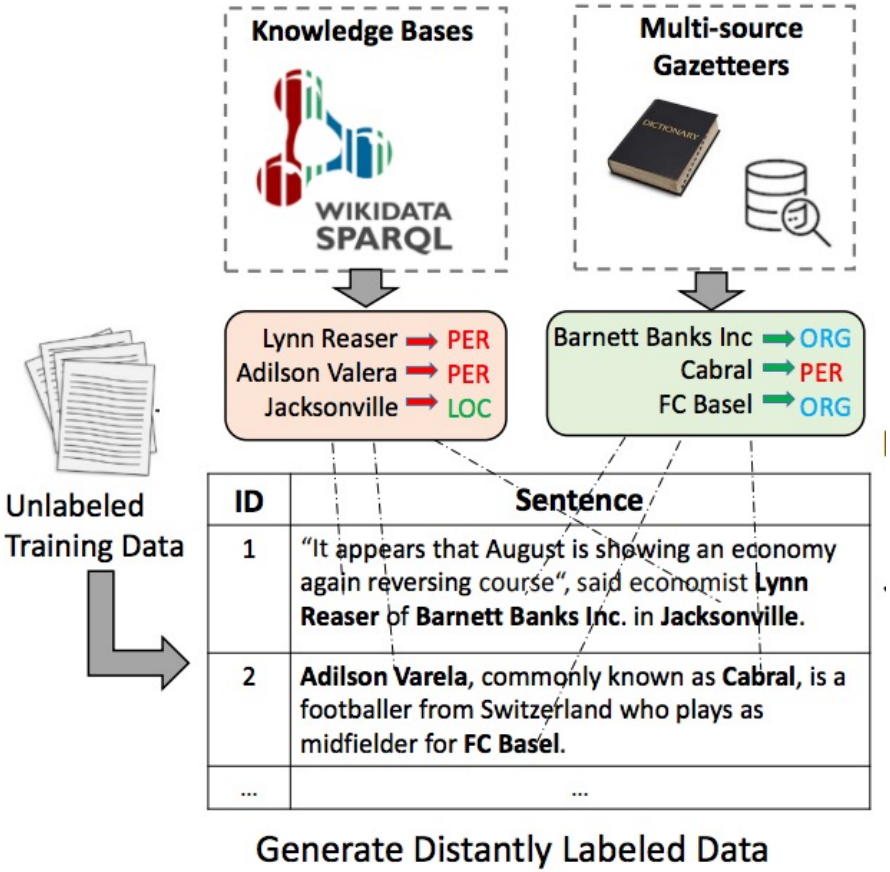
Ruder and Plank (2018)

- Strong baselines for neural semi-supervised learning under domain shift
- Bootstrapping brought up to date
- Char + word embedding (GloVe) + Bi-LSTM
- They evaluate self-training, tri-training and tri-training with disagreement
- PoS (not NER) and sentiment tasks
- Standard tri-training performed well on PoS SANCL 2021

Distant Supervision

- Use a knowledge base as a large set of examples of a given label (e.g. entity) type
- Commonly used KBs: DBPedia, Wikidata, YAGO, Cyc
- KBs typically contain tuples/relations, so Relation Extraction task fits well
- RE process can also be used to perform NER
- Basic process:
  1. Select relation type (e.g. places of birth)
  2. Get tuples from KB that express that relation (e.g. person->birthPlace->location)
  3. Find sentences in an unlabelled corpus (e.g. the Web) that contain both entities
  4. Label those entities as person and location and add sentence to training data OR train a model on the context surrounding those entities
- Matching techniques (between KB and unlabelled corpus):
  - String or RegEx
  - Heuristics (e.g. PoS rules, titles, word shape)

Distant Supervision



- Distant labelling example with direct matching
- From Liang et al. (2020)

Disadvantages of Distant Supervision

- Opposing problems
  - Incomplete annotation
    - KBs have limited coverage
    - Entities in unlabelled corpus unmatched
    - False negatives / limited *recall*
  - Noisy annotation
    - Ambiguity causes mis-labelling
    - E.g. KB contains "Manchester" (a LOC), corpus contains "Manchester Utd." (an ORG)
    - False positives / limited *precision*

Distant Supervision Solutions

- Incomplete annotation solutions
  - Use more sources!
  - Or knowledge of language (i.e. pre-trained LMs)
- Noisy annotation solutions
  - Induce labels based on occurrence counts

    Not good for open-domain with high-ambiguity labels
  - Filter out sentences with low matching quality (e.g. indicators of unlabelled entities, e.g. Mr/Ms or Inc. next to unlabelled tokens)

    Can be effective, but hand-written for each entity type
- Liang et al. (2020) tackle both problems

Distant Supervision

- Other methods that may qualify as "distant"
  - Wikipedia hyperlinks and page object_type
  - KB used directly as a gazeteer input to model
- Applications exist to support Distant Supervsion:
  - Stanford Snorkel

    Clean and integrate weak/distant data sources

    Create labelling functions

    Evaluate accuracy, coverage, overlaps, conflicts of functions

    Combine functions and sources into a labelling model

Unsupervised Learning

- For NER – the techniques used are generally heuristics, patterns and clustering
  - Hearst patterns (1992) is still a good baseline to beat: Lexical patterns used to identify semantically similar nouns e.g. the city of <city>
  - Clustering embeddings of words to find words of similar types (what is close to "London" in the vector space?)

Week 5 summary

1. Named Entity Recognition
   - What and why
   - Challenges
   - NE types
   - Sequence labelling
   - evaluation
   - features

2. Approaches to NER
   - Rule-based
   - Generative (e.g. HMM)
   - Discriminative (e.g. CRF)
   - Neural Models (e.g. CNN-BiLSTM-CRF, ACE)
   - Transformer LMs

3. Getting Data
   - Supervised
   - Semi-supervised
   - Unsupervised

Before the seminar

- Please read Ma and Hovy (2016)