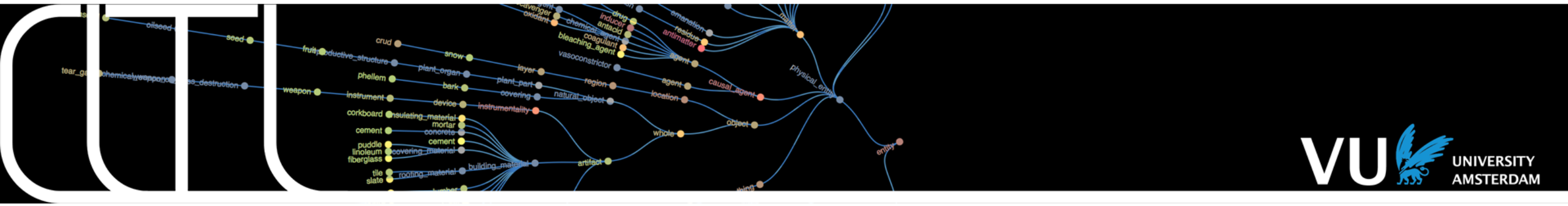
Text Mining CBS 2019

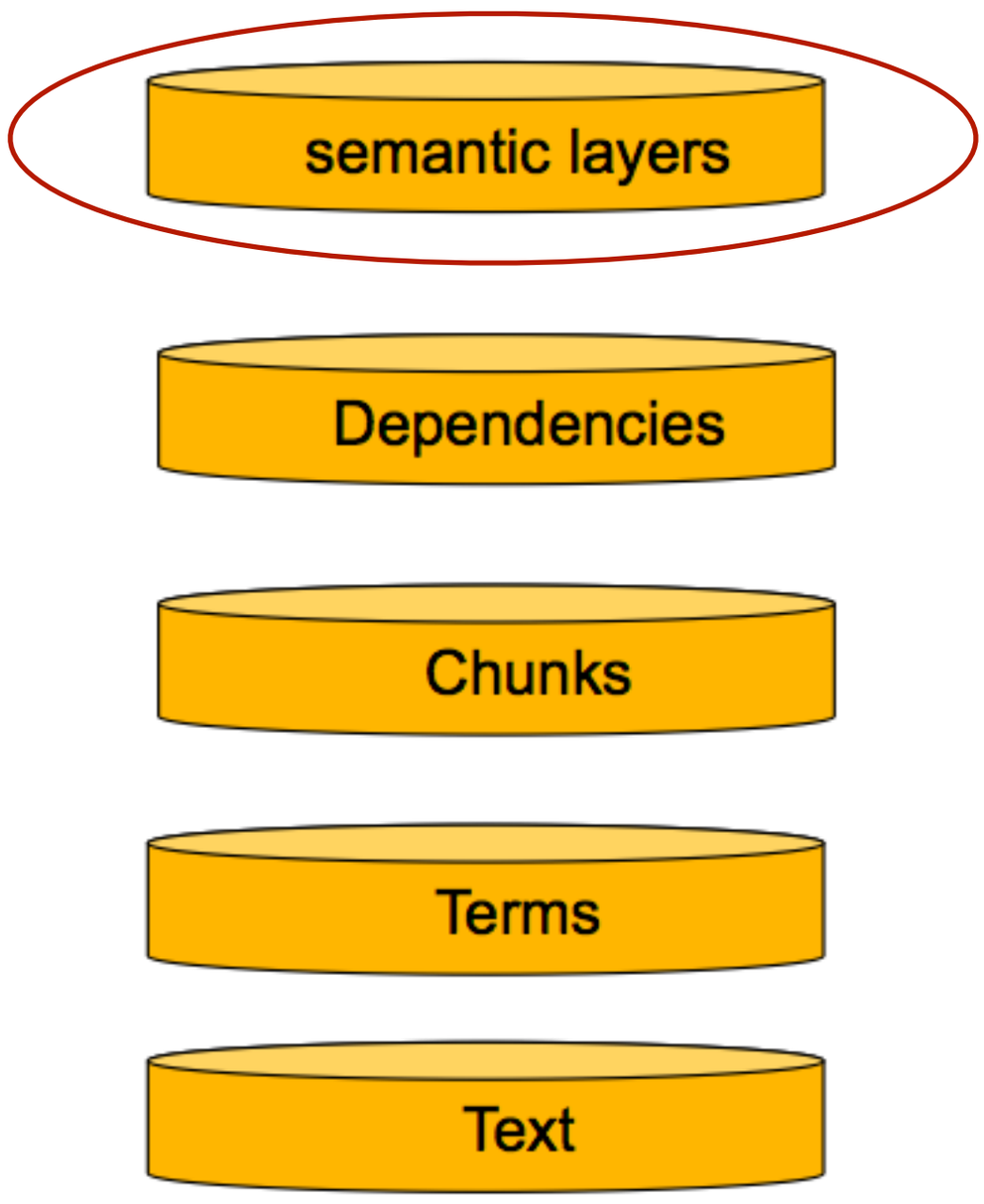
Lecture 4: Named entity detection and classification

Piek Vossen



Overview

- What is NER, NEC and NEL?
- Approaches to NERC
- Evaluating a NERC system
- What is coreference of entity expressions, phrases and pronouns
- What is named entity disambiguation (NEL)



What is Named Entity Recognition?

- Named Entity Recognition is the task of finding and classifying names in text.

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

What is Named Entity Recognition?

- Named Entity Recognition is the task of finding and classifying names in text.

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

People

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

What is Named Entity Recognition?

- Named Entity Recognition is the task of finding and classifying names in text.

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

People

Locations

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

What is Named Entity Recognition?

- Named Entity Recognition is the task of finding and classifying names in text.

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

People

Locations

Organisations

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

What is Named Entity Recognition?

- Named Entity Recognition is the task of finding and classifying names in text.

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

People

Time

Locations

Organisations

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

What is Named Entity Recognition?

- Named Entity Recognition is the task of finding and classifying names in text.

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

People

Time

Locations

Events

Organisations

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

Why are Named Entities important?

- Named entities can be indexed
- Sentiment can be attributed to entities to show trends
- We store knowledge extracted from text in Wikipedia's for Named entities

https://trends.google.com/trends/topcharts#geo&date=2017

	Searches		People		Global News
1	Hurricane Irma	1	Matt Lauer	1	Hurricane Irma
2	iPhone 8	2	Meghan Markle	2	Bitcoin
3	iPhone X	3	Nadia Toffa	3	Las Vegas Shooting
4	Matt Lauer	4	Harvey Weinstein	4	North Korea
5	Meghan Markle	5	Kevin Spacey	5	Solar Eclipse

Document Forensics
Network Institute project VU - Deloitte

- Investigating unsavoury business practices (e.g., slavery, fraud, bribery) can involve processing large numbers of contracts, yearly reports and external (news) sources that may reflect on a company's reputation and relations.

- Labour intensive task mainly using text search to identify relevant documents that are then manually processed.

- Project goals:

  - Extract the relevant concepts from unstructured texts (e.g. news) as well as semi-structured (e.g., contracts and financial) documents:

    - name of suppliers; the type of relationship between companies, executive management

  - Populate knowledge graphs and link them to publicly available knowledge graphs.

  - Knowledge graphs should reflect the temporal binding and provenance of the extracted relations and properties.

  - Enable automated reasoning about companies and their relationships such as structure of ownership or supply chains and their dynamics

Diligence detection

Auzina and Kim, 2019, Automated Due Diligence: Building Knowledge Graphs from News, Network Institute, VU University

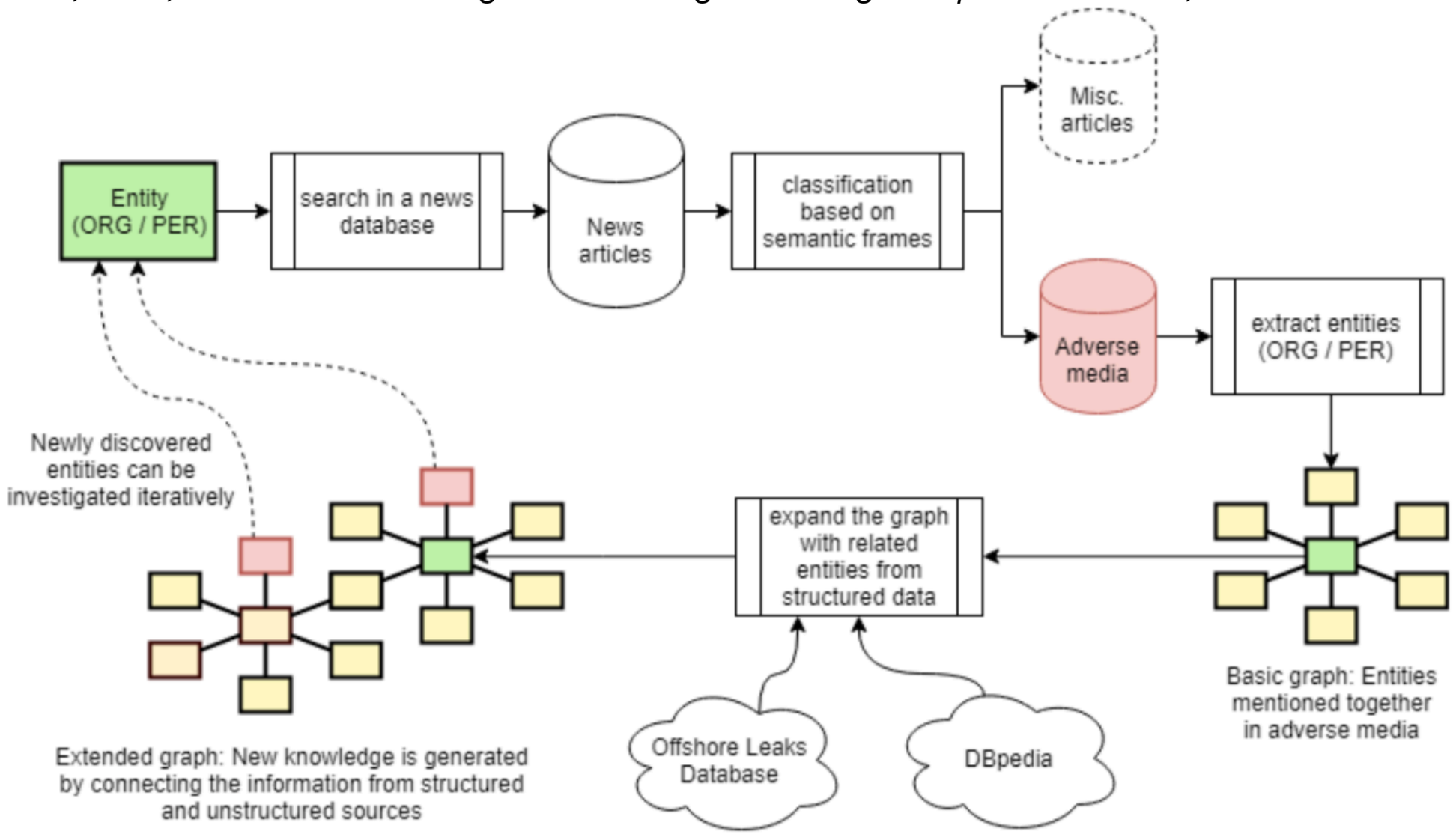


Figure 1: Overview of the proposed automated due diligence solution.

Adverse media classifier

Port of Moerdijk

Mitsubishi Materials

Kobe Steel

	MM dataset	KS dataset
Source	Nexis Uni[7]	Nexis Uni
Search term	Mitsubishi Materials	Kobe Steel
Content type	news	news
Language	English	English
Dates	06/91-02/19	01/00-04/19
# articles	707	1,774
# unique frames	460	540

Table 1: Datasets overview

Topic	# articles
MM dataset	
data falsification	65
forced labor during WWII	36
groundwater contamination	2
condos on contaminated soil	2
factory blast	1
KS dataset	
data falsification	115
tax evasion	2
asbestos-related employee death	1
employee embezzlement	1
safety and health violations	1

Table 2: Adverse media topics encountered during annotation

Model	Test set	Class	Precision	Recall	F1-score	Support
MM_10	KS: active learning sample (N=300)	0	0.65	0.89	0.75	177
		1	**0.67**	**0.31**	**0.42**	**123**
MM_10	KS: random sample (N=300)	0	0.90	0.95	0.92	242
		1	**0.73**	**0.55**	**0.63**	**58**
KS_10	MM: active learning sample (N=300)	0	0.82	0.91	0.86	194
		1	**0.80**	**0.63**	**0.71**	**106**
KS_10	MM: random sample (N=300)	0	0.91	0.97	0.94	240
		1	**0.82**	**0.62**	**0.70**	**60**

Table 3: Quantitative Evaluation Results (class 1: adverse media)

Auzina and Kim, 2019

Entity relationship graph

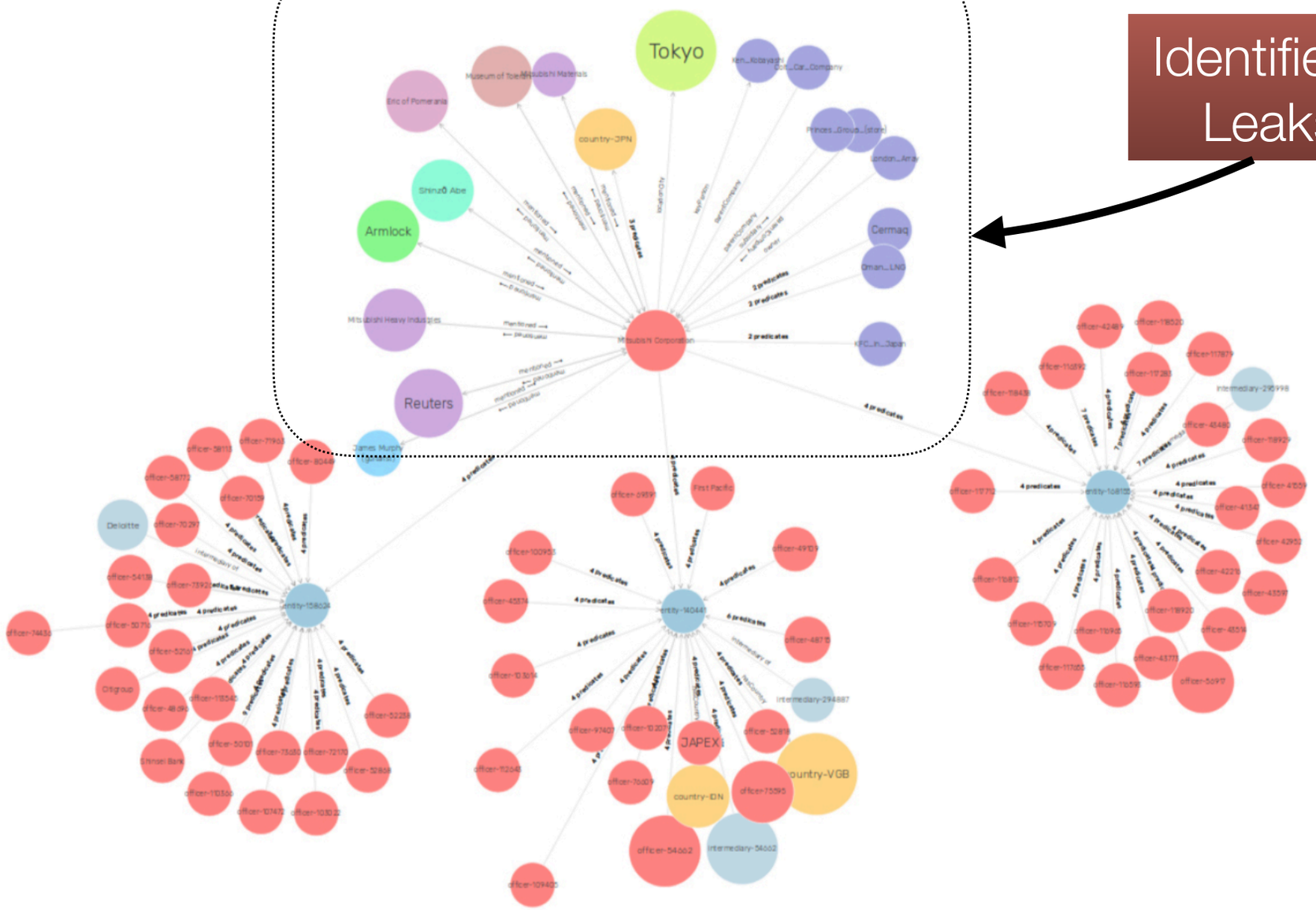


Figure 7: Extracted Officers and Intermediaries of the 3 identified entities

Auzina and Kim, 2019

Subtasks

- NER(Recognition): detecting the phrase that is the name of an entity
- NEC(Classification): assigning an entity type to the phrase
- NEL(Linking): establishing the identity of the entity in a given reference database (Wikipedia, DBPedia, YAGO) *Also called NED (**D**isambiguation)*
- Coreference: any phrase that makes reference to an entity instance, including pronouns, noun phrases, abbreviations, acronyms, etc…
- Preprocessing: tokenisation, sentence splitting, Part-of-speech tagging, lookup, grammar rules, coreference

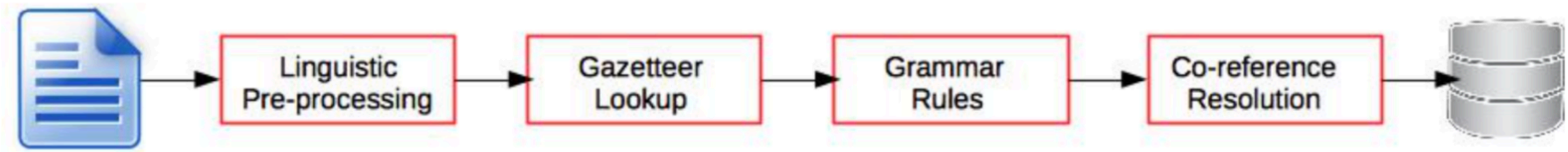


Figure 3.1: Typical NERC pipeline

What makes it a hard task?

- **variation** (IBM, The Big Blue, New York, NY, The Big Apple) and **ambiguity** **(**distinguish named entities and entities):
  - *MAY MAY RULE IN MAY*
  - *Austin Reed, Parkinson's disease, Pythagoras' Theorem*
- **extent**: *Sir Robert Walpole; [Abraham Lincoln, [the 16th President of the [United States]]]* <— nested entities
- **types**: e.g. *Criminal* as a subclass of *Person*, http://nerd.eurecom.fr/ontology, Fine-grained entity typing (e.g. FIGER uses 112 types from Freebase)
- **Time**: 8am, yesterday, last week, this month/year (TimeML types DAY, TIME, DURATION, SET)
- **Metonymy**: US, Holland, The Netherlands, Ford, Volkswagen

Machine learning

- Training
  1. Collect a set of representative training documents
  2. Label each token for its entity class (Person, Location, etc.) or other (O)
  3. Extract features to guide the classifier (often based on linguistic preprocessing)
  4. Train a classifier to predict the labels —> this results in a model
- Testing
  5. Take a set of test documents
  6. Represent each token with features as is done during training
  7. Run the *trained model* to label each token
  8. Output the recognised entities and the rest of the text (e.g. IOB tagging)

Features to use

- **Words & Word shapes**
  - Current words of the name expression, “New”, “York”
  - Previous/next word (preceding tokens such as “Mr.” or “President” could be helpful)
  - Capitalisation (initial capital, allcaps)
- **Linguistic information**
  - Part-of-speech, chunks (Noun Phrases, Propositional Phrases, Subject)
- **Label context**
  - Previous/next part-of-speech label and/or previous/next word
- Listed in a **gazetteer**: names of places, companies, countries, people
- https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus

CoNLL style of token labelling, using IO and IOB format

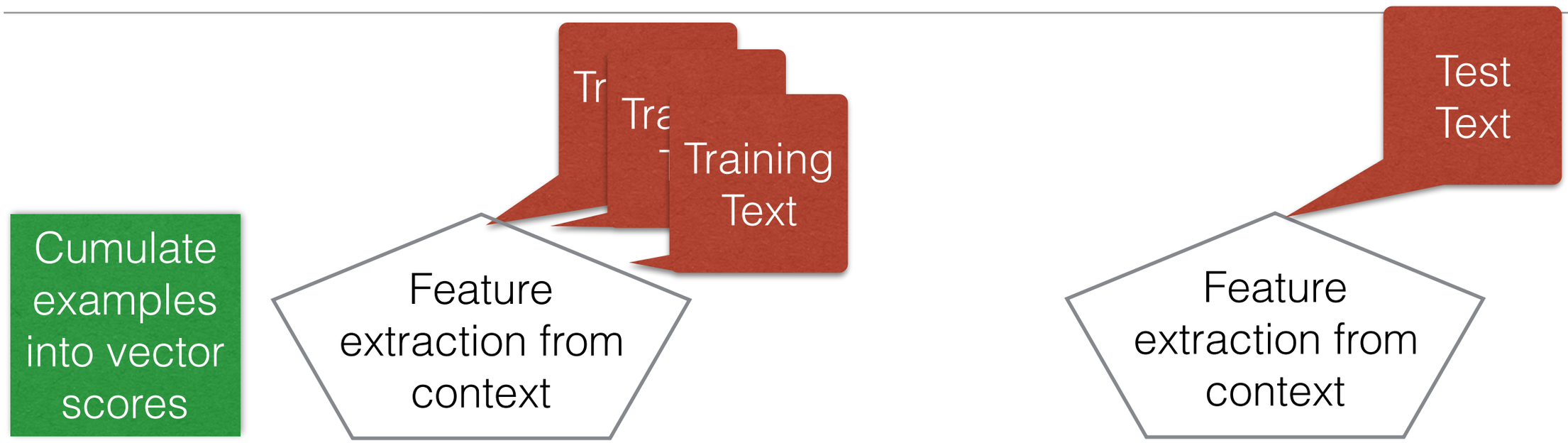
Tokens	**IO encoding**	**IOB encoding**
Abraham	PER	B-PER
Lincoln	PER	I-PER
(	O	O
February	TIME	B-TIME
12	TIME	I-TIME
,	TIME	I-TIME
1809	TIME	I-TIME
-	O	O
April	TIME	B-TIME
15	TIME	I-TIME
,	TIME	I-TIME
1865	TIME	I-TIME
was	O	O
the	O	O
16th	PER	B-PER

Tokens	POS	CHUNK	IO encoding
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

- See also the: BILUO tagging scheme to describe the entity boundaries: Begin, In, Last (final entity token), Unit (single token entity), Out (non-entity token)

- Problem for IO(B) and BILUO are nested entities —> ignored

Anything can be a feature and turned into a vector
Anything can be a predicted label, e.g. I, O, B



same vector positions and length across

LABEL [0,1,0,0,1,0,1,1,0]

LABEL [1,0,1,0,1,0,0,0,1]

[1,0,0,0,1,0,1,0,0]

feature vectors

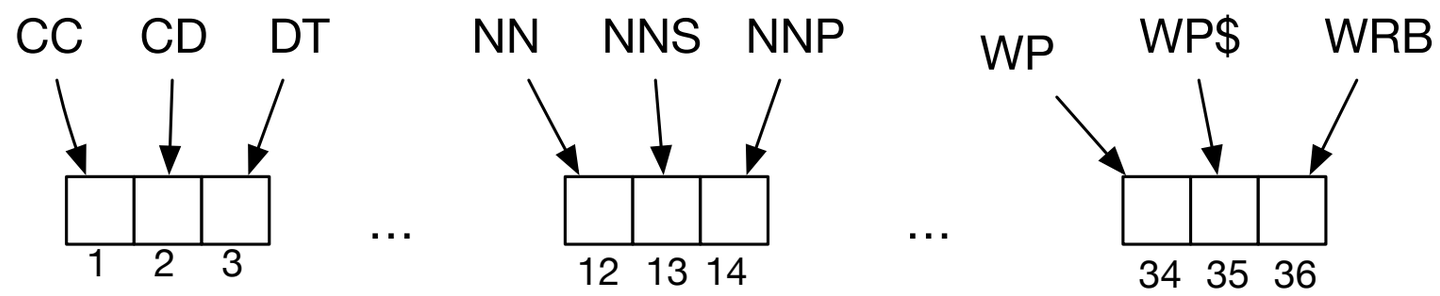
feature vector

Cosine similarity of vectors
normalised dot product =

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} \frac{a_i b_i}{n} = \frac{a_1 b_1 + a_2 b_2 + \cdots + a_n b_n}{n}$$

Part-of-Speech: one-hot-representation

- 36 tags in the Penn Treebank: 36 dimensions (one-hot representation)



CC <1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

CD <0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

DT <0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

NN <0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

NNS <0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

NNP <0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

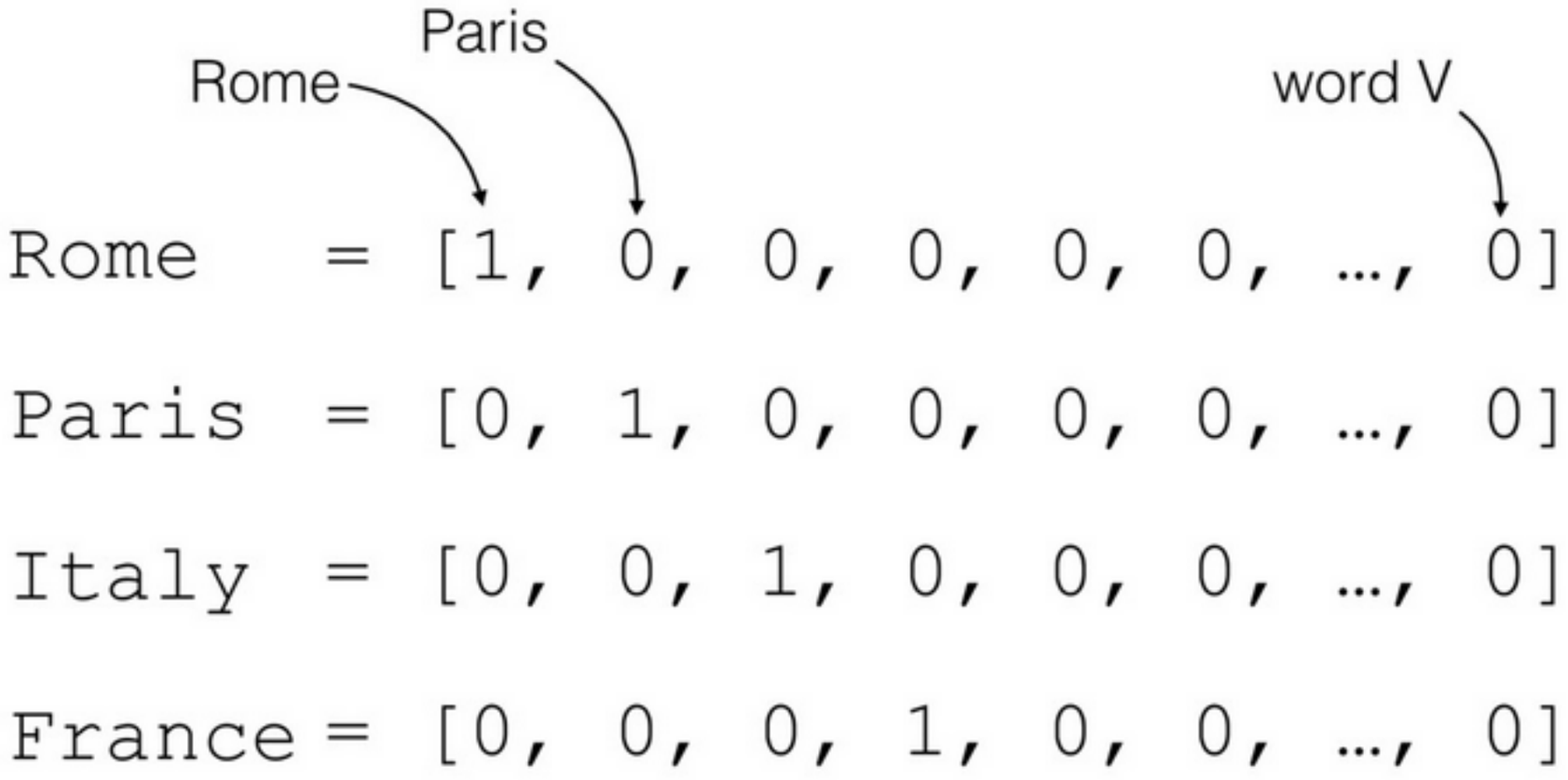
WP <0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0>

WP$ <0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0>

WRB <0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1>

Word occurrences

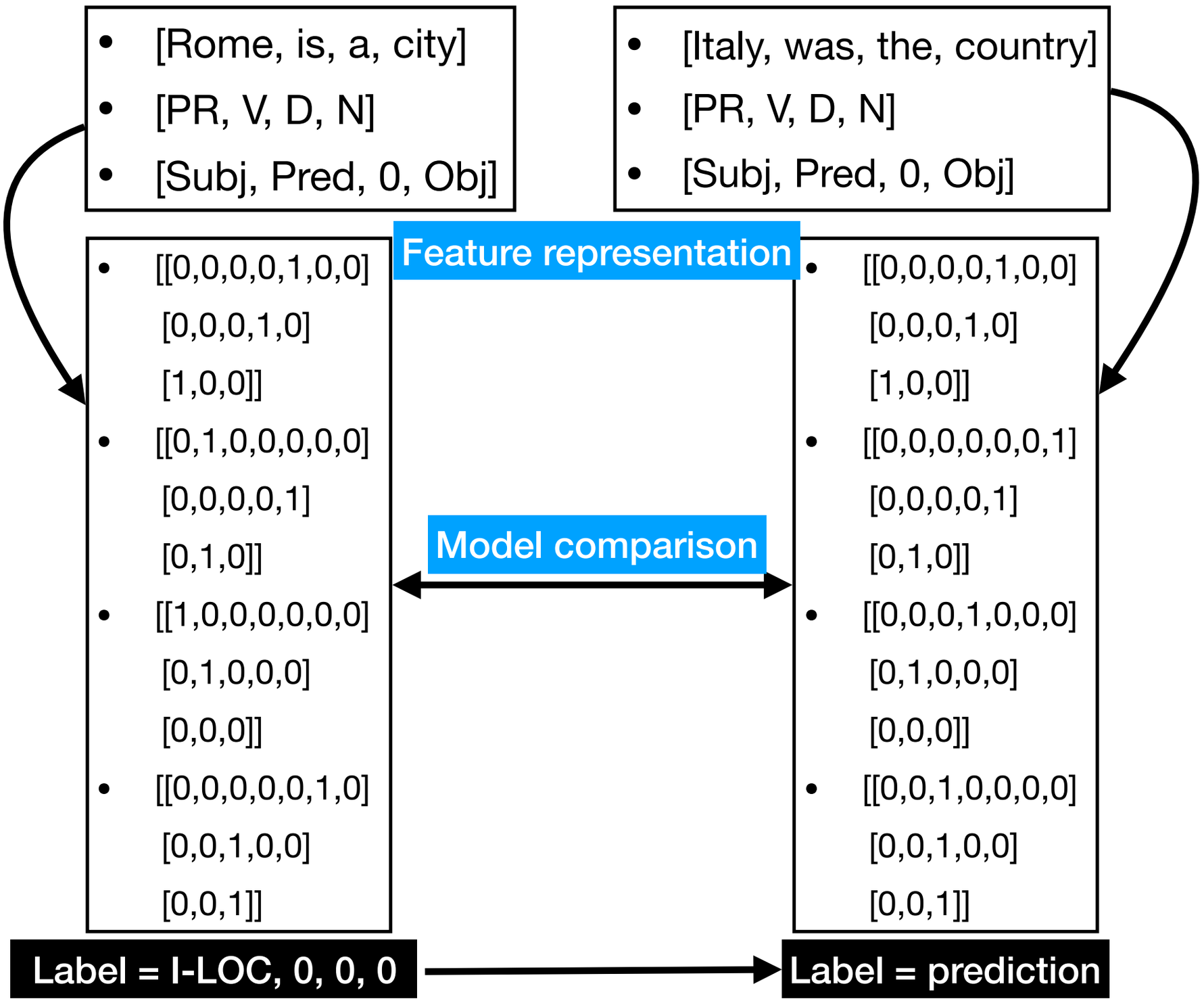
- One-hot representations: vocabulary-size dimensions



Source image: Shaffy, Athif (2017) Vector Representation of Text for Machine Learning. https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7

Machine Learning
Using One-Hot Encoding

- **Feature space**
- vocabulary = [a, is, poem, the, this, tweet, was]
- PoS = [A, D, N, PR, V]
- Dependencies = [Subj, Pred, Obj]
- Vector dimensions
  - [[0,0,0,0,0,0,0]
    [0,0,0,0,0]
    [0,0,0]]



Converting features to vectors

- "The **O** president **O** of **O** **[Apple]** **I-ORG** eats **O** an **O** apple **O**"

- Feature vector matrix representing sequences of tokens and features:

  - every vector row is a word (token)

  - every vector column is a type of feature: word, position, length, PoS, etc.

  - every cell contains a value or zero

Converting features to vectors

- [$CASE_{<0>}$, $LENGTH_{<1>}$, $POS_{<2>}$, $WORD_{<5>}$, $DISTANCE_{<6>}$,$GAZETTEER_{<7>}$]
- [true, 3, “the”, Det, -3, 0]
- [false, 9, “president”, N, -2, 0]
- [false, 2, “of ”, Prep, -1, 0]
- [true, 5, “apple”, N, 0, 1]
- [false, 4, “eats”, V, +1, 0]
- [false, 2, “an”, Det, +2, 0]
- [false, 5, “apple”, N, +3, 1]

Vectorizing values as one-hot-encoding or real values

See DictVectorizer from the Sklearn package as explained in the notebooks

- Vectorizing values: [$CASE_{<0>}$, $LENGTH_{<1>}$, $POS_{<2>}$, $WORD_{<5>}$, $DISTANCE_{<6>}$,$GAZETTEER_{<7>}$]
- The vocabulary across all documents: [a, an, apple, banana, eats, google, manager, president, of, swallows, the], mapped to some feature index: [0,1,2,3,4,5,6,7,8,9,10]
- Map every part-of-speech (PoS) to a feature index [Det, Prep, N, V, A, B] = [0,1,2,3,4,5]
- Word length vector [1,2,3,4,5,6,7,8,9,10] length, Position vector [-3,-2,-1, 0, 1, 2, 3]
- Representation of a sentence as a sequence of ***numeric*** features:
  - the [1, 3, 0, 10, -3, 0]+president [0, 9, 3, 7, -2, 0]+of [0, 2, 1, 8, -1, 0]+Apple [1, 5, 2, 2, 0, 1]+ eats [0, 4, 3, 4, 1, 0] + an [0, 2, 0, 1, 2, 0]+ apple [0, 5, 2, 2, 3, 1]
- Representation of word features as a concatenation of ***one-hot-vectors***:
  - president $[0,1]_{Case}$ + $[0,0,0,0,0,0,0,0,1,0]_{Length}$ +$[0,0,1,0,0,0]_{PoS}$ + $[0,0,0,0,0,0,1,0,0,0,0]_{Word}$ + $[0,1,0,0,0,0,0]_{Distance}$ + $[0]_{Gazetteer}$
  - president $[0,1]_{Case}$ + $[9]_{Length}$ +$[0,0,1,0,0,0]_{PoS}$ + $[0,0,0,0,0,0,1,0,0,0,0]_{Word}$ + $[3]_{Distance}$ + $[0]_{Gazetteer}$

Converting features to numeric values & vectors

- "The president of Apple eats an apple"
  - [1, 3, 0, 10, -3, 0]+[0, 9, 3, 7, -2, 0]+[0, 2, 1, 8, -1, 0]+[1, 5, 2, 2, 0, 1]+[0, 4, 3, 4, 1, 0] +[0, 2, 0, 1, 2, 0]+ [0, 5, 2, 2, 3, 1]
  - The $[1,0]_{Case}$ $[3]_{Length}$ + $[1,0,0,0,0,0]_{PoS}$ +$[0,0,00,0,0,0,0,0,0,1]_{Word}$ + $[-3]_{Distance}$ + $[0]_{Gazetteer}$
  - etc…
- "A manager of Google swallows the banana"
  - [1, 1, 0, 0, -3, 0]+[0, 7, 2, 6, -2, 0]+[0, 2, 1, 8, -1, 0]+[1, 6, 2, 5, 0, 1]+[0, 8, 3, 9, 1, 0] [0, 2, 0, 10, 2, 0]+ [0, 6, 2, 3, 3, 0]
  - A $[1,0]_{Case}$ $[1]_{Length}$ + $[1,0,0,0,0,0]_{PoS}$ +$[1,0,00,0,0,0,0,0,0,0]_{Word}$ + $[-3]_{Distance}$ + $[0]_{Gazetteer}$
  - etc.
- Use ***padding*** to make vector representations equal if the two texts have different word length: add zero vectors to the shortest one to match the longest
- ***What will happen if you replace the one-hot-word vector by word embeddings?***

Vector comparison

Word embeddings

Word is represented by context in use

- **voice**, **register** wise. The **bass range** starts at a certain **C**.
- the **alto range** is from **G3** referring to **G** below **key** of **C**
- your **family** on the **lake**, **bass fishing** is a difficult pastime
- Important information about **trout fishing** on **Lake** Taupo

vector size = vocabulary

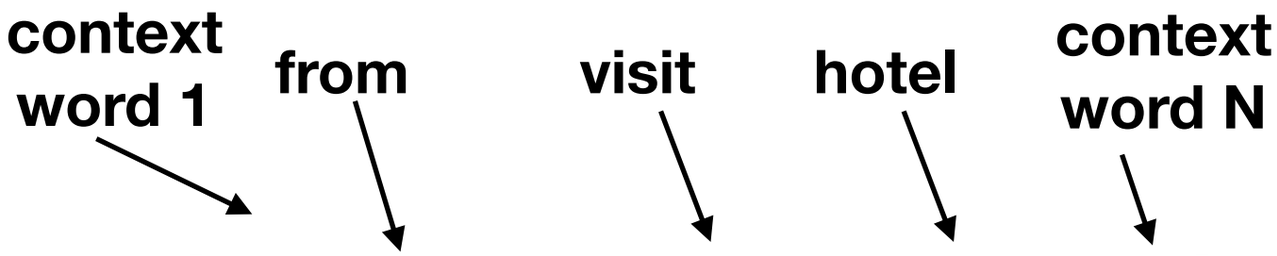
	C	family	fishing	G	G3	lake	key	range	register	voice
[	1,	0,	0,	0,	0,	0,	0,	1,	1,	1]
[	1,	0,	0,	1,	1,	0,	1,	1,	0,	0]
[	0,	1,	1,	0,	0,	1,	0,	0,	0,	0]
[	0,	0,	1,	0,	0,	1,	0,	0,	0,	0]

similarity of word context is the normalised dot product of the vectors =

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} \frac{a_i b_i}{n} = \frac{a_1 b_1 + a_2 b_2 + \cdots + a_n b_n}{n}$$

Embeddings represent families of related words as features

Lengths of the vocabulary is the size of the embeddings



Rome = [0.1, 0.4, 0.3, 0.9, 0.1, 0.8, 0.1]

Paris = [0.1, 0.3, 0.4, 0.8, 0.1, 0.9, 0.1]

Italy = [0.2, 0.7, 0.3, 0.5, 0.1, 0.4, 0.1]

France = [0.2, 0.7, 0.3, 0.5, 0.1, 0.4, 0.1]

Using GenSim and GloVe to represent words

```
gensimmodel.most_similar('berlin')
```

```
[('munich', 0.665853972033691),
 ('hamburg', 0.6416077017784119),
 ('stuttgart', 0.602530002593941),
 ('germany', 0.597375750541687),
 ('nationalgalerie', 0.596231520175938),
 ('münchner', 0.594410419464113),
 ('leipzig', 0.588937163352963),
 ('charlottenburg', 0.583330333232796),
 ('bochum', 0.580147862434387),
 ('düsseldorf', 0.580057740211486)]
```

```
gensimmodel.most_similar('rome')
```

```
[('viterbo', 0.600153028964996),
 ('ravenna', 0.600064635276794),
 ('italy', 0.592195272445678),
 ('aquileia', 0.58372366428375),
 ('perugia', 0.581674814224243),
 ('civitavecchia', 0.578354597091675),
 ('anagni', 0.572831213474274),
 ('tarentum', 0.570874810218811),
 ('fiesole', 0.561999797821045),
 ('vigilius', 0.560334444046021)]
```

```
gensimmodel.get_vector('rome')
```

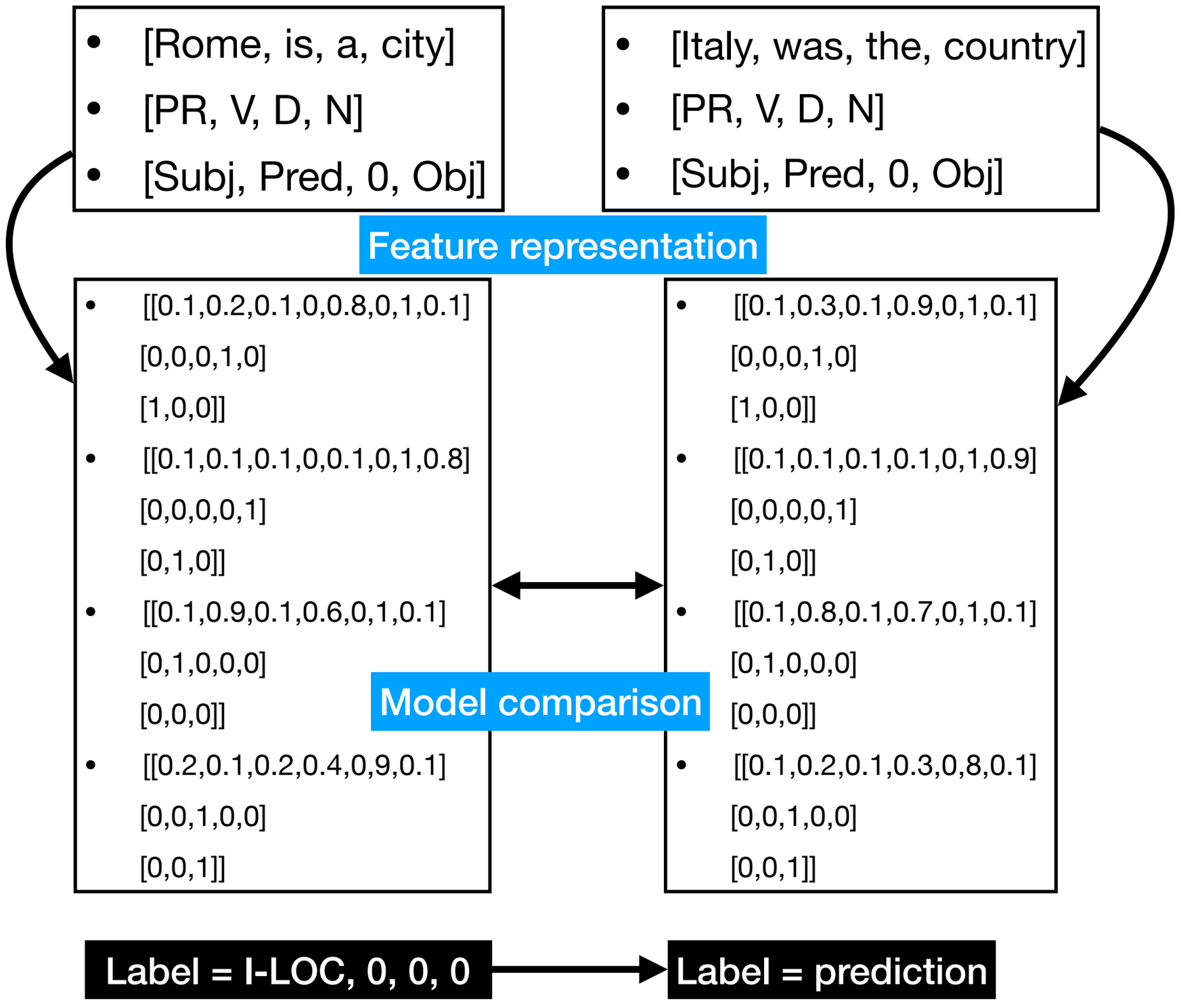
```
array([-2.79722e-01,  4.18270e-02, -1.84113e-01,
-1.74999e-01,
       3.41771e-01, -2.45552e-01, -9.80010e-02, -5.23456e-01,
       1.25658e-01,  5.60530e-02, -2.36231e-01,  1.59386e-01,
      -7.44650e-02, -1.02767e-01,  2.82295e-01,  5.69800e-02,
       1.49481e-01, -1.35064e-01, -7.17500e-02,  4.90650e-02,
      -2.56283e-01,  3.47652e-01, -6.65500e-03, -3.70058e-01,
      -1.30084e-01, -5.62090e-02, -5.44500e-03, -2.89338e-01,
      -2.74530e-01, -5.11710e-02, -9.45050e-02, -3.46780e-02,
      -3.86790e-02, -1.21387e-01,  6.13910e-02, -2.72660e-02,
      -1.51958e-01, -1.80081e-01,  9.48250e-02,  1.23013e-01,
       1.92840e-01, -2.00165e-01,  2.13930e-01,  1.46464e-01,
      -2.79190e-01, -1.09058e-01,  1.18088e-01, -2.45300e-02,
etc…. 400 dominations
```

```
gensimmodel.get_vector('rome').size = 400
```

Machine Learning Using Word Embeddings

- **Feature space**
- vocabulary = [embedding dimensions]
- PoS = [A, D, N, PR, V]
- Dependencies = [Subj, Pred, Obj]
- Vector dimensions
  - [[0,0,0,0,0,0,0,0]
    [0,0,0,0,0]
    [0,0,0]]

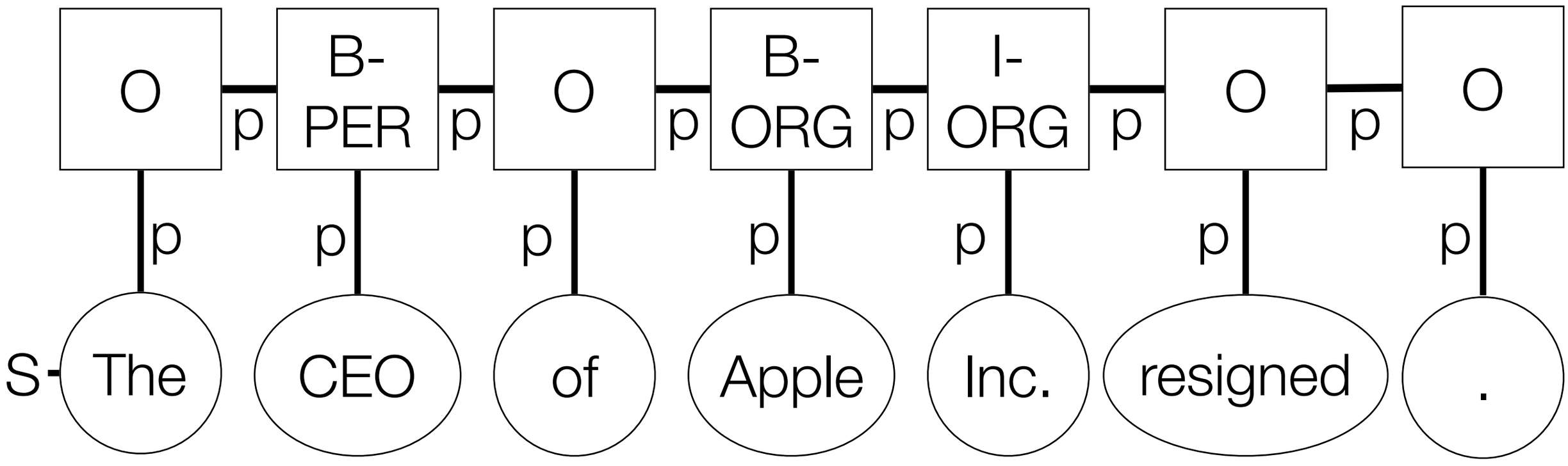
Size of embeddings, e.g. 400 dimensions



State of the art systems

- CRFs (Conditional Random Fields) are one of the most widely used algorithms for NERC (Stanford NERC)
    - Graphical models, view NERC as a sequence labelling task
    - Named entities consist of a beginning token (B), inside tokens (I), and outside tokens (O)
        - Abraham (B-PER) Lincoln (I-PER) ((O) February (B-T) 12 (I-T) , (I-T) 1809 (I-T)
    - Strong dependence between features and sequence, e.g. I-LOC never occurs immediately after B-PER
- Recurrent neural networks, Long-Short-Term-Memory (LSTM)
- https://www.quora.com/What-are-the-pros-and-cons-of-these-three-sequence-models-MaxEnt-Markov-Model-Conditional-random-fields-and-

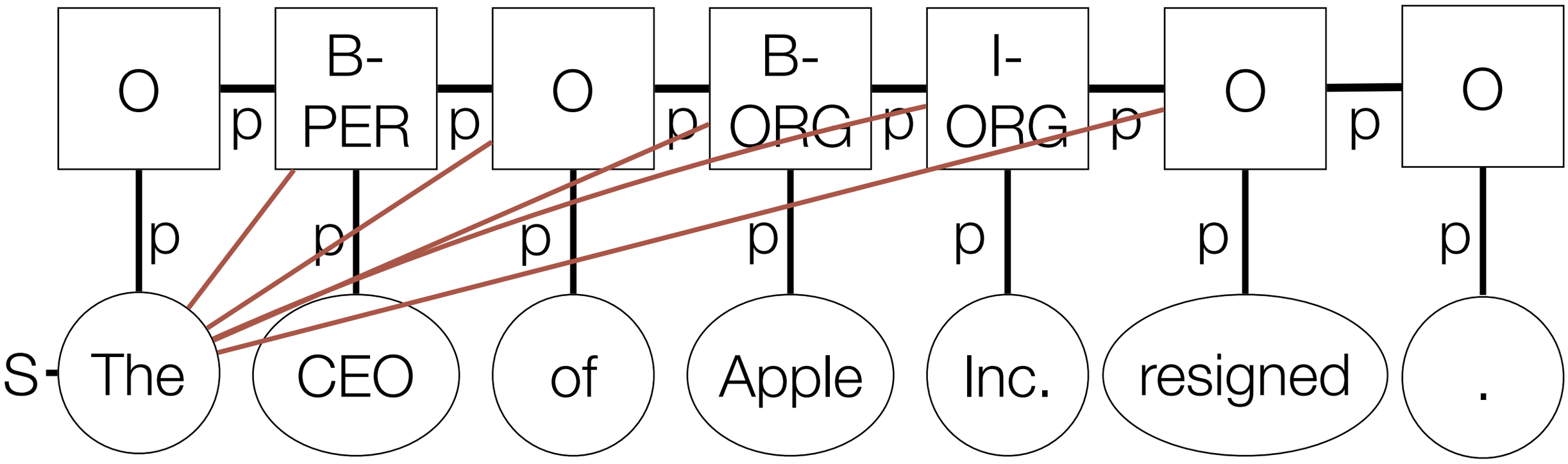
Conditional Random Field for IOB sequences



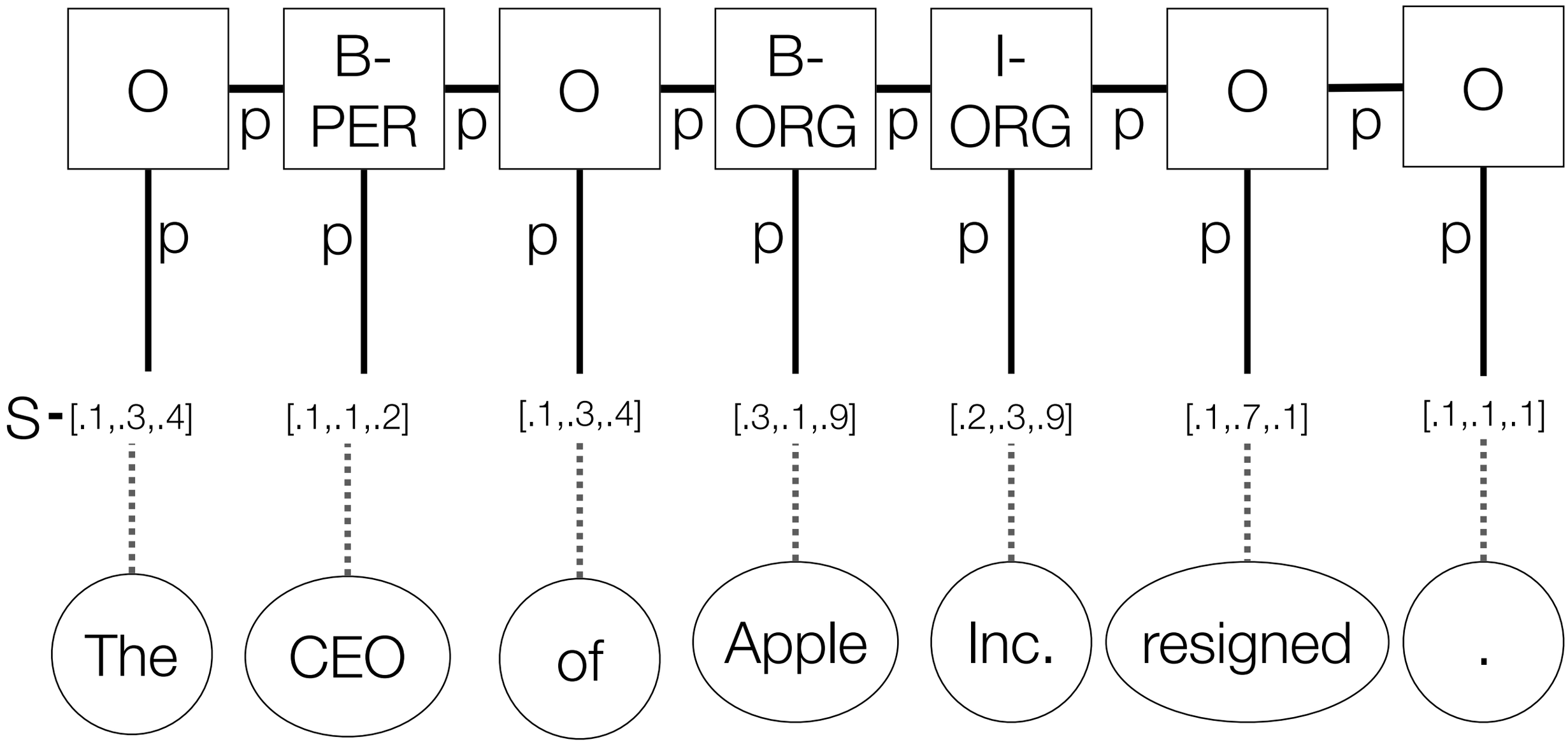
if Initial Capital preceded by O then B

if Initial Capital followed by “Inc.” then ORG

Conditional Random Field for IOB sequences



Conditional Random Field for IOB sequences



Neural network (LSTM) and CRF

https://github.com/guillaumegenthial/tf_ner

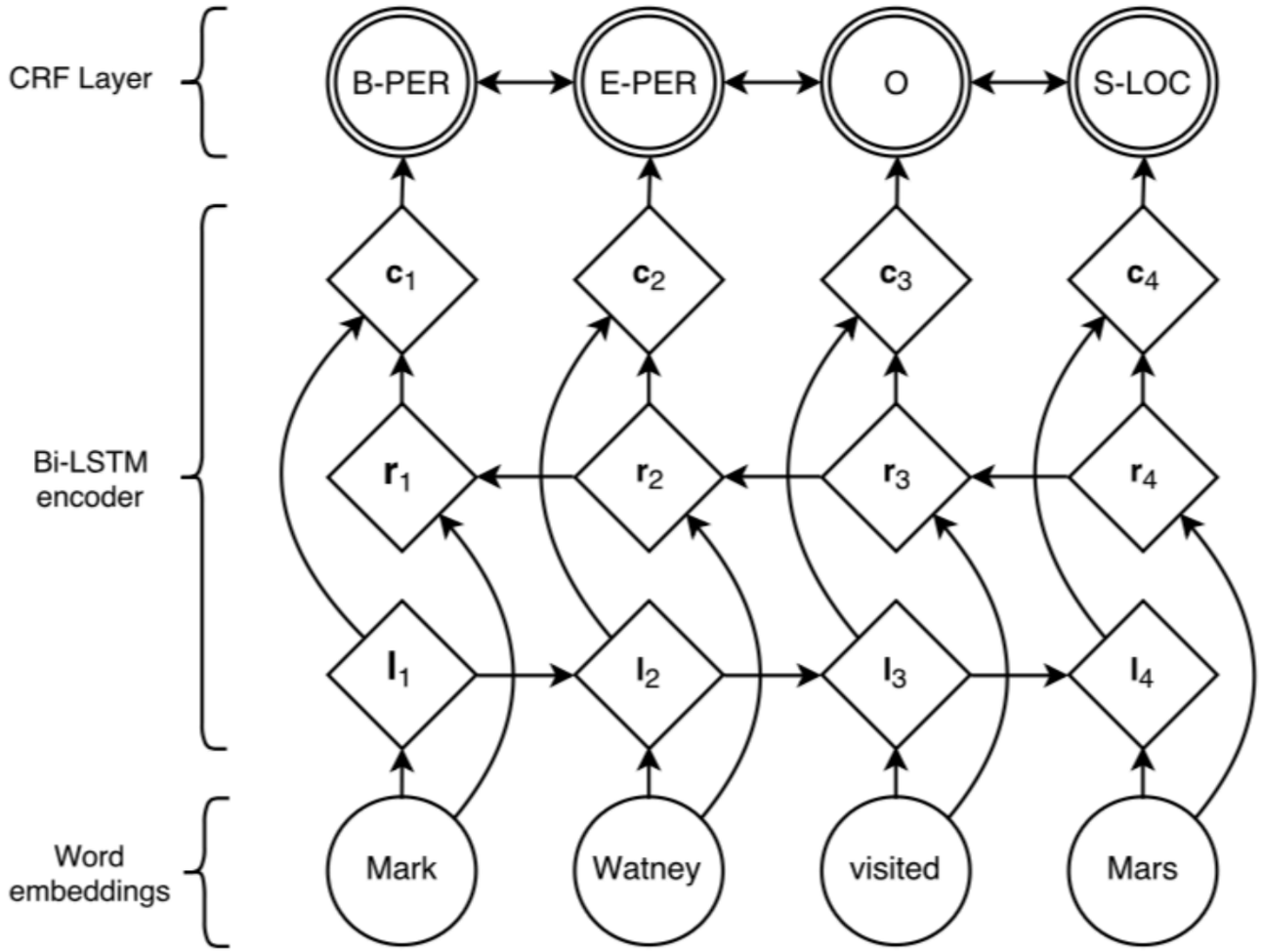


Figure 1: Main architecture of the network. Word embeddings are given to a bidirectional LSTM. $\mathbf{l}_i$ represents the word i and its left context, $\mathbf{r}_i$ represents the word i and its right context. Concatenating these two vectors yields a representation of the word i in its context, $\mathbf{c}_i$.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer, 2016, Neural Architectures for Named Entity Recognition, NAACL.

Long Short-Term Memory LSTM

Zhiheng Huang, Wei Xu, Kai Yu 2015, Bidirectional LSTM-CRF Models for Sequence Tagging, arXiv.1508.01991v1

NERC performance

Feature-engineered machine learning systems	Dict	SP	DU	EN	GE
Carreras et al. (2002) binary AdaBoost classifiers	Yes	81.39	77.05	-	-
Malouf (2002) - Maximum Entropy (ME) + features	Yes	73.66	68.08	-	-
Li et al. (2005) SVM with class weights	Yes	-	-	88.3	-
Passos et al. (2014) CRF	Yes	-	-	90.90	-
Ando and Zhang (2005a) Semi-supervised state of the art	No	-	-	89.31	75.27
Agerri and Rigau (2016)	Yes	**84.16**	**85.04**	**91.36**	**76.42**
Feature-inferring neural network word models					
Collobert et al. (2011) Vanilla NN +SLL / Conv-CRF	No	-	-	81.47	-
Huang et al. (2015) Bi-LSTM+CRF	No	-	-	84.26	-
Yan et al. (2016) Win-BiLSTM (English), FF (German) (Many fets)	Yes	-	-	88.91	**76.12**
Collobert et al. (2011) Conv-CRF (SENNA+Gazetteer)	Yes	-	-	89.59	-
Huang et al. (2015) Bi-LSTM+CRF+ (SENNA+Gazetteer)	Yes	-	-	**90.10**	-
Feature-inferring neural network character models					
Gillick et al. (2015) – BTS	No	**82.95**	**82.84**	**86.50**	**76.22**
Kuru et al. (2016) CharNER	No	82.18	79.36	84.52	70.12
Feature-inferring neural network word + character models					
Yang et al. (2017)	Yes	85.77	**85.19**	91.26	-
Luo (2015)	Yes	-	-	91.20	-
Chiu and Nichols (2015)	Yes	-	-	**91.62**	-
Ma and Hovy (2016)	No	-	-	91.21	-
Santos and Guimaraes (2015)	No	82.21	-	-	-
Lample et al. (2016)	No	85.75	81.74	90.94	**78.76**
Bharadwaj et al. (2016)	Yes	**85.81**	-	-	-
Dernoncourt et al. (2017)	No	-	-	90.5	-
Feature-inferring neural network word + character + affix models					
Re-implementation of Lample et al. (2016) (100 Epochs)	No	85.34	85.27	90.24	78.44
Yadav et al. (2018)(100 Epochs)	No	86.92	87.50	90.69	78.56
Yadav et al. (2018) (150 Epochs)	No	**87.26**	**87.54**	90.86	**79.01**

Table 1: Comparison of NER systemsin four languages: CoNLL 2002 Spanish (SP), CoNLL 2002 Dutch (DU), CoNLL 2003 English (EN), and CoNLL 2003 German (GE). Dict indicates whether or not the approach makes use of dictionary lookups. Best performance in each category is highlighted in bold.

Yadav, Vikas, and Steven Bethard. "A survey on recent advances in named entity recognition from deep learning models." In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2145-2158. 2018.

Feature-inferring NN systems outperform feature-engineered systems, despite the latter's access to domain specific rules, knowledge, features, and lexicons

Bidirectional LSTM models for NERC

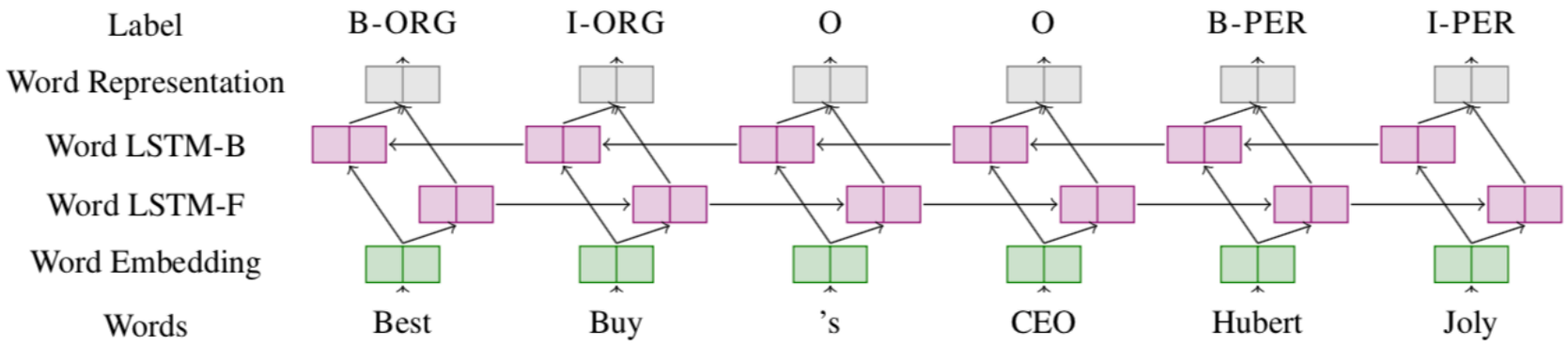


Figure 1: Word level NN architecture for NER

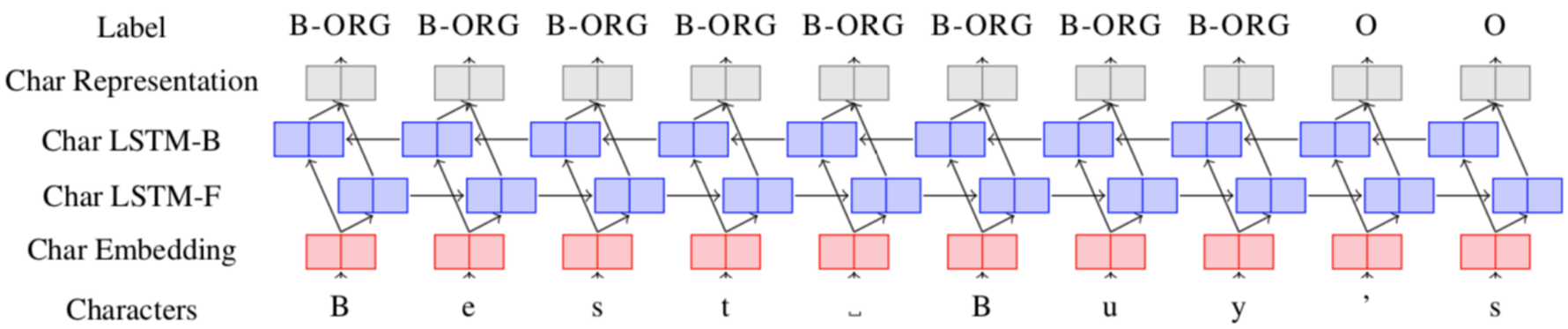


Figure 2: Character level NN architecture for NER

Yadav & Bethard 2018

Bidirectional LSTM models for NERC

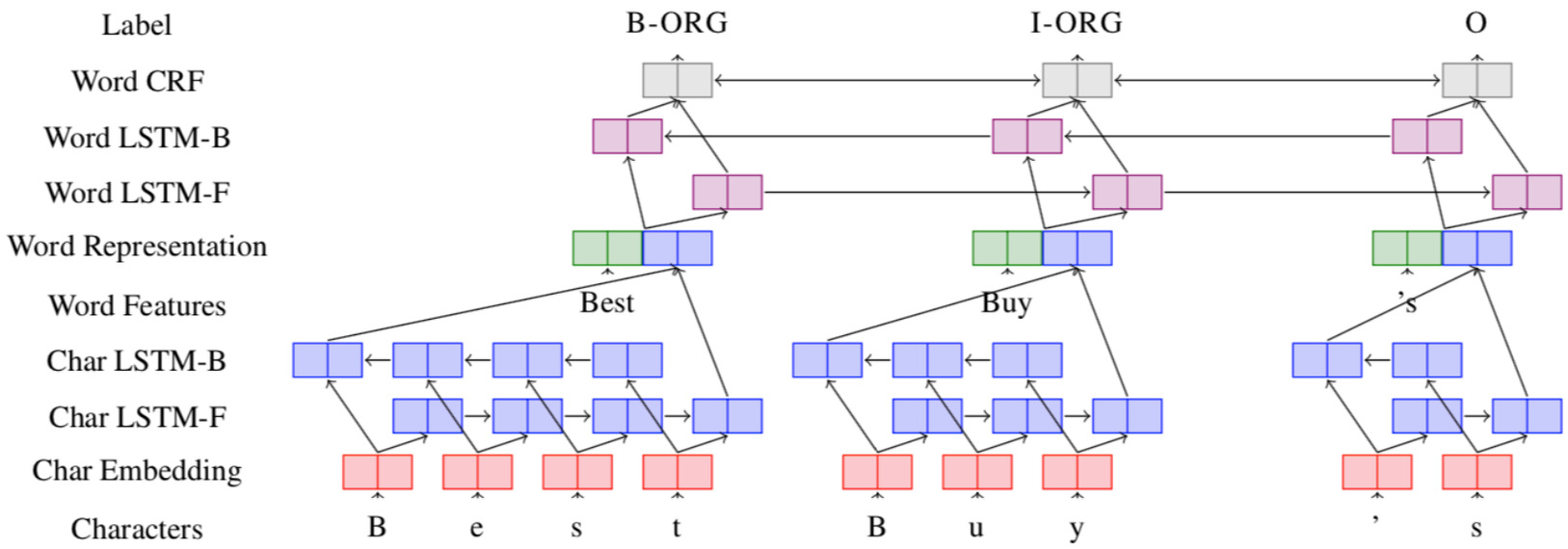


Figure 3: Word+character level NN architecture for NER

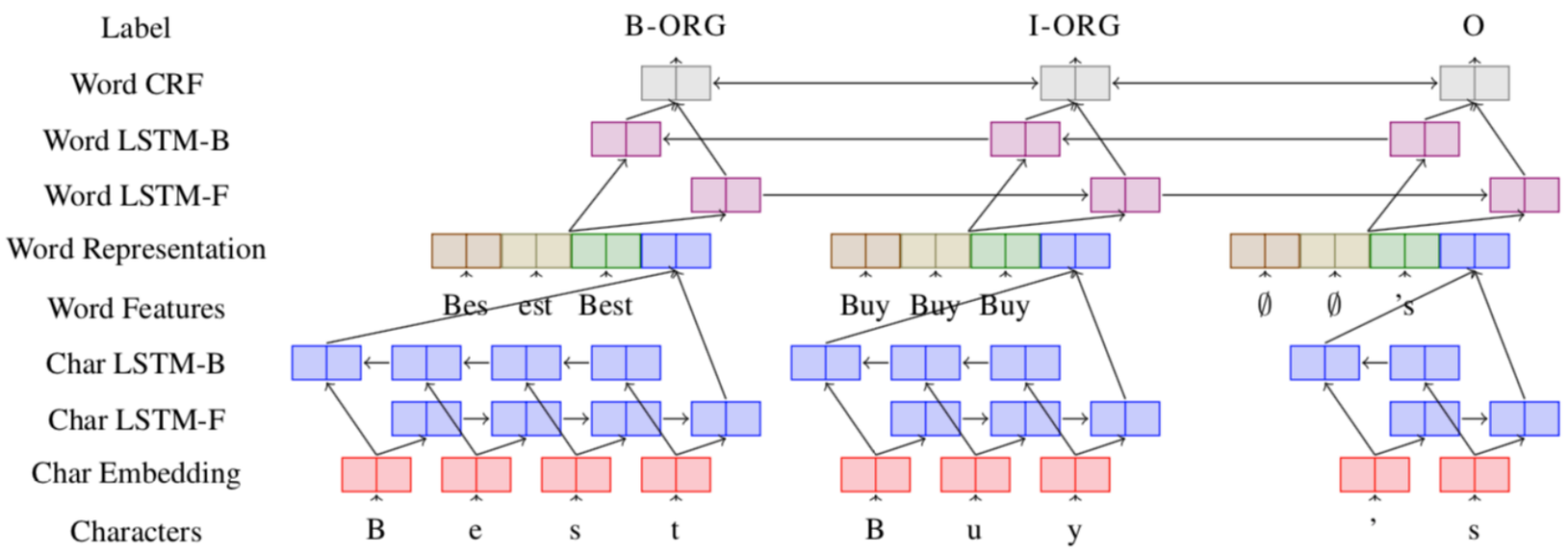


Figure 4: Word+character+affix level NN architecture for NER

Affix embeddings from all n-gram prefixes and suffixes of words in the training corpus

Yadav & Bethard 2018

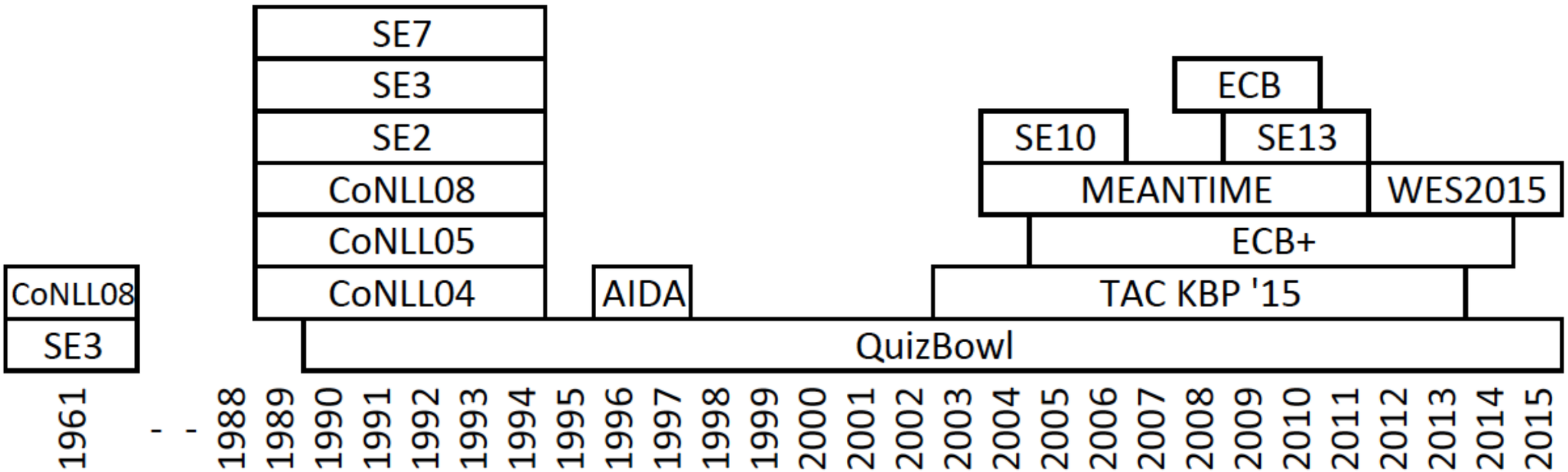
Factors that impact performance for NERC

- The annotation of the spans, annotation of nesting
  - [[[White House] [press] secretary] Scott McClellan]
  - [The [CEO] of the [US]-Based company [Facebook]]
- Type of text: news or tweets/ social media
- Entity types: people, organisations, amounts, dates, events
- Amount of training data
- Difference between training data and test data:

Measuring performance for entities

- Is simple precision and recall enough?
- Neil Young & Crazy Horse
- Score per chunk (only give a true positive score if the entire NE is correctly classified)
    - Here “Neil” gets a false negative score, “&” gets a false positive score, “Horse” again a false negative.
- Some other metrics exist (e.g., MUC) that give partial credit (complex rules)

How specific is our data?



Ilievski, Postma, and Vossen, Semantic overfitting: what `world' do we consider when evaluating disambiguation of text? COLING 2016.

What is the effect of gazetteers on data over time?

Performance drops when shifting data

Agerri and Rigau 2016

Table 6: NERC CoNLL 2003 testb results.

	Precision	Recall	F1
Newsreader (ixa-pipe-nerc)	91.64	90.21	**90.92**
Stanford NER (CRF)	-	-	88.08
Ratinov et al. (2009)	-	-	90.57
Passos et al. (2014)	-	-	90.90

Table 7: NERC Intra-document Benchmarking with Wikinews.

System	mention extent	Precision	Recall	F1
Newsreader (ixa-pipe-nerc)	Inner phrase-based	62.15	76.06	**68.41**
Stanford NER (all english crf distsim)	Inner phrase-based	63.53	68.21	65.79
Newsreader (ixa-pipe-nerc)	Inner token-based	72.17	79.31	**75.57**
Stanford NER (all english crf distsim)	Inner token-based	77.14	71.77	74.36
Newsreader (ixa-pipe-nerc)	Outer phrase-based	53.01	68.03	**59.59**
Stanford NER (all english crf distsim)	Outer phrase-based	52.86	59.51	55.99
Newsreader (ixa-pipe-nerc)	Outer token-based	73.40	67.20	**70.16**
Stanford NER (all english crf distsim)	Outer token-based	78.22	60.63	68.31

P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A. Minard, A. P. Aprosio, G. Rigau, M. Rospocher, and R. Segers, “NewsReader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news”, Special issue knowledge-based systems, elsevier, 2016. dx.doi.org/10.1016/j.knosys.2016.07.013

Domain specific NER

	Dict	MedLine (80.10%)			DrugBank (19.90%)			Complete dataset		
		P	R	F1	P	R	F1	P	R	F1
Feature-engineered machine learning systems										
Rocktäschel et al. (2013)	Yes	60.70	55.80	58.10	88.10	87.50	87.80	73.40	69.80	71.50
Liu et al. (2015) (baseline)	No	-	-	-	-	-	-	78.41	67.78	72.71
Liu et al. (2015) (MED. emb.)	No	-	-	-	-	-	-	82.70	69.68	75.63
Liu et al. (2015) (state of the art)	Yes	78.77	60.21	68.25	90.60	88.82	**89.70**	84.75	72.89	**78.37**
NN word model										
Chalapathy et al. (2016) (relaxed performance)	No	52.93	52.57	52.75	87.07	83.39	85.19	-	-	-
NN word + character model Yadav et al. (2018)	No	73	62	67	87	86	87	79	72	75
NN word + character + affix model Yadav et al. (2018)	No	74	64	**69**	89	86	87	81	74	77

91+ on CoNLL 2003

90+ on CoNLL 2003

Table 2: DrugNER results on the MedLine and DrugBank test data (80.10% and 19.90% of the test data, respectively). The Yadav et al. (2018) experiments report no decimal places because they were run after the end of shared task, and the official evaluation script outputs no decimal places.

Yadav & Bethard 2018

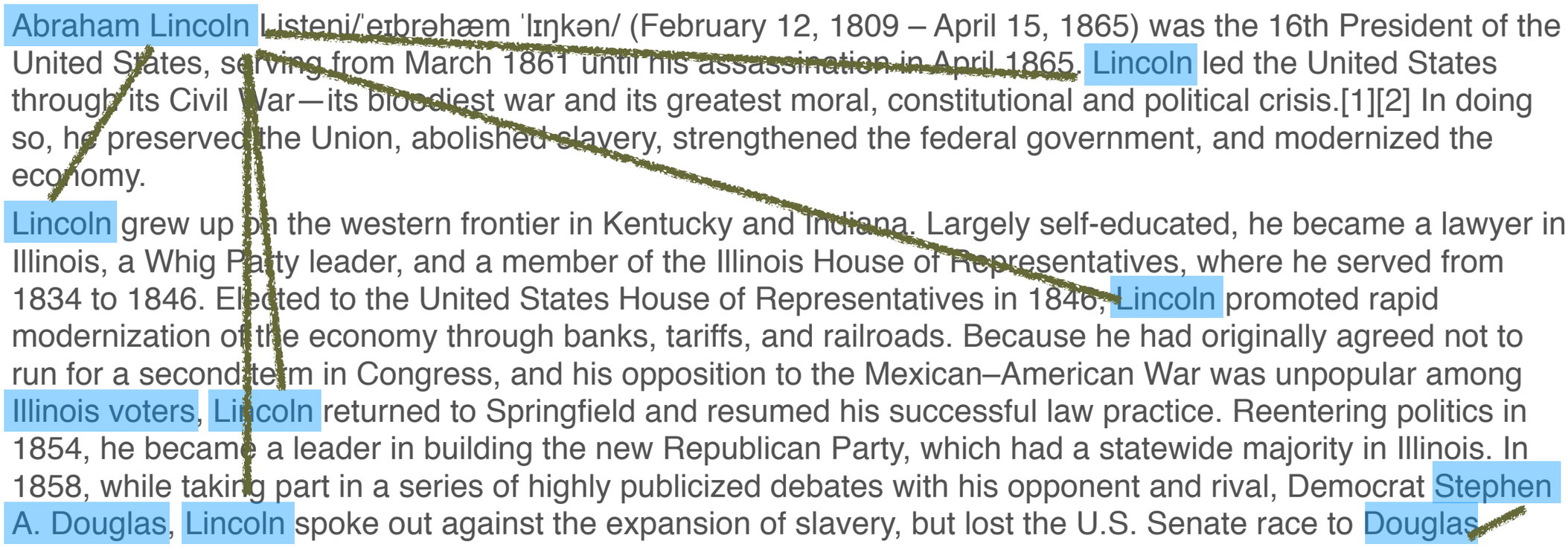
There is more than named entity expressions

- Identities: people with the same name (Joe Smith) are not necessarily people with the same identity

- Coreference: also phrases (“the president”) and pronouns (“he”, “she”) can make reference to the same entity

What is Coreference Resolution

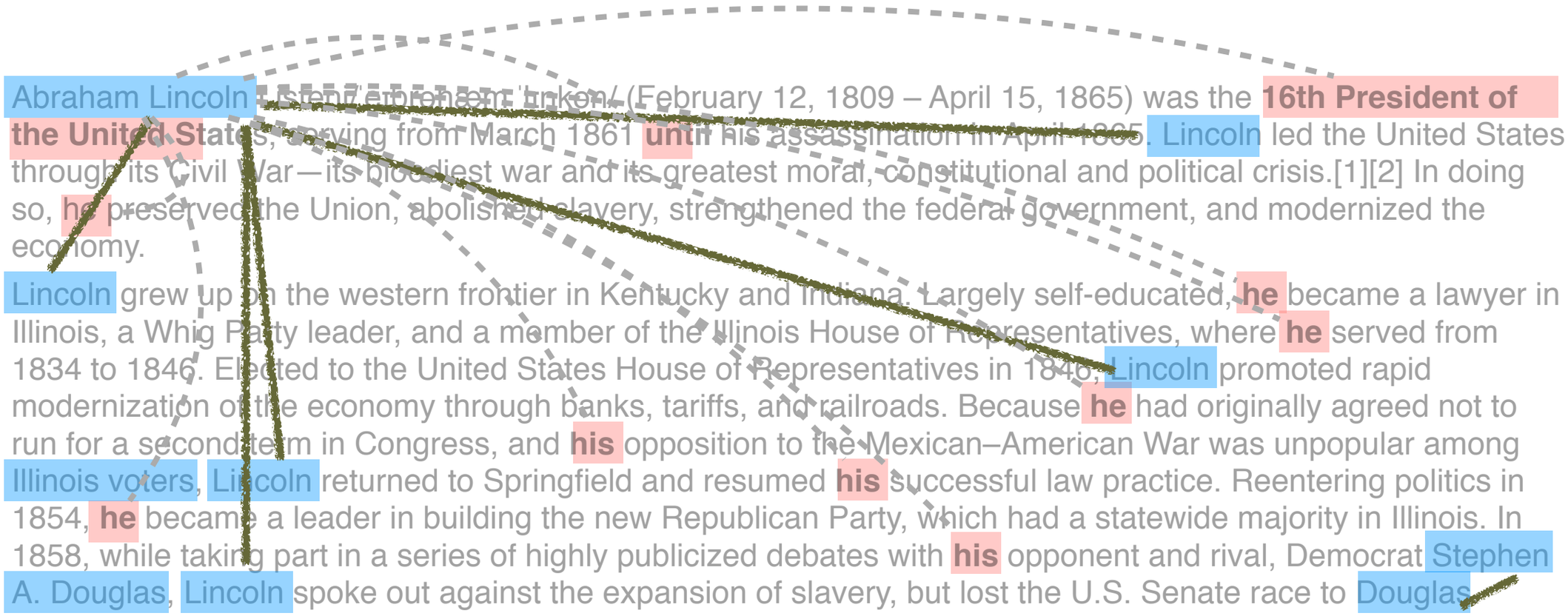
- Coreference resolution is the task of finding out which words/phrases refer to the same entity



source: http://en.wikipedia.org/wiki/Abraham_Lincoln

But it's actually more complicated

- Coreference resolution is the task of finding out which words/phrases refer to the same object



source: http://en.wikipedia.org/wiki/Abraham_Lincoln

But it’s actually more complicated

- Coreference resolution is the task of finding out which words/phrases refer to the same object

Abraham Lincoln Listeni/ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis.[1][2] In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy.

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he had originally agreed not to run for a second term in Congress, and his opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln returned to Springfield and resumed his successful law practice. Reentering politics in 1854, he became a leader in building the new Republican Party, which had a statewide majority in Illinois. In 1858, while taking part in a series of highly publicized debates with his opponent and rival, Democrat Stephen A. Douglas, Lincoln spoke out against the expansion of slavery, but lost the U.S. Senate race to Douglas.

source: http://en.wikipedia.org/wiki/Abraham_Lincoln

How to do coreference resolution

Antecedent	Anaphor	Corefers?
Abraham Lincoln	He$_1$	yes
16th president of the United States	He$_1$	yes
Lincoln	His$_5$	yes
Stephen A. Douglas	He$_4$	no
Abraham Lincoln	Lincoln$_6$	yes
Member of the Illinois House of Representatives	He$_2$	yes

Lincoln grew up on the western frontier in Kentucky and Indiana. Largely self-educated, he$_1$ became a lawyer in Illinois, a Whig Party leader, and a member of the Illinois House of Representatives, where he$_2$ served from 1834 to 1846. Elected to the United States House of Representatives in 1846, Lincoln$_3$ promoted rapid modernization of the economy through banks, tariffs, and railroads. Because he$_4$ had originally agreed not to run for a second term in Congress, and his$_5$ opposition to the Mexican–American War was unpopular among Illinois voters, Lincoln$_6$ returned to Springfield and resumed his$_7$ successful law practice.

Why is coreference resolution important?

- Coreference is a frequently used natural language phenomenon
- Coreference resolution is essential to aggregate all knowledge and properties of the entities that a text makes reference to
- Coreference resolution is difficult because it stretches across sentences (syntax) and involves semantics and discourse

How to do coreference resolution (1)

Rule-based (Stanford multi-sieve, Lee eval 2013)

1. String matching

   - Will help you with proper names (*Smith* & *Smith*), common NPs (risky): *a man*, *another man*, *the man*
   - Partial matching is a problem (with/without titles?)
   - Fails on abbreviations and acronyms (and anything that doesn't use the same strings, e.g. *Lincoln, he*)

2. Agreement heuristics (anaphora must agree with their antecedents in name, gender and animacy)

3. Scoping (identify a text region where you expect to find the antecedent): Most recent matching subject is the most likely antecedent:

   - *John gave Bill a book. He ….*
   - *John gave Mary a book, She…*
   - *John gave Bill a book. Bill did not read it/ He did not read it/He asked him about the title*

How to do coreference resolution (2)

Machine Learning

- Annotated data marked up with co-reference chains
- Supervised technique to identify antecedents to anaphora
    - Clustering to merge pairwise coreference decisions into coreference chains
- Varied features used: part-of-speech tags, parse information, named entities, semantic class lookup, NP chunks, proximity, aliases, number, gender

Featured used for entity-coreference resolution

Type	Features
Mention	String match, part-of-speech, alias, number, gender (Soon, Ng, and Lim 2001), appositive, animacy, speaker (Lee et al. 2011), WordNet relation (Culotta, Wick, and McCallum 2007), modifier (Culotta, Wick, and McCallum 2007), overlap, quotation (Ng and Cardie 2002b), syntax subtree (Versley et al. 2008), dependency label (Björkelund and Nugues 2011), dependency path (Bergsma and Lin 2006), named-entity type (Denis and Baldridge 2009), semantic class (Soon, Ng, and Lim 2001), selectional preference (Dagan, Dagan, and Itai 1990), semantic roles (Ponzetto et al. 2006)
Textual context	Saliency (Lappin and Leass 1994), recency McCarthy (1996, pp. 87), narrative chain (Rahman and Ng 2012; Peng and Roth 2016)
Entity linking	Wikipedia (Ponzetto et al. 2006), Freebase attribute (Hajishirzi et al. 2013)

Table 1: A non-comprehensive list of features used in the literature. Each feature can be instantiated in many ways and sometimes one system contains more than one version.

Minh, Fokkens, Vossen, (under review), A Content-based Analysis of Coreference Resolution

Coreference performance

- CoNLL-2012 (Pradhan et al. 2012) standard benchmark in entity coreference resolution in recent years.:
    - 2,385 annotated English documents, totaling at 1.6M words, from various genres such as newswire, weblogs, and telephone conversations.
    - Highest reported result (after six years) is only 73% (Lee, He, and Zettlemoyer 2018).
- Some genres get much lower performance than others
    - Stanford Sieve (Lee et al. 2013) is lowest for newswire (55%) and highest for bible (67%).
    - The neural model of Clark and Manning (2016b) displays more than 10% difference between broadcast conversations (64%) and bible (78%).
- References:

    - Clark, Kevin and Christopher D. Manning. 2016a. Deep Reinforcement Learning for Mention-Ranking Coreference Models. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16), pages 2256–2262.
    - Clark, Kevin and Christopher D. Manning. 2016b. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 643–653.
    - Lee, Heeyoung, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. Computational Linguistics, 39(4):885–916.
    - Lee, Kenton, Luheng He, and Luke Zettlemoyer. 2018. Higher-order Coreference Resolution with Coarse-to-fine Inference. pages 687–692.
    - Pradhan, Sameer, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. EMNLP-CoNLL 2012.
