CS447: Natural Language Processing

http://courses.engr.illinois.edu/cs447

Lecture 24: Information Extraction

(Sequence Labeling, Named Entity Recognition
Relation Extraction)

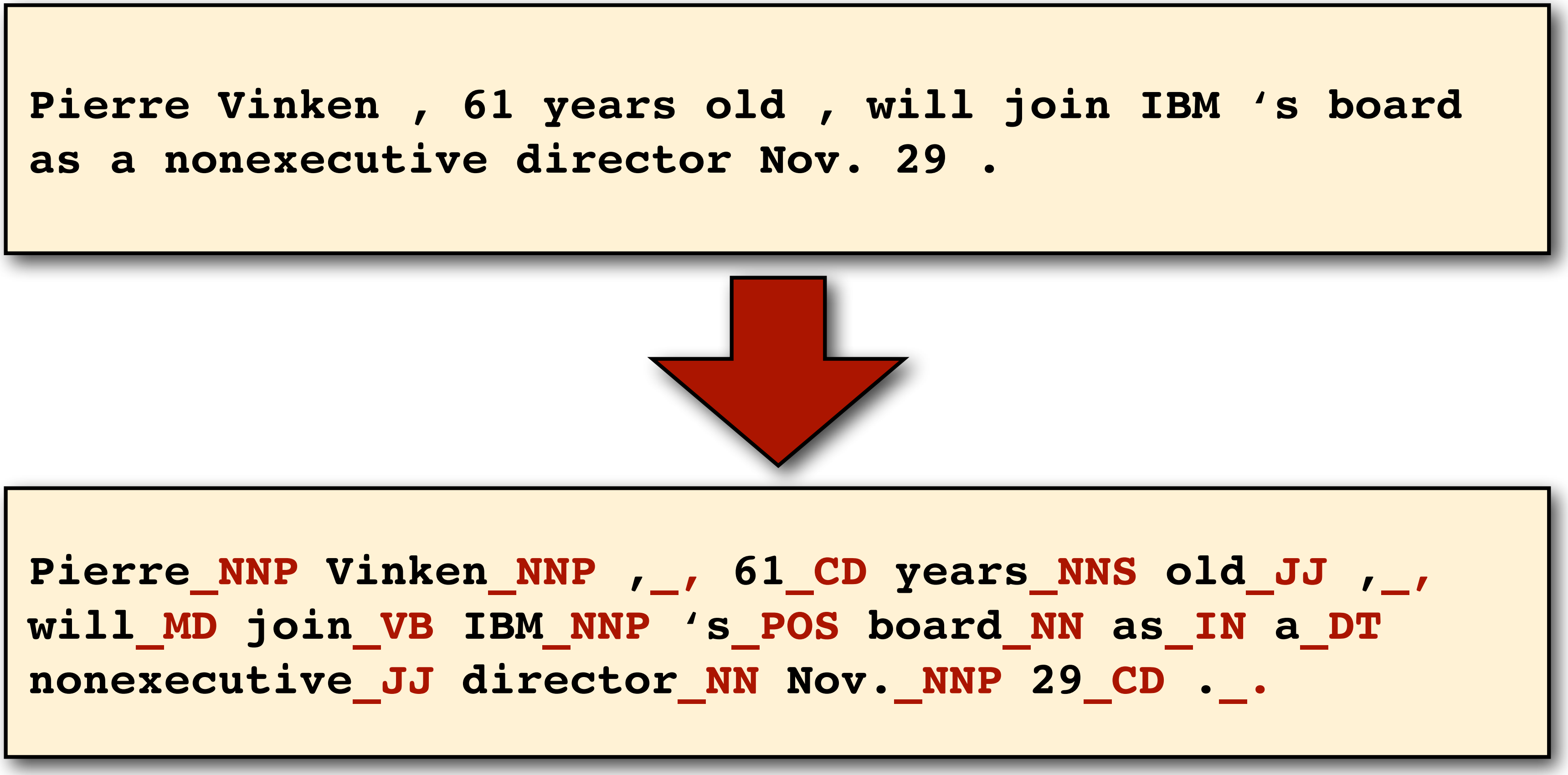
Julia Hockenmaier

juliahmr@illinois.edu

3324 Siebel Center

Sequence Labeling

POS tagging

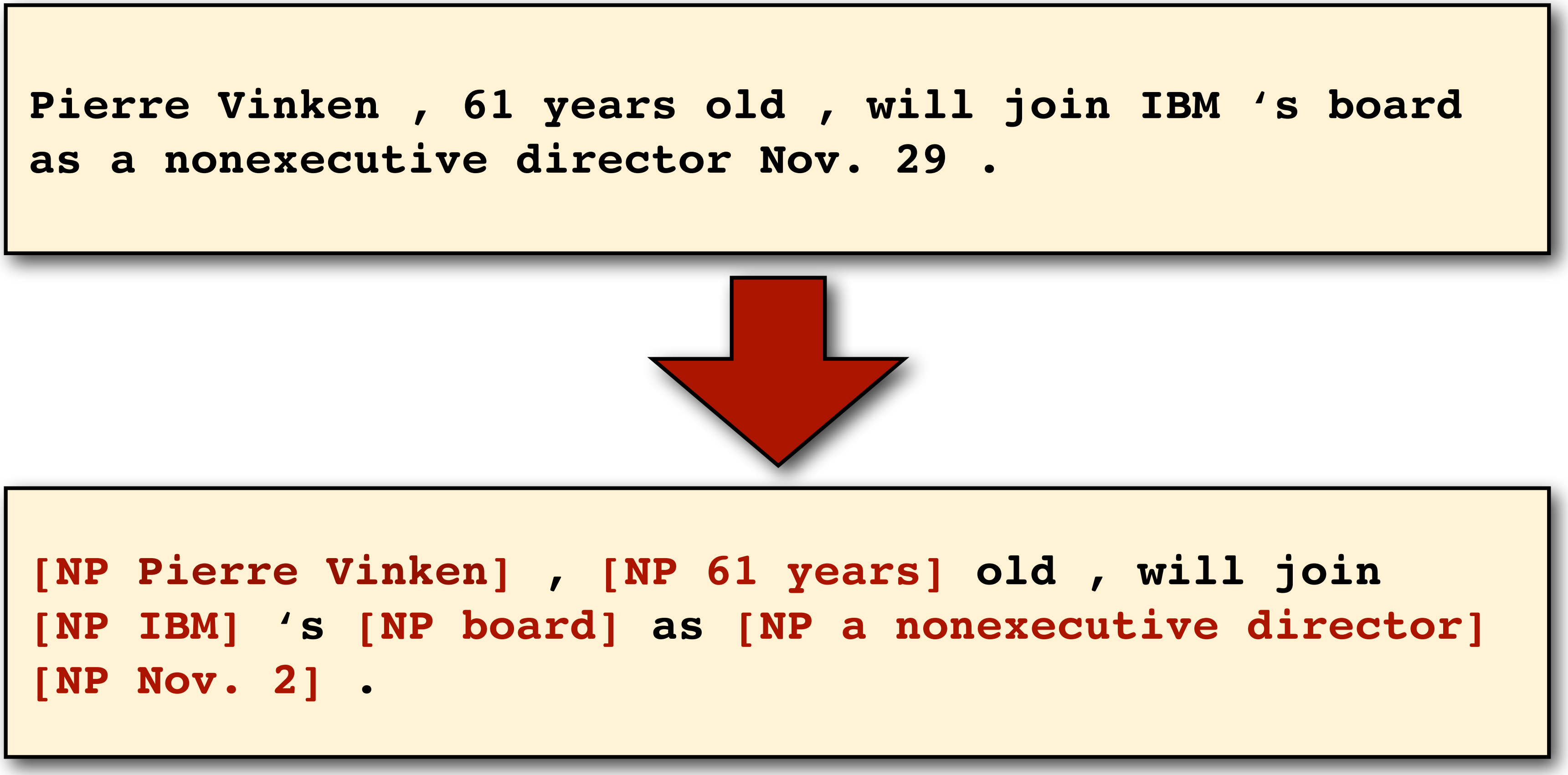


```
Pierre Vinken , 61 years old , will join IBM ‘s board
as a nonexecutive director Nov. 29 .
```

```
Pierre_NNP Vinken_NNP ,_, 61_CD years_NNS old_JJ ,_,
will_MD join_VB IBM_NNP ‘s_POS board_NN as_IN a_DT
nonexecutive_JJ director_NN Nov._NNP 29_CD ._.
```

Task: assign POS tags to words

Noun phrase (NP) chunking



Pierre Vinken , 61 years old , will join IBM ‘s board
as a nonexecutive director Nov. 29 .

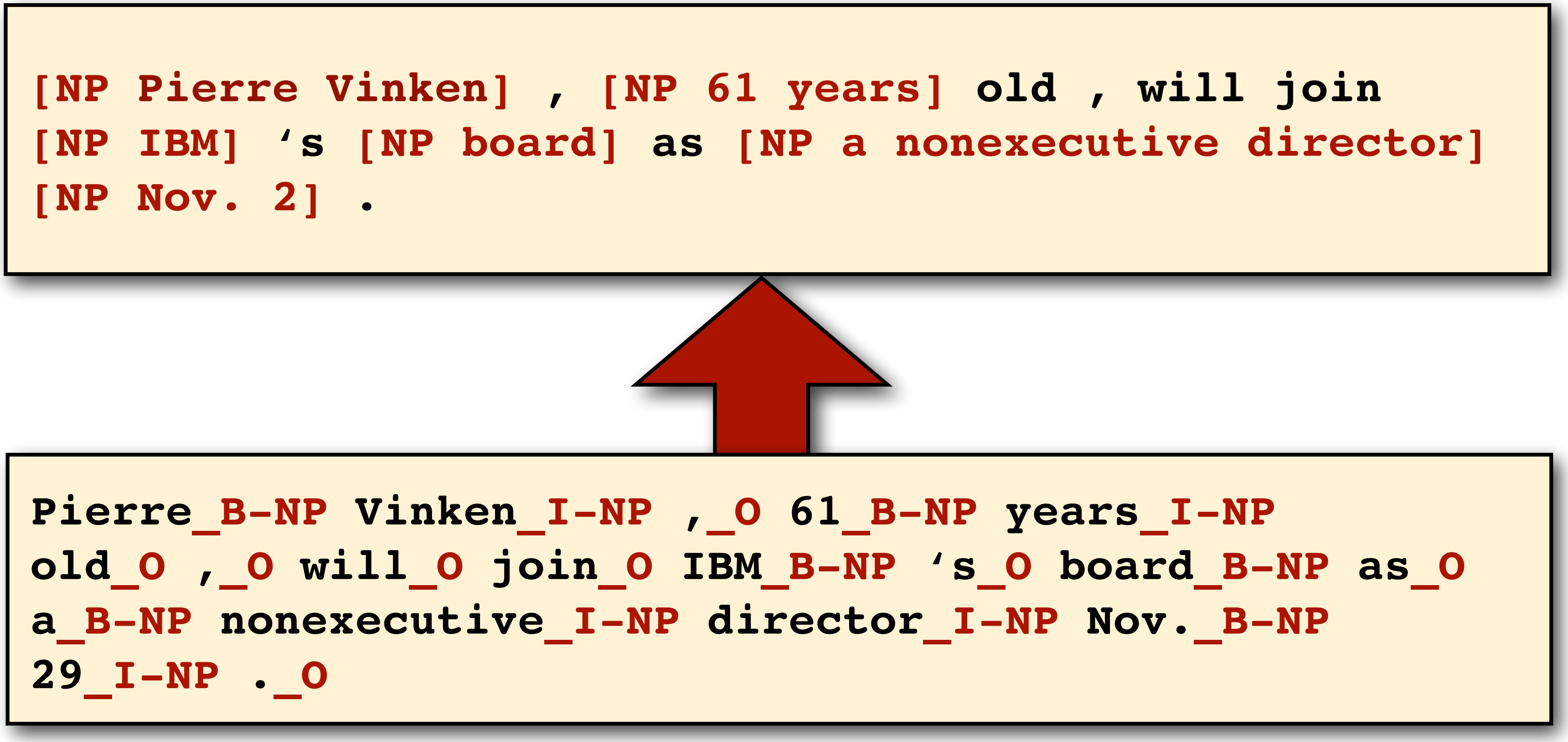
[NP Pierre Vinken] , [NP 61 years] old , will join
[NP IBM] ‘s [NP board] as [NP a nonexecutive director]
[NP Nov. 2] .

Task: identify all non-recursive NP chunks

The BIO encoding

We define three new tags:

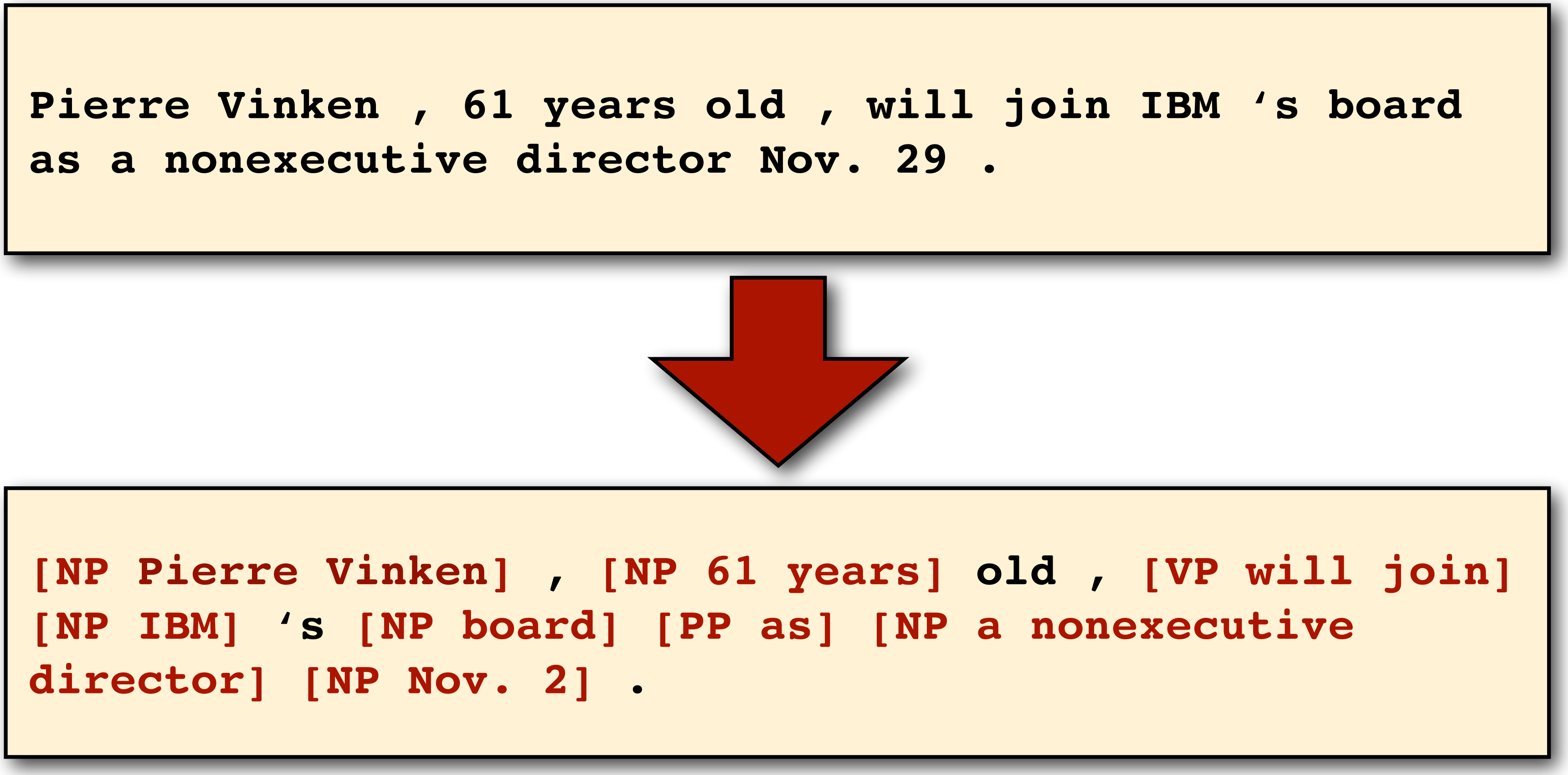
- **B-NP**: beginning of a noun phrase chunk
- **I-NP**: inside of a noun phrase chunk
- **O**: outside of a noun phrase chunk



```
[NP Pierre Vinken] , [NP 61 years] old , will join
[NP IBM] ‘s [NP board] as [NP a nonexecutive director]
[NP Nov. 2] .
```

```
Pierre_B-NP Vinken_I-NP ,_O 61_B-NP years_I-NP
old_O ,_O will_O join_O IBM_B-NP ‘s_O board_B-NP as_O
a_B-NP nonexecutive_I-NP director_I-NP Nov._B-NP
29_I-NP ._O
```

Shallow parsing



Pierre Vinken , 61 years old , will join IBM ‘s board as a nonexecutive director Nov. 29 .

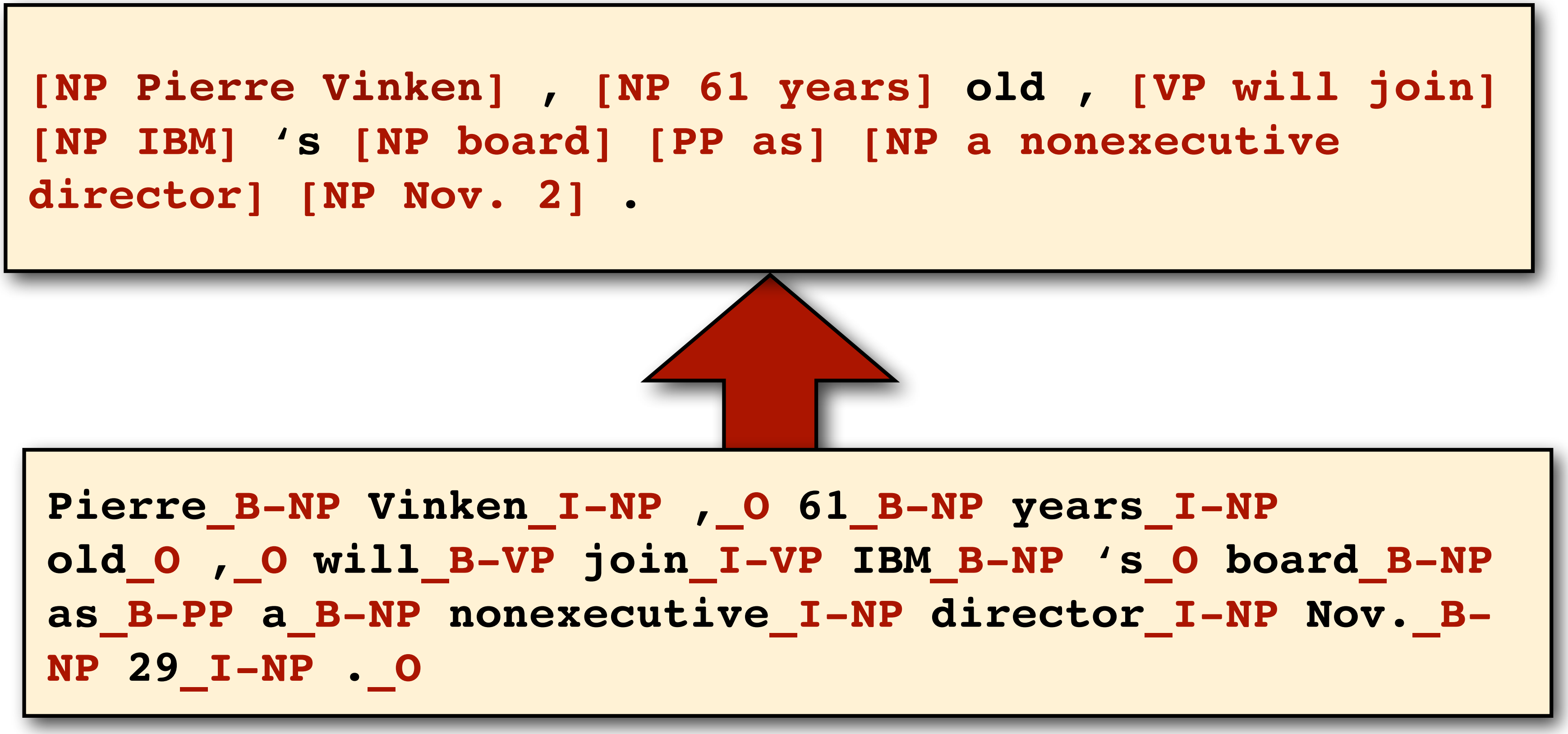
[NP Pierre Vinken] , [NP 61 years] old , [VP will join] [NP IBM] ‘s [NP board] [PP as] [NP a nonexecutive director] [NP Nov. 2] .

Task: identify all non-recursive NP,
verb (“VP”) and preposition (“PP”) chunks

The BIO encoding for shallow parsing

We define several new tags:

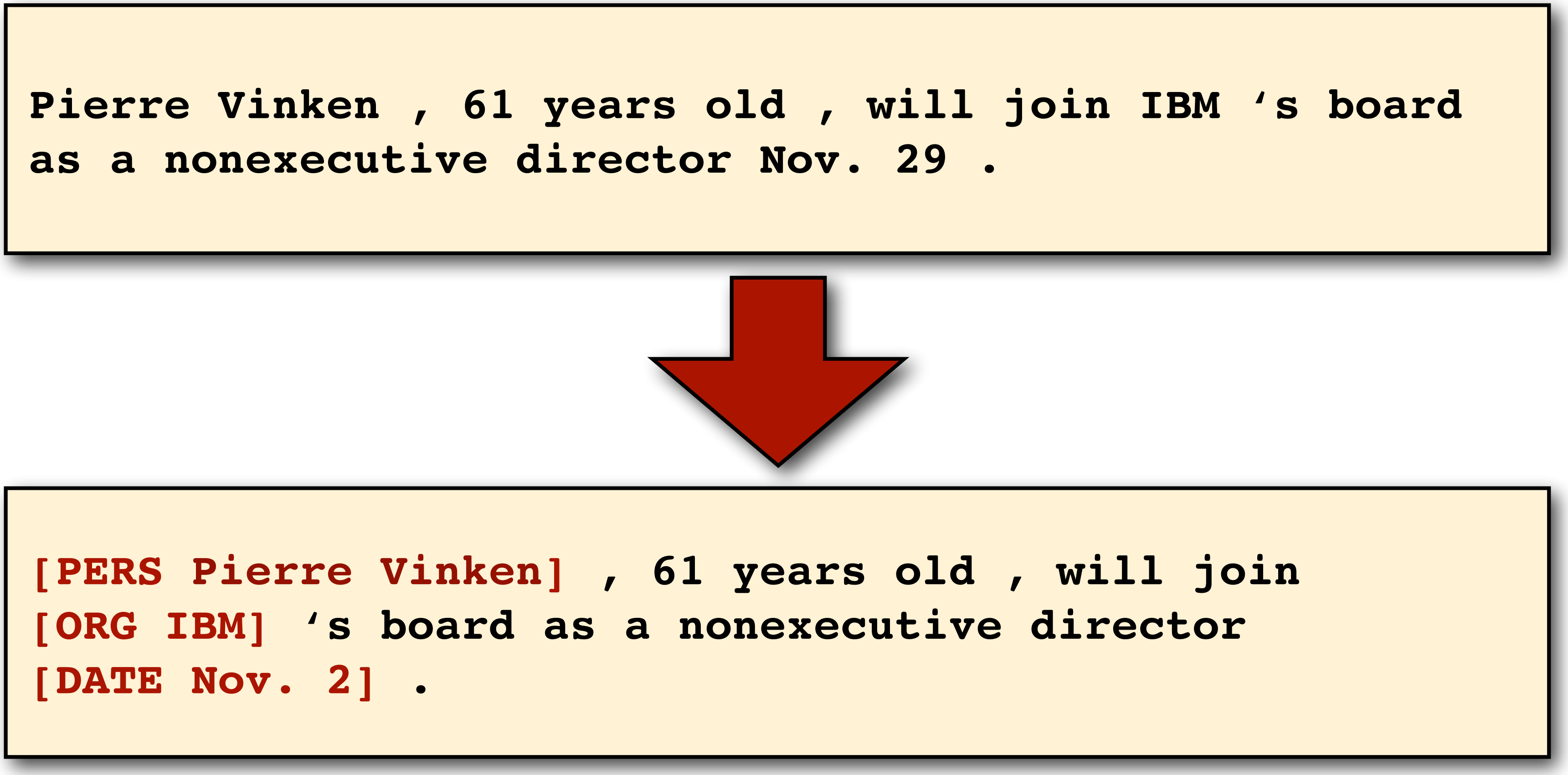
- **B-NP B-VP B-PP**: beginning of an NP, “VP”, “PP” chunk
- **I-NP I-VP I-PP**: inside of an NP, “VP”, “PP” chunk
- **O**: outside of any chunk



```
[NP Pierre Vinken] , [NP 61 years] old , [VP will join]
[NP IBM] ‘s [NP board] [PP as] [NP a nonexecutive
director] [NP Nov. 2] .
```

```
Pierre_B-NP Vinken_I-NP ,_O 61_B-NP years_I-NP
old_O ,_O will_B-VP join_I-VP IBM_B-NP ‘s_O board_B-NP
as_B-PP a_B-NP nonexecutive_I-NP director_I-NP Nov._B-
NP 29_I-NP ._O
```

Named Entity Recognition



```
Pierre Vinken , 61 years old , will join IBM ‘s board
as a nonexecutive director Nov. 29 .
```

```
[PERS Pierre Vinken] , 61 years old , will join
[ORG IBM] ‘s board as a nonexecutive director
[DATE Nov. 2] .
```

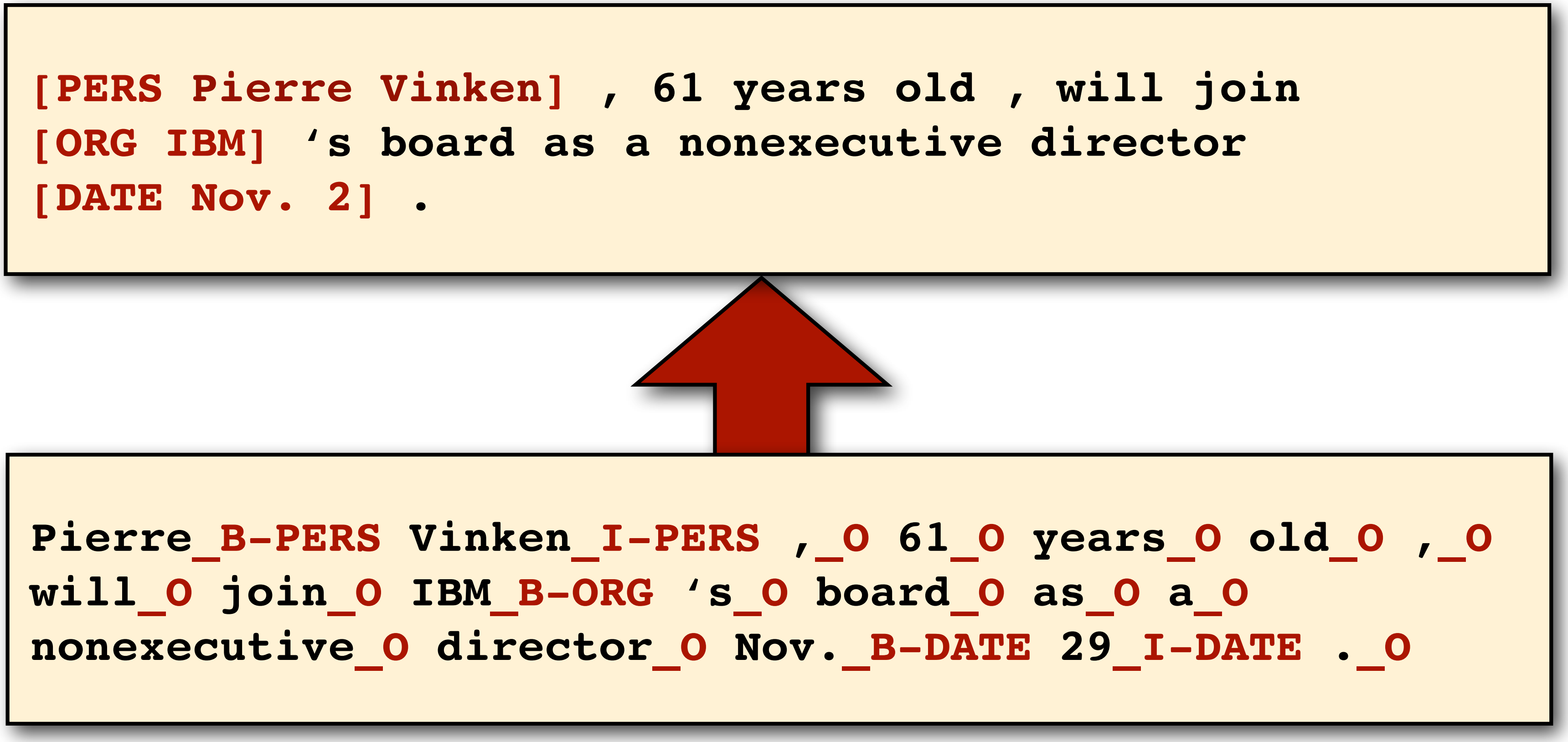
Task: identify all mentions of named entities (people, organizations, locations, dates)

The BIO encoding for NER

We define many new tags:

– **B-PERS**, **B-DATE, …:** beginning of a mention of a person/ date...

– **I-PERS**, **I-DATE, …:** inside of a mention of a person/date...



```
[PERS Pierre Vinken] , 61 years old , will join
[ORG IBM] ‘s board as a nonexecutive director
[DATE Nov. 2] .
```

```
Pierre_B-PERS Vinken_I-PERS ,_O 61_O years_O old_O ,_O
will_O join_O IBM_B-ORG ‘s_O board_O as_O a_O
nonexecutive_O director_O Nov._B-DATE 29_I-DATE ._O
```

Sequence Labeling

Input: a sequence of *n* tokens/words:

Pierre Vinken , 61 years old , will join IBM ‘s board as a nonexecutive director Nov. 29

Output: a sequence of *n* labels, such that each token/word is associated with a label:

POS-tagging: Pierre_**NNP** Vinken_**NNP** ,_**,** 61_**CD** years_**NNS** old_**JJ** ,_**,** will_**MD** join_**VB** IBM_**NNP** ‘s_**POS** board_**NN** as_**IN** a_**DT** nonexecutive_**JJ** director_**NN** Nov._**NNP** 29_**CD** ._**.**

Named Entity Recognition: Pierre_**B-PERS** Vinken_**I-PERS** ,_**O** 61_**O** years_**O** old_**O** ,_**O** will_**O** join_**O** IBM_**B-ORG** ‘s_**O** board_**O** as_**O** a_**O** nonexecutive_**O** director_**O** Nov._**B-DATE** 29_**I-DATE** ._**O**

BIO encodings in general

BIO encoding can be used to frame any task that requires the identification of non-overlapping and non-nested text spans as a sequence labeling problem, e.g.:

- NP chunking
- Shallow Parsing
- Named entity recognition

Sequence labeling algorithms

Statistical models:

- Maximum Entropy Markov Models (MEMMs)
- Conditional Random Fields (CRFs)

Neural models:

- Recurrent networks (or transformers) that predict a label at each time step, possibly with a CRF output layer.

Maximum Entropy Markov Models

MEMMs use a **logistic regression** ("Maximum Entropy") classifier for each $P(t^{(i)} | w^{(i)}, t^{(i-1)})$

$$P(t^{(i)} = t_k \mid t^{(i-1)}, w^{(i)}) = \frac{\exp(\sum_j \lambda_{jk} f_j(t^{(i-1)}, w^{(i)}))}{\sum_l \exp(\sum_j \lambda_{jl} f_j(t^{(i-1)}, w^{(i)}))}$$

Here, $t^{(i)}$: label of the i-th word vs. t_i = i-th label in the inventory

This requires the definition of a **feature function** $f(t^{(i-1)}, w^{(i)})$ that returns an *n*-dimensional feature vector for predicting label $t^{(i)}=t_j$ given inputs $t^{(i-1)}$ and $w^{(i)}$

Training returns weights λ_{jk} for each feature j used to predict label t_k

Conditional Random Fields (CRFs)

Conditional Random Fields have the same mathematical definition as MEMMs, but:

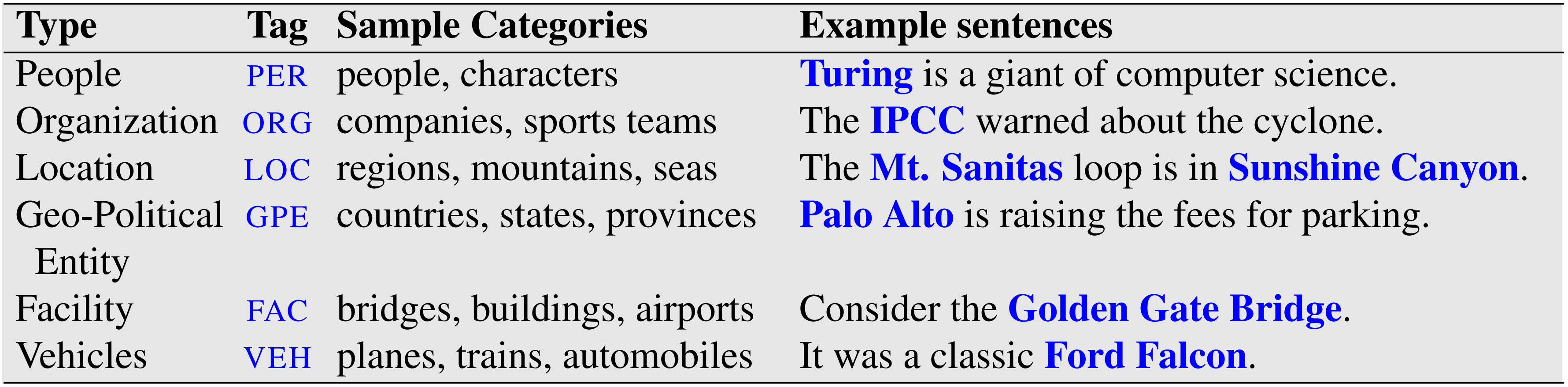
- CRFS are trained globally to maximize the probability of the overall sequence,
- MEMMs are trained locally to maximize the probability of each individual label

This requires dynamic programming

- Training: akin to the Forward-Backward algorithm used to train HMMs from unlabeled sequences)
- Decoding: Viterbi

Named Entity Recognition (NER)

Named Entity Types



Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	The **Mt. Sanitas** loop is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states, provinces	**Palo Alto** is raising the fees for parking.
Facility	FAC	bridges, buildings, airports	Consider the **Golden Gate Bridge**.
Vehicles	VEH	planes, trains, automobiles	It was a classic **Ford Falcon**.

Figure 18.1 A list of generic named entity types with the kinds of entities they refer to.

These types were developed for the news domain as part of NIST’s Automatic Content Extraction (ACE) program.

Other domains (e.g. biomedical text) require different types (proteins, genes, diseases, etc.)

Features for NER

Lists of common names exist for many entities

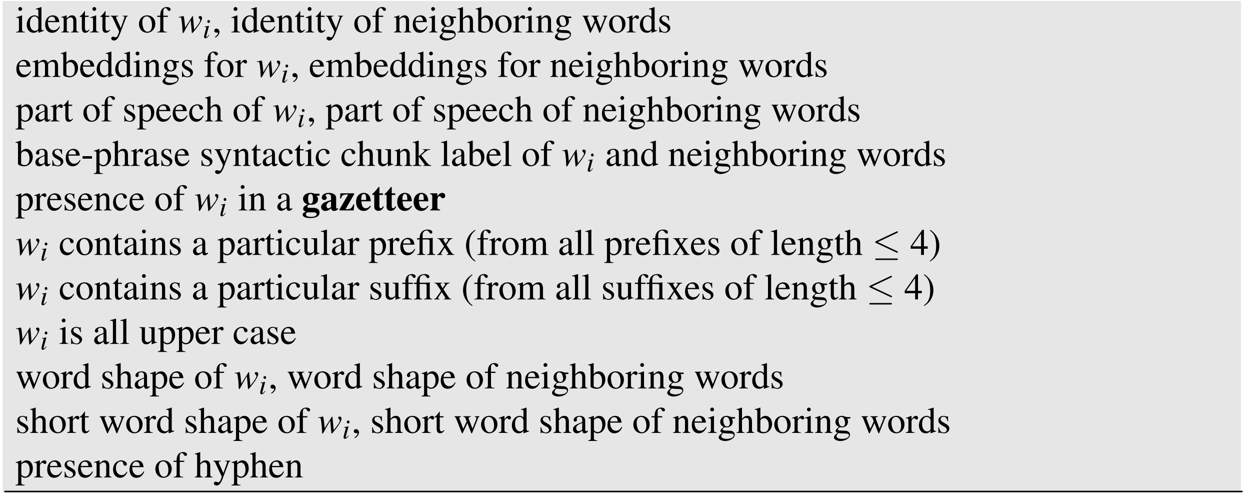
- — Gazetteers (place names, www.geonames.org),
- — Census-derived lists of first names and surnames,
- — Genes, proteins, diseases, etc.
- — Company names

Such lists can be helpful, but:

… Zipf's Law: these lists are typically not exhaustive, (and the distribution of names has a long tail)

… Ambiguity: many entity names either refer to different types of entities (*Washington*: person, places named after the person), or are used to refer to different types of entity (metonymy: *Washington* as reference to the US governement)

Feature-based NER



identity of w_i, identity of neighboring words
embeddings for w_i, embeddings for neighboring words
part of speech of w_i, part of speech of neighboring words
base-phrase syntactic chunk label of w_i and neighboring words
presence of w_i in a **gazetteer**
w_i contains a particular prefix (from all prefixes of length ≤ 4)
w_i contains a particular suffix (from all suffixes of length ≤ 4)
w_i is all upper case
word shape of w_i, word shape of neighboring words
short word shape of w_i, short word shape of neighboring words
presence of hyphen

Figure 18.5 Typical features for a feature-based NER system.

Train a sequence labeling model (MEMM or CRF), using features such as the ones listed above for English

— Word Shape: replace all upper-case letters with one symbol (e.g. “X”), all lower-case letters with another symbol (“x”), all digits with another symbol (“d”), and leave punctuation marks as is (“L’Occitane → “X’Xxxxxxxx”)

— Short Word Shape: remove adjacent letters that are identical in word shape “L’Occitane → “X’Xxxxxxxx” → “X’Xx”)

Neural NER

Sequence RNN (e.g. biLSTM or Transformer) with a CRF output layer.

Input: word embeddings, possibly concatenated with character embeddings and other features, e.g.:

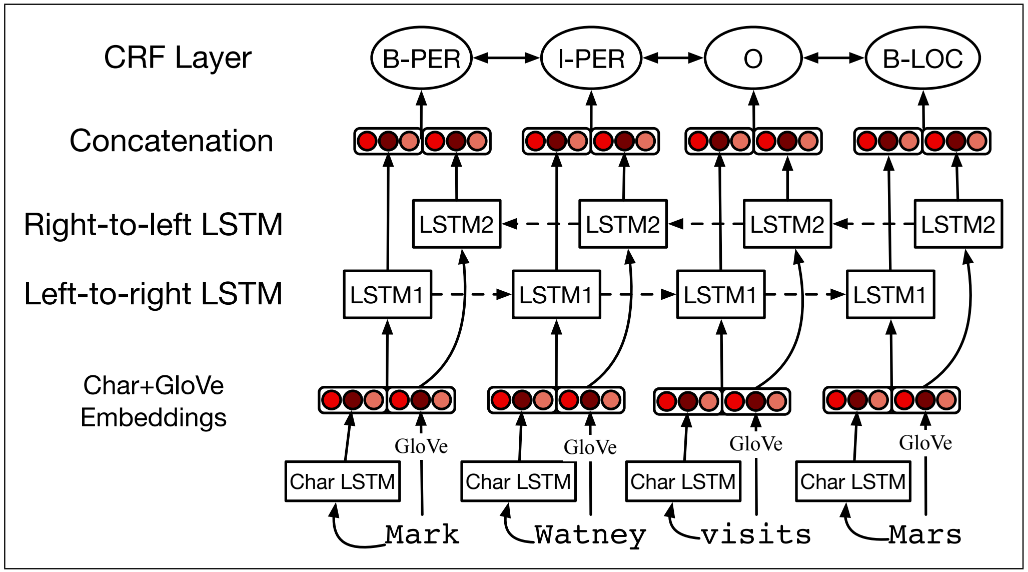


Figure 18.8 Putting it all together: character embeddings and words together in a bi-LSTM sequence model. After Lample et al. (2016).

Rule-based NER

The textbook gives an example of an iterative approach that makes multiple passes over the text:

- Pass 1: Use high-precision rules
  to label (a small number of) unambiguous mentions
- Pass 2: Propagate the labels of the previously detected
  named entities to any mentions
  that are substrings (or acronyms?) of these entities
- Pass 3: Use application-specific name lists
  to identify further likely names (as features?)
- Pass 4: Now use a sequence labeling approach for NER,
  keeping the already labeled entities
  as high-precision anchors.

The basic ideas behind this approach (label propagation, using high-precision items as anchors) can be useful for other tasks as well.

Relations and Relation Extraction

WordNet as a database of relations between *concepts*

Hyponym relations (is-a relation)

cats are mammals

Meryonym relations (part-of/has-a relations):

Part meronyms: bumpers are parts of cars,
cars have bumpers

Member meronyms: musicians belong to bands/orchestras,

Substance meronyms: dough contains flour

NB: some of these are inherited via hypernyms:
'musician' is a member meronym of 'musical organization',
which has hyponyms such as 'orchestra', 'band', 'choir', etc.

Domain knowledge expressed as relations

Wikipedia's **infoboxes** provide structured facts about **named entities**:

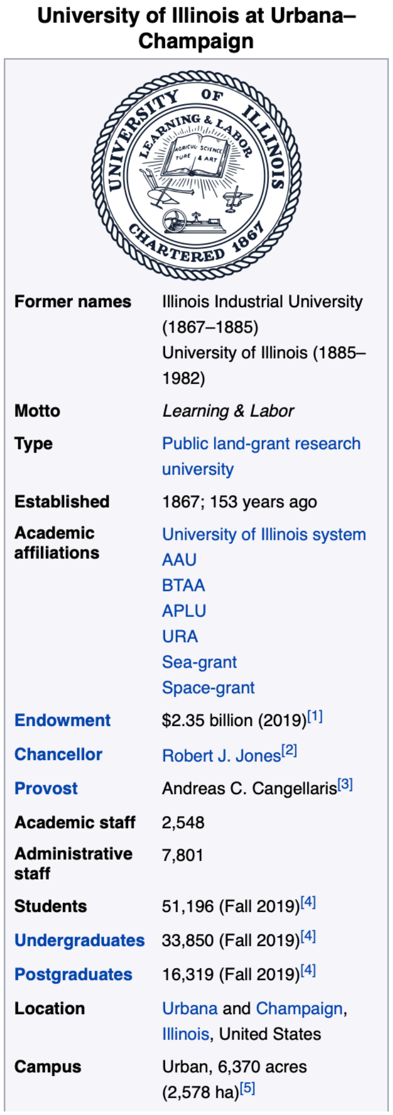
These can be turned into **structured relations** between these entities, e.g.

location-of(UIUC, Illinois)

or **RDF** (Resource Description Framework) **triples**

```
(entity, relation, entity):
(UIUC, location, Illinois)
```

Freebase and **DBPedia** (2 billion RDF triples) are both very large knowledge bases of such relations, extracted from Wikipedia.

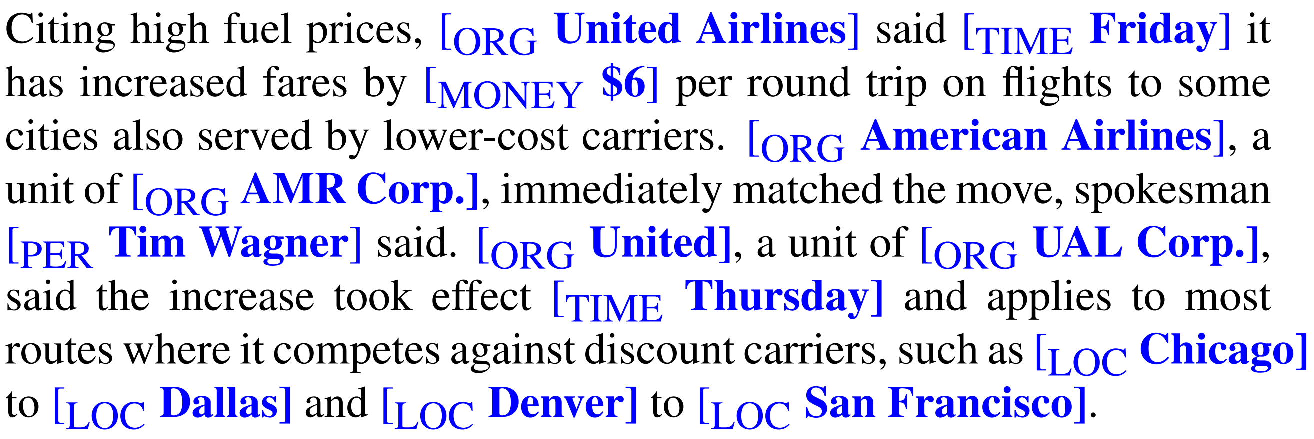


University of Illinois at Urbana–Champaign

Former names	Illinois Industrial University (1867–1885) University of Illinois (1885–1982)
Motto	*Learning & Labor*
Type	Public land-grant research university
Established	1867; 153 years ago
Academic affiliations	University of Illinois system AAU BTAA APLU URA Sea-grant Space-grant
Endowment	$2.35 billion (2019)[1]
Chancellor	Robert J. Jones[2]
Provost	Andreas C. Cangellaris[3]
Academic staff	2,548
Administrative staff	7,801
Students	51,196 (Fall 2019)[4]
Undergraduates	33,850 (Fall 2019)[4]
Postgraduates	16,319 (Fall 2019)[4]
Location	Urbana and Champaign, Illinois, United States
Campus	Urban, 6,370 acres (2,578 ha)[5]

https://en.wikipedia.org/wiki/University_of_Illinois_at_Urbana–Champaign

Relation Extraction from text



Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

Can we identify that…

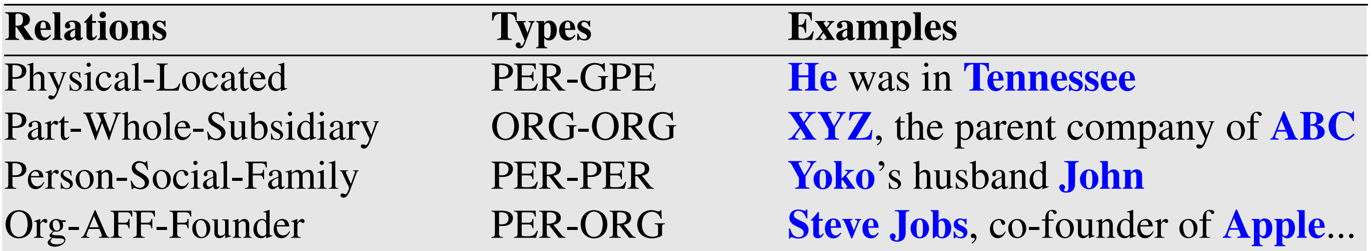
…American Airlines is part of (a unit of) AMR,

…United Airlines is part of (a unit of) UAL Corp,

…Tim Wagner is employed by (a spokesman of) AMR

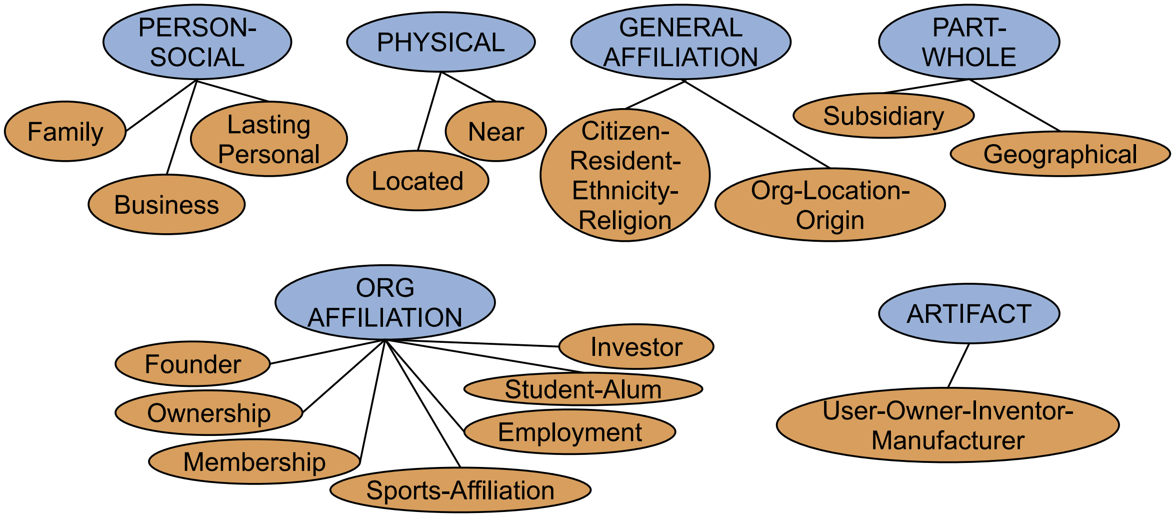
Relation Extraction from text

Identify **relations between named entities**, typically from a small set of predefined relations.



Relations	Types	Examples
Physical-Located	PER-GPE	**He** was in **Tennessee**
Part-Whole-Subsidiary	ORG-ORG	**XYZ**, the parent company of **ABC**
Person-Social-Family	PER-PER	**Yoko**'s husband **John**
Org-AFF-Founder	PER-ORG	**Steve Jobs**, co-founder of **Apple**...

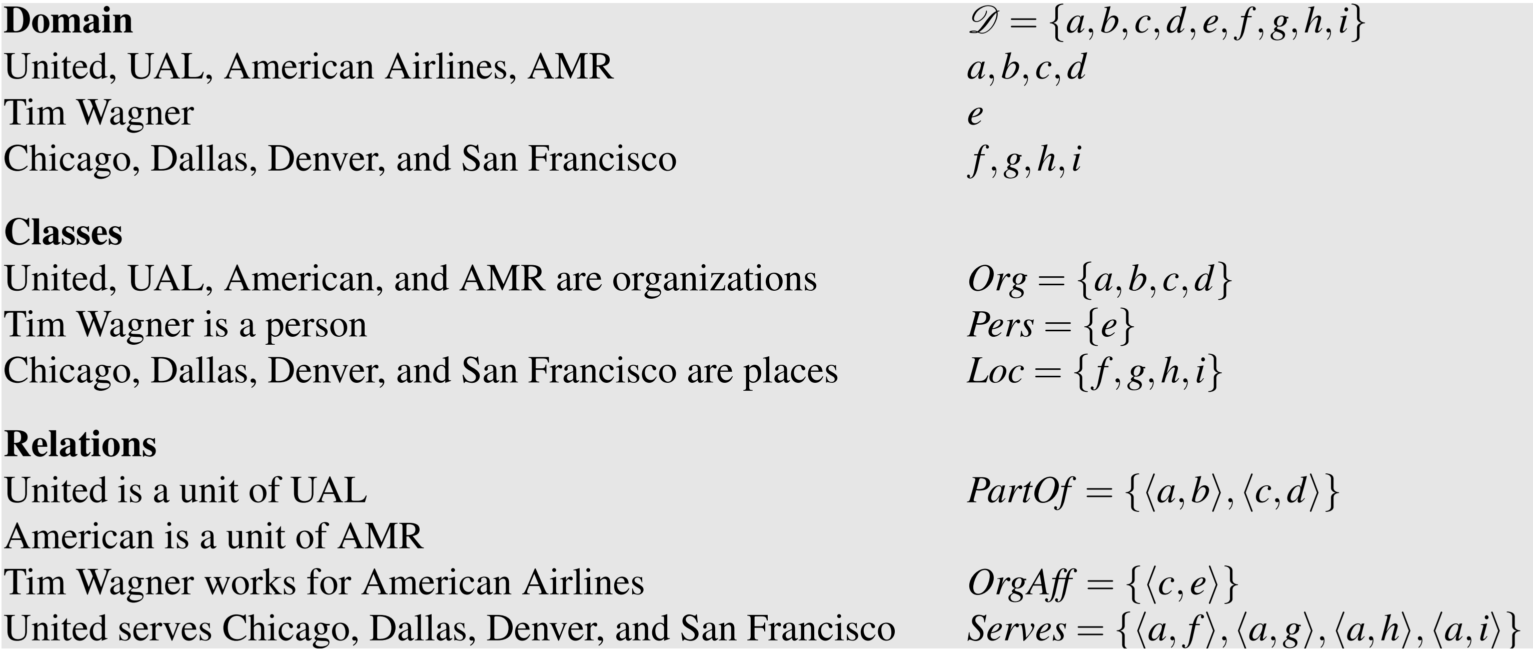
The 17 relations (orange) used in ACE:



A logical interpretation

We can construct a model for these relations:

- The **domain** (universe) is a **set of named entities,** partitioned into different types or classes of entities
- Each **relation** is a **set of tuples** of entities (restricted to relation-specific tuples of types)



Domain	$\mathscr{D} = \{a,b,c,d,e,f,g,h,i\}$
United, UAL, American Airlines, AMR	a,b,c,d
Tim Wagner	e
Chicago, Dallas, Denver, and San Francisco	f,g,h,i
Classes	
United, UAL, American, and AMR are organizations	$Org = \{a,b,c,d\}$
Tim Wagner is a person	$Pers = \{e\}$
Chicago, Dallas, Denver, and San Francisco are places	$Loc = \{f,g,h,i\}$
Relations	
United is a unit of UAL	$PartOf = \{\langle a,b\rangle, \langle c,d\rangle\}$
American is a unit of AMR	
Tim Wagner works for American Airlines	$OrgAff = \{\langle c,e\rangle\}$
United serves Chicago, Dallas, Denver, and San Francisco	$Serves = \{\langle a,f\rangle, \langle a,g\rangle, \langle a,h\rangle, \langle a,i\rangle\}$

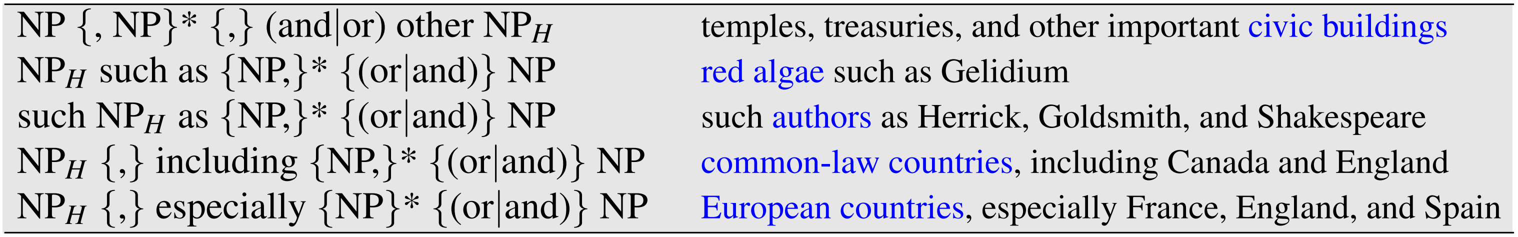
Rule-based relation extraction

Handwritten rules to identify **lexico-syntactic patterns** (Hearst, 1992) can be used for high-precision (and low-recall) relation extraction:

> Agar is a substance prepared from a mixture of **red algae, such as Gelidium**, for laboratory or industrial use

The **pattern** "X, such as Y (and/or Z)"

implies that X is a hypernym of Y and Z.



$\text{NP}\ \{,\ \text{NP}\}^*\ \{,\}\ (\text{and} \vert \text{or})\ \text{other}\ \text{NP}_H$	temples, treasuries, and other important civic buildings
$\text{NP}_H\ \text{such as}\ \{\text{NP},\}^*\ \{(\text{or} \vert \text{and})\}\ \text{NP}$	red algae such as Gelidium
$\text{such}\ \text{NP}_H\ \text{as}\ \{\text{NP},\}^*\ \{(\text{or} \vert \text{and})\}\ \text{NP}$	such authors as Herrick, Goldsmith, and Shakespeare
$\text{NP}_H\ \{,\}\ \text{including}\ \{\text{NP},\}^*\ \{(\text{or} \vert \text{and})\}\ \text{NP}$	common-law countries, including Canada and England
$\text{NP}_H\ \{,\}\ \text{especially}\ \{\text{NP}\}^*\ \{(\text{or} \vert \text{and})\}\ \text{NP}$	European countries, especially France, England, and Spain

Figure 18.12 Hand-built lexico-syntactic patterns for finding hypernyms, using {} to mark optionality (Hearst 1992a, Hearst 1998).

Relation Extraction via supervised learning

Learn a classifier that identifies whether there is a relation between a pair of entities that appear in the same sentence (or nearby within a document).

Classifier output: *n*+1 classes for *n* rels (incl. NONE)

Useful features:

— the words appearing in and next to the entities

— the words between the entities

— the NER types of both entities

— the distance between both entities (#words, #NERs,…)

— the syntactic path between the entities

Semi-supervised Relation Extraction

Use **high-precision seed patterns** (e.g. “X’s Y”) relations to identify **high-confidence seed tuples**.

Ryanair’s hub Charleroi `-> (Ryanair, has-hub-in, Charleroi)`

Bootstrap a classifier with increasing coverage:

— Find sentences containing entity pairs from seeds.

“Ryanair, which uses Charleroi as hub”
“Ryanair’s Belgian hub at Charleroi”

— These will contain new patterns
(as well as some noise: “Sydney has a ferry hub at Circular Quay”)

— Noise needs to be controlled so as not to propagate
(Confidence values, combined across patterns via noisy-or)

Distant Supervision for Relation Extraction

— Use **a very large database of known relations** (Freebase, DBPedia) to obtain a very large number of seed tuples.

```
(John F. Kennedy, died-in, Dallas)
(Princess Diana, died-in, Paris)
(Elvis Presley, died-in, Memphis)
```

— Search **large amounts of text** for sentences containing pairs of entities in a known relation

(plus entities in this list not in any known relation, to get no-relation examples)

— Process these sentences with NER, syntactic parsing, etc.

— **Learn a classifier** on these sentences to predict relations between entities that are not in the database

What is the intuition why this might work?

This returns a lot of noise: Elvis performed/lived/is buried in/sang about/… Memphis
But if trained on enough data, high-confidence predictions of this classifier are likely to be correct (since many true positive examples will be similar to each other)

Unsupervised Relation Extraction (“Open Information Extraction/IE”)

Goal: Extract any relation (from large amounts of text, e.g. web)
without being restricted to a predefined set of relations

Relations: Raw strings of words (often beginning with verbs, and possibly subject to some predefined syntactic constraints)

Example: The ReVerb algorithm:

— Run a POS tagger and entity chunker over each sentence

— Identify any potential relations (any string between entities that starts with a verb and obeys predefined constraints)

— Normalize relations (remove inflection, auxiliary verbs, adjectives, adverbs)

— Add relations that occur with at least N different arguments to database

— Train a classifier on small number (1000) hand-labeled sentences to obtain confidence scores for relations in the database.