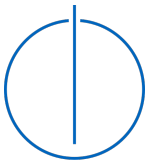


Faculty for Informatics

Technical
University
of Munich

Natural Language Processing
IN2361

PD Dr. Georg Groh

Social Computing
Research Group

Chapter 18
Information Extraction

- content is based on [1]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] or from [1] are omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!

Named Entity Recognition

- Named Entity: Anything referred to by a proper name, often extended to temporal or numerical expressions

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

- 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 **Tappan Zee Bridge**.
Vehicles	VEH	planes, trains, automobiles	It was a classic **Ford Falcon**.
...			

Named Entity Recognition

- Named entity recognition: finding spans of text that constitute NEs + classification

- categorial ambiguities:

Name	Possible Categories
Washington	Person, Location, Political Entity, Organization, Vehicle
Downing St.	Location, Organization
IRA	Person, Organization, Monetary Instrument
Louis Vuitton	Person, Organization, Commercial Product

[$_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, [$_{\text{GPE}}$ Washington] passed a primary seatbelt law.
The [$_{\text{VEH}}$ Washington] had proved to be a leaky ship, every passage I made...

NER as Sequence Labeling Task

- use supervised sequence classifier such as MEMM or RNN, with IOB tagging (for n entity types → 2n+1 corresp. IOB classes) or IO tagging (n+1 corresp. IO classes):

[ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim Wagner**] said.

Words	IOB Label	IO Label
American	B-ORG	I-ORG
Airlines	I-ORG	I-ORG
,	O	O
a	O	O
unit	O	O
of	O	O
AMR	B-ORG	I-ORG
Corp.	I-ORG	I-ORG
,	O	O
immediately	O	O
matched	O	O
the	O	O
move	O	O
,	O	O
spokesman	O	O
Tim	B-PER	I-PER
Wagner	I-PER	I-PER
said	O	O
.	O	O

NER as Sequence Labeling Task

- features used (similar to supervised POS tagging, esp. for unknown words (= often NEs)):

identity of w_i
identity of 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

example: *L'Occitane*
prefix(w_i) = L
prefix(w_i) = L'
prefix(w_i) = L'O
prefix(w_i) = L'Oc
suffix(w_i) = tane
suffix(w_i) = ane
suffix(w_i) = ne
suffix(w_i) = e
word-shape(w_i) = X'Xxxxxxxx
short-word-shape(w_i) = X'Xx

NER as Sequence Labeling Task

- very important for English NER: word shape features: map lower-case letters to ‘x’, upper-case to ‘X’, numbers to ’d’, and retaining punctuation.
  examples: *I.M.F* → X.X.X. or *DC10-30* → XXdd-dd or *Hannah* → Xxxxxx

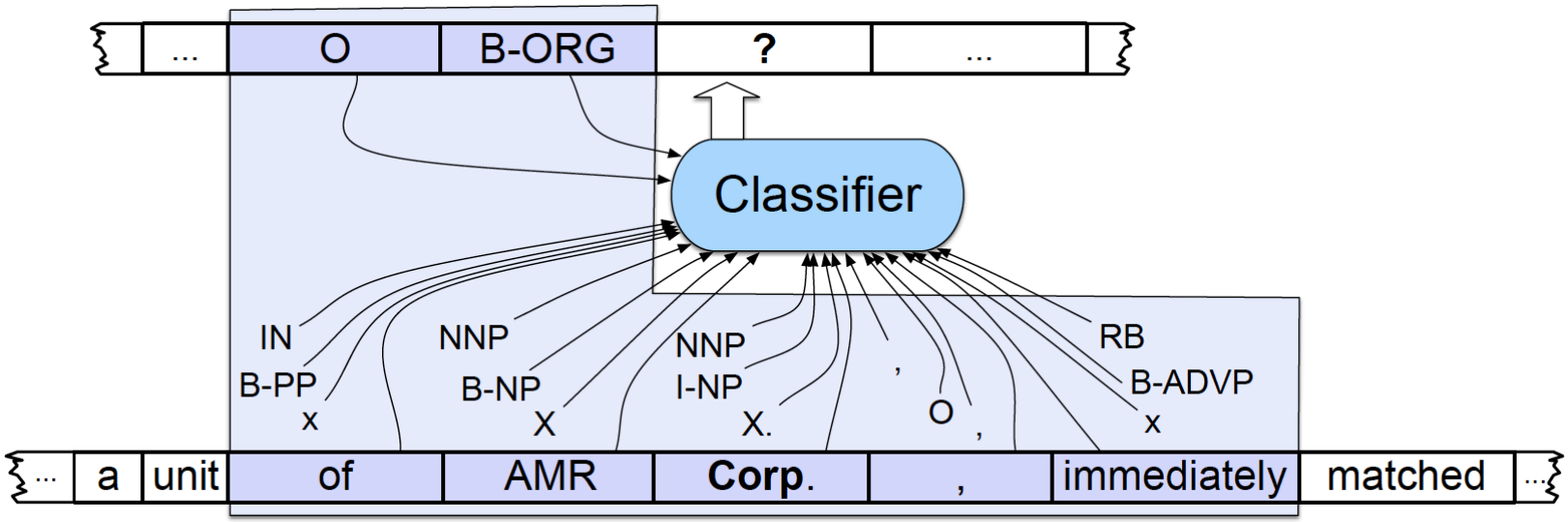
- shorter word shape features: consecutive character types removed
  examples: *I.M.F* → X.X.X. or *DC10-30* → Xd-d

- Gazetteer: list of geographical place names (cities, regions, etc.) (useful for NER) and other lists (companies, persons, first names, last names etc.)
  (often less useful: corresponding categories are more open)

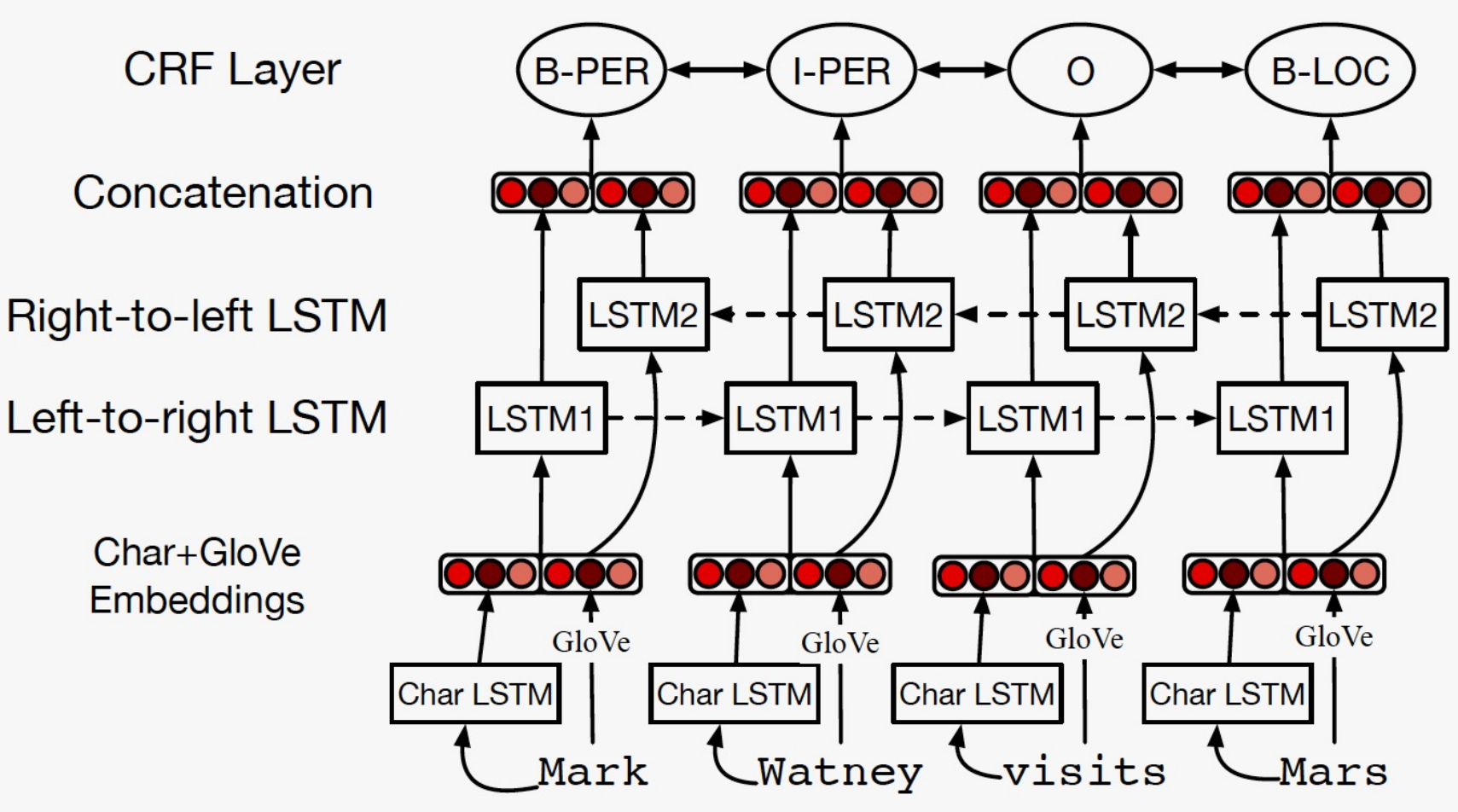
- usefulness of features: strongly dependent on language and text category
  (e.g. social media texts often neglect capitalization)

NER as Sequence Labeling Task

Word	POS	Chunk	Short shape	Label
American	NNP	B-NP	Xx	B-ORG
Airlines	NNPS	I-NP	Xx	I-ORG
,	,	O	,	O
a	DT	B-NP	x	O
unit	NN	I-NP	x	O
of	IN	B-PP	x	O
AMR	NNP	B-NP	X	B-ORG
Corp.	NNP	I-NP	Xx.	I-ORG
,	,	O	,	O
immediately	RB	B-ADVP	x	O
matched	VBD	B-VP	x	O
the	DT	B-NP	x	O
move	NN	I-NP	x	O
,	,	O	,	O



NER as Sequence Labeling Task – NN Methods



(more on the details in the second half of the lecture)

NER as Sequence Labeling Task – Rule-Based & Bootstrapping

- Rule-Based Systems: example **IBM System T:** text "understanding" architecture: user specifies complex declarative (rule-like) constraints for tagging
  - using formal query language that includes regular expressions, dictionaries, semantic constraints, NLP operators, and table structures, all of which the system compiles into an efficient extractor

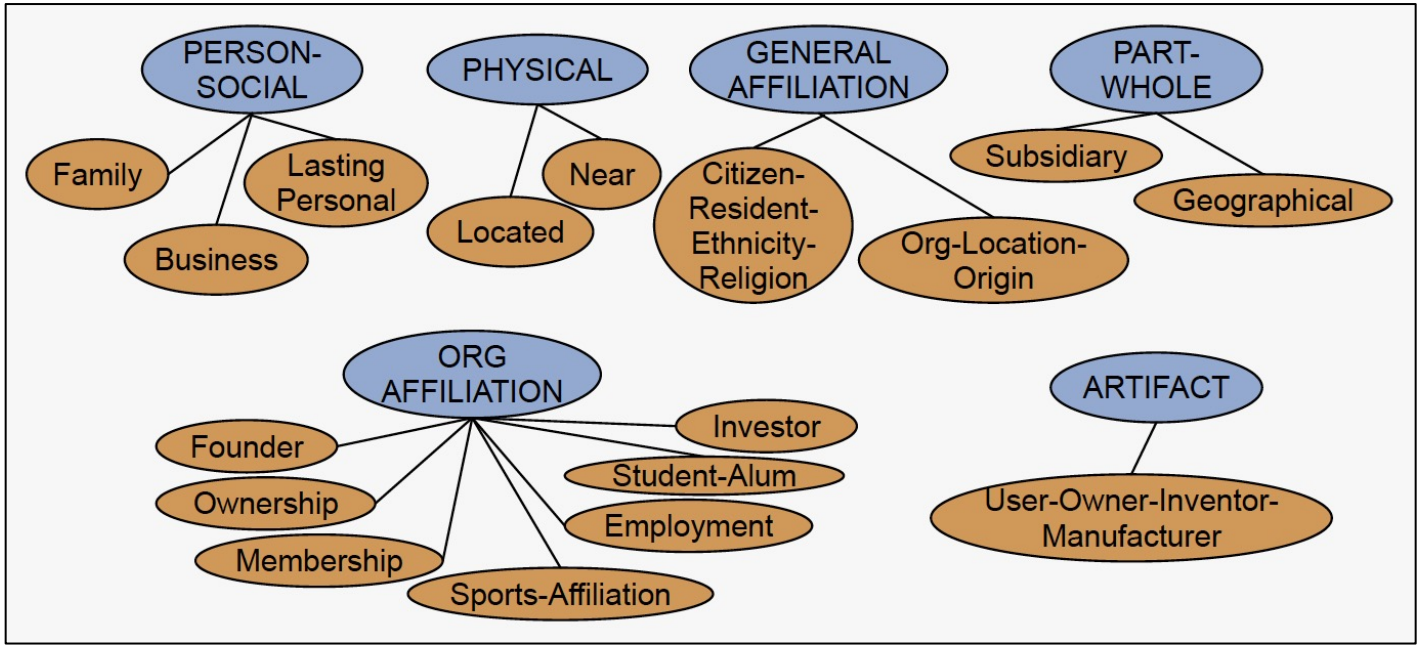
- Bootstrapping: multi-pass architectures:
  1. use high-precision (but low recall) rules to tag unambiguous entity mentions.
  2. search for substring matches of the previously detected names.
  3. consult application-specific name lists to identify likely named entity mentions from the given domain.
  4. apply probabilistic sequence labeling making use of tags from previous stages as additional features.

NER as Sequence Labeling Task – Evaluation

- Precision, Recall, F-Measure:
  evaluating performance of NER task: entity rather than the word is the unit of response → label *American* but not *American Airlines* as ORG → two errors: false positive for O and false negative for I-ORG

Relation Extraction

- 17 relations from ACE relation extraction task



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**...

Relation Extraction

model-based view of example text in terms of unary (←→ NER) and binary relations

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

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 American is a unit of AMR	$PartOf = \{\langle a,b\rangle, \langle c,d\rangle\}$
Tim Wagner works for American Airlines	$OrgAff = \{\langle c,e\rangle\}$
United serves Chicago, Dallas, Denver, and San Francisco	$Serves = \{\langle a,f\rangle, \langle a,g\rangle, \langle a,h\rangle, \langle a,i\rangle\}$

Relational / Ontological Databases

- the Semantic Web (with ontology standards OWL, RDF(S), SPARQL etc.)
  - RDF: subject-predicate-object triples:

subject	predicate	object
Golden Gate Park	location	San Francisco

  - DBPedia: (extensional) ontology derived from Wikipedia with > 3 billion RDF triples

- Wikipedia: semi-structured source for relations:
  example info-boxes: article about *Stanford University* →
  state = "California" , president = "Mark Tessier-Lavigne".

- Freebase: relations like:

people/person/nationality
location/location/contains
people/person/place-of-birth
biology/organism_classification

Relational / Ontological Databases

- WordNet:
  - hypernym hypernym relation:
    Giraffe is-a ruminant is-a ungulate is-a mammal is-a vertebrate is-a animal …
  - instance-of relation:
    San Francisco instance-of city .

- domain specific ontologies: example: Unified Medical Language System (UMLS): 134 broad subject categories, entity types, and 54 relations between the entities

Entity	Relation	Entity
Injury	disrupts	Physiological Function
Bodily Location	location-of	Biologic Function
Anatomical Structure	part-of	Organism
Pharmacologic Substance	causes	Pathological Function
Pharmacologic Substance	treats	Pathologic Function

Doppler echocardiography can be used to diagnose left anterior descending artery stenosis in patients with type 2 diabetes

→ UMLS relation: *Echocardiography, Doppler* Diagnoses Acquired stenosis

Relation Extraction Using Patterns

- *Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use.*
  → without detailed knowledge we can infer: hyponym(Gelidium, red-algae)

- → lexico-syntactic patterns e.g.

  if: $NP_0\ such\ as\ NP_1\{,NP_2\ldots,(and|or)NP_i\}, i \geq 1$

  then: $\forall NP_i, i \geq 1, \text{hyponym}(NP_i, NP_0)$

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

- patterns using NE types:

PER, POSITION of ORG:
George Marshall, Secretary of State of the United States

PER (named|appointed|chose|etc.) PER Prep? POSITION
Truman appointed Marshall Secretary of State

PER [be]? (named|appointed|etc.) Prep? ORG POSITION
George Marshall was named US Secretary of State

Relation Extraction Using Patterns

- *Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use.*
  → without detailed knowledge we can infer: hyponym(Gelidium, red-algae)

- → lexico-syntactic patterns e.g.

  if: $NP_0\ such\ as\ NP_1\{,NP_2\ldots,(and|or)NP_i\}, i \geq 1$

  then: $\forall NP_i, i \geq 1, \text{hyponym}(NP_i, NP_0)$

NP {, NP}* {,} (and|or) other NP_H	temples, treasuries, and other important civic bu
NP_H such as {NP,}* {(or|and)} NP	red algae such as Gelidium
such NP_H as {NP,}* {(or|and)} NP	such authors as Herrick, Goldsmith, and Shakesp
NP_H {,} including {NP,}* {(or|and)} NP	common-law countries, including Canada and E
NP_H {,} especially {NP}* {(or|and)} NP	European countries, especially France, England, and Spain

high precision, low recall

- patterns using NE types:

PER, POSITION of ORG:
George Marshall, Secretary of State of the United States

PER (named|appointed|chose|etc.) PER Prep? POSITION
Truman appointed Marshall Secretary of State

PER [be]? (named|appointed|etc.) Prep? ORG POSITION
George Marshall was named US Secretary of State

Relation Extraction Using Supervised ML

- 3 sub-classifiers necessary:

  (unfortunately: hand labelling is very costly here, and resulting classifiers are brittle (do not generalize well across domains)

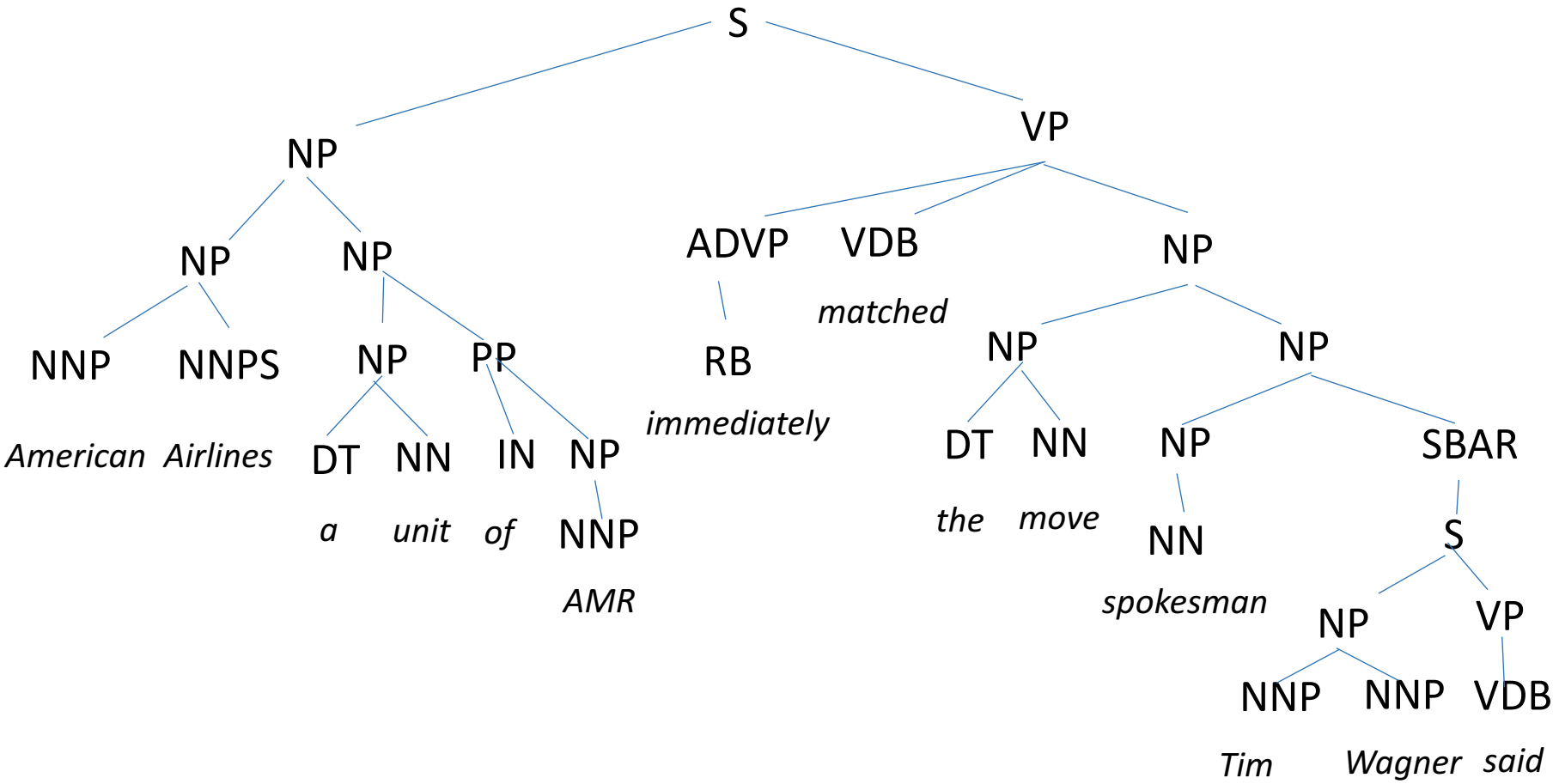
```
function FINDRELATIONS(words) returns relations

    relations ← nil
    entities ← FINDENTITIES(words)
    forall entity pairs ⟨e1, e2⟩ in entities do
        if RELATED?(e1, e2)
            relations ← relations+CLASSIFYRELATION(e1, e2)
```

- possible features:

  American Airlines [mention M1]*, a unit of AMR, immediately matched the move, spokesman Tim Wagner* [mention M2] *said*

M1 headword	*airlines*
M2 headword	*Wagner*
Word(s) before M1	NONE
Word(s) after M2	*said*
Bag of words between	{*a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman* }
M1 type	ORG
M2 type	PERS
Concatenated types	ORG-PERS
Constituent path	$NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
Base phrase path (= (chunk sequence))	$NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP$
Typed-dependency path	$Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$

Relation Extraction Using Supervised ML



Concatenated types		ORG-PERS
Constituent path		$NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
Base phrase path	= (chunk sequence)	$NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP$
Typed-dependency path		$Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$

Semi-Supervised Relation Extraction via Bootstrapping

- learning new rules / patterns from seed rules or seed tuples

function BOOTSTRAP(*Relation R*) **returns** *new relation tuples*

tuples ← Gather a set of seed tuples that have relation *R*

newpairs ← *tuples*

iterate

sentences ← find sentences that contain entities in *newpairs*

patterns ← generalize the context between and around entities in *sentences*

newpairs ← use *patterns* to grep for more tuples

newpairs ← *newpairs* with high confidence

tuples ← *tuples* + *newpairs*

return tuples

Semi-Supervised Relation Extraction via Bootstrapping

example:

- we know: Ryanair has a hub at Charleroi → seed: hub(Ryanair, Charleoi)
- find other sentences with *Ryanair, hub, Charleroi*:
  - Budget airline Ryanair, which uses Charleroi as a hub, scrapped all weekend flights out of the airport.
  - All flights in and out of Ryanair's Belgian hub at Charleroi airport were grounded on Friday...
  - A spokesman at Charleroi, a main hub for Ryanair, estimated that 8000 passengers had already been affected.
- use context of words between entity mentions, words before mention one, word after mention two, NE types of the two mentions, … → extract general patterns:

```
/ [ORG], which uses [LOC] as a hub /
/ [ORG]'s hub at [LOC] /
/ [LOC] a main hub for [ORG] /
```

Semi-Supervised Relation Extraction via Bootstrapping

Sydney has a ferry hub at Circular Quay → (Sydney, Circular Quay) ∈ hub?

→ assign confidence values to new tuples to avoid semantic drift:

- given :
  - document collection D,
  - current set of tuples T,
  - proposed pattern p

- define:
  - hits : set of tuples in T that p matches while looking in D
  - finds : total set of tuples that p finds in D

- then

$$Conf_{RlogF}(p) = \frac{hits_p}{finds_p} \times log(finds_p)$$

("reliability" * "frequency")

- using noisy or: combine evidence for new instance tuple t from all rules P' supporting it in D

$$Conf(t) = 1 - \prod_{p \in P'} (1 - Conf(p))$$

assumptions:
- for a proposed tuple to be false, all of its supporting patterns must have been in error
- the sources of their individual failures are independent.

Distant Supervision for Relation Extraction

hand-labelled training data is expensive → use relation databases to produce large number of seeds

function DISTANT SUPERVISION(*Database D, Text T*) **returns** *relation classifier C*

foreach relation *R*
  foreach tuple (*e1,e2*) of entities with relation *R* in *D*
    sentences ← Sentences in *T* that contain *e1* and *e2*
    f ← Frequent features in *sentences*
    observations ← observations + new training tuple (*e1, e2, f, R*)
C ← Train supervised classifier on *observations*
return *C*

also learn a "no –relation"-relation using tuples not in database

Distant Supervision for Relation Extraction

example: learn place-of-birth relationship between people and their birth cities

- DBPedia or Freebase: over 100,000 tuples of place-of-birth: <Edwin Hubble, Marshfield> , <Albert Einstein, Ulm>, …
- find all sentences in D with two NEs matching one of those tuples
  - …Hubble was born in Marshfield…
  - …Einstein, born (1879), Ulm…
  - …Hubble's birthplace in Marshfield…
- for each tuple (e.g. <born-in, Albert Einstein, Ulm>: from all sentences matching the tuple extract conjunctions of features (e.g. NE types of the two mentions, words and dependency paths in between the mentions, neighboring words etc.)
  - example sentence: American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said
    → conjunction of features:
    M1 = ORG & M2 = PER & nextword="said"& path= NP↑NP↑S↑S↓NP

Distant Supervision for Relation Extraction

- very large training sets → rich features that are conjunctions of individual features
- uses hand-created knowledge (although in a distant way) → high-precision evidence for the relation between entities
- able to use large number of features simultaneously → unlike iterative expansion of patterns in seed-based systems, there's no semantic drift.
- doesn't use a labeled training corpus of texts directly → no genre bias
- only useful if large extensional database for sought relations exists

Unsupervised Relation Extraction

(x, r, y)

example: ReVerb (2001):

- POS tag and chunk sentence s
- For each verb in s, find the longest sequence of words w that start with a verb and satisfy syntactic and lexical constraints, merging adjacent matches.
- For each phrase w , find the nearest NP x to the left which is not a relative pronoun, wh-word or existential “there”. Find the nearest NP y to the right.
- assign confidence c to the relation instance r = (x, w, y) using a confidence classifier and return it.

- example for a constraint specification (e.g. satisfied by *have a hub in*):

V | VP | VW*P
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

Unsupervised Relation Extraction

- confidence classifier:
  - run system on 1000 random web sentences, hand labelling each extracted relation as correct or incorrect → training data
  - train confidence classifier on this training data using features of the relation and the surrounding words such as

(x,r,y) covers all words in s
the last preposition in *r* is *for*
the last preposition in *r* is *on*
len(s) $\leq$ 10
there is a coordinating conjunction to the left of *r* in *s*
r matches a lone V in the syntactic constraints
there is preposition to the left of *x* in s.
there is an NP to the right of *y* in *s*.

- advantage of unsupervised relation extraction: ability to handle large number of relations without having to specify them in advance.

- example: *United has a hub in Chicago, which is the headquarters of United Continental Holdings*

→ r1: <United, has a hub in, Chicago>
r2: <Chicago, is the headquarters of, United Continental Holdings>

Evaluating Relation Extraction

- supervised RE: as usual

- semisupervised / unsupervised RE:
  - estimate precision by sampling the output and labelling the sample (ignore number of times a relation has been discovered):

$$\hat{P} = \frac{\text{\# of correctly extracted relation tuples in the sample}}{\text{total \# of extracted relation tuples in the sample.}}$$

  - estimate recall indirectly: compare estimated precision at different sample sizes (the top 1000 new relations, the top 10,000 new relations, the top 100,000, and so on)

Extracting Temporal Expressions

- Absolute points in time, durations, and relations between them

Absolute	Relative	Durations
April 24, 1916	yesterday	four hours
The summer of '77	next semester	three weeks
10:15 AM	two weeks from yesterday	six days
The 3rd quarter of 2006	last quarter	the last three quarters

- Temporal expressions: grammatical constructions with temporal triggers as heads.
  Lexical triggers: nouns, proper nouns, adjectives, or adverbs; temporal expressions: noun phrases, adjective phrases, and adverbial phrases

Category	Examples
Noun	*morning, noon, night, winter, dusk, dawn*
Proper Noun	*January, Monday, Ides, Easter, Rosh Hashana, Ramadan, Tet*
Adjective	*recent, past, annual, former*
Adverb	*hourly, daily, monthly, yearly*

Extracting Temporal Expressions

- TimeML:

  A fare increase initiated <TIMEX3>*last week*</TIMEX3> *by UALCorp's United Airlines was matched by competitors over* <TIMEX3>*the weekend*</TIMEX3>*, marking the second successful fare increase in* <TIMEX3>*two weeks*</TIMEX3> .

- Rule-based approaches: automata, regExs etc.

```
# yesterday/today/tomorrow
$string =~ s/(($OT+(early|earlier|later?)$CT+\s+)?(($OT+the$CT+\s+)?$OT+day$CT+\s+
$OT+(before|after)$CT+\s+)?$OT+$TERelDayExpr$CT+(\s+$OT+(morning|afternoon|
evening|night)$CT+)?)/<TIMEX2 TYPE=\"DATE\">$1<\/TIMEX2>/gio;

$string =~ s/($OT+\w+$CT+\s+)
<TIMEX2 TYPE=\"DATE\"[^>]*>($OT+(Today|Tonight)$CT+)<\/TIMEX2>/$1$2/gso;

# this/that (morning/afternoon/evening/night)
$string =~ s/(($OT+(early|earlier|later?)$CT+\s+)?$OT+(this|that|every|the$CT+\s+
$OT+(next|previous|following))$CT+\s*$OT+(morning|afternoon|evening|night)
$CT+(\s+$OT+thereafter$CT+)?)/<TIMEX2 TYPE=\"DATE\">$1<\/TIMEX2>/gosi;
```

Extracting Temporal Expressions

- supervised sequence labelling with any ML seq. classifier using IOB:

A fare increase initiated last week by UAL Corp's...
O O O O B I O O O

- features that may be used:

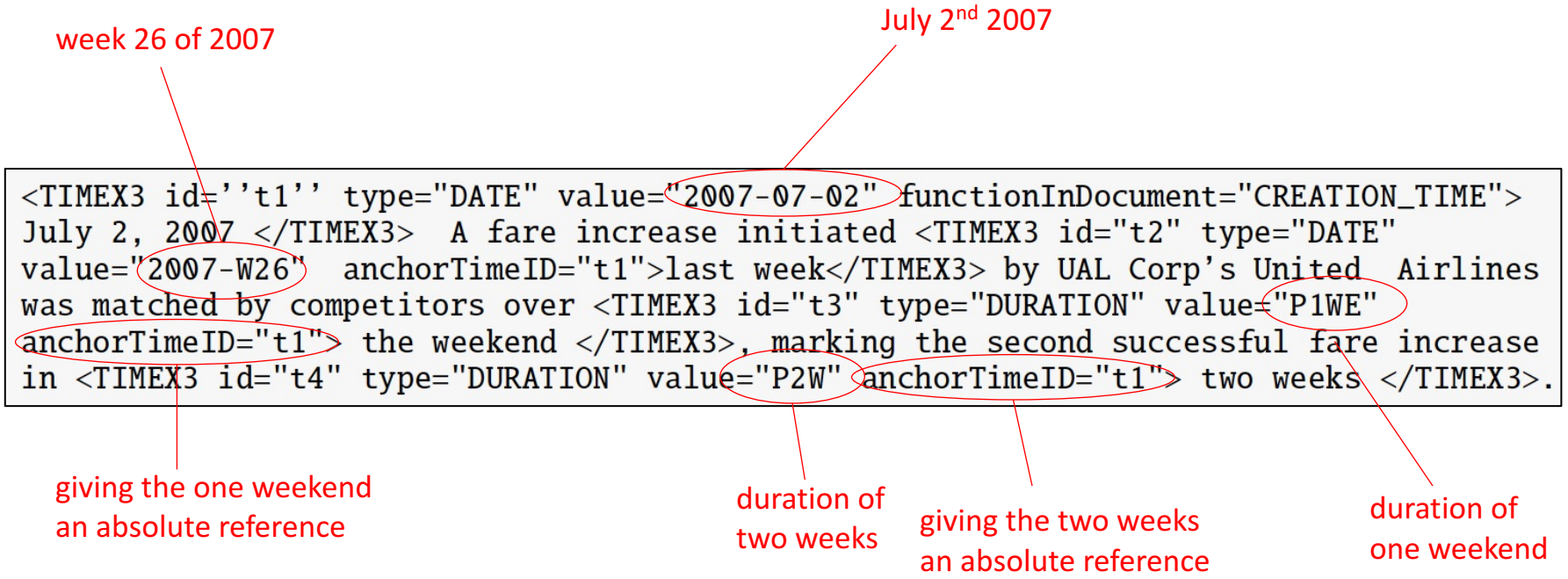
Feature	Explanation
Token	The target token to be labeled
Tokens in window	Bag of tokens in the window around a target
Shape	Character shape features
POS	Parts of speech of target and window words
Chunk tags	Base-phrase chunk tag for target and words in a window
Lexical triggers	Presence in a list of temporal terms

- challenge: false positives:

1984 tells the story of Winston Smith...
...U2's classic *Sunday* Bloody *Sunday*

Temporal Normalization

- → map to ISO8601 standard



- further examples:

Unit	Pattern	Sample Value
Fully specified dates	YYYY-MM-DD	1991-09-28
Weeks	YYYY-Wnn	2007-W27
Weekends	PnWE	P1WE
24-hour clock times	HH:MM:SS	11:13:45
Dates and times	YYYY-MM-DDTHH:MM:SS	1991-09-28T11:00:00
Financial quarters	Qn	1999-Q3

Temporal Normalization

- document or communication act has a logical temporal anchor (e.g. time of creation, time of publication, today, now, etc.)

- → logical temporal arithmetic:
  - anchor + *today* → today → *tomorrow* = today + 1d , *yesterday* = today - 1d
  - 50 weeks later than week 26 of 2007 = week 26 + 50 mod 53 of 2007 + 1

- but: complexity of absolute referencing may be high:
  - *…was matched by competitors over the weekend…* → "last weekend" (relative to anchor)
  - *Security checks will continue at least through the weekend* → "coming weekend" (relative to anchor)

    for both cases: indicator: tense of verb
  - *…next Friday…* : "immediate next Friday" or "Friday next week"? → heuristic: the closer anchor to "immediate next Friday" the more probable is "Friday next week"

Event Extraction

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

- most events: verbs / VP; also possible: NP
  - VP counterexamples:
    - *…took effect…*,
    - light verbs such as *make*, *take*, or *have*: event is expressed by their object: *took a flight*

Event Extraction

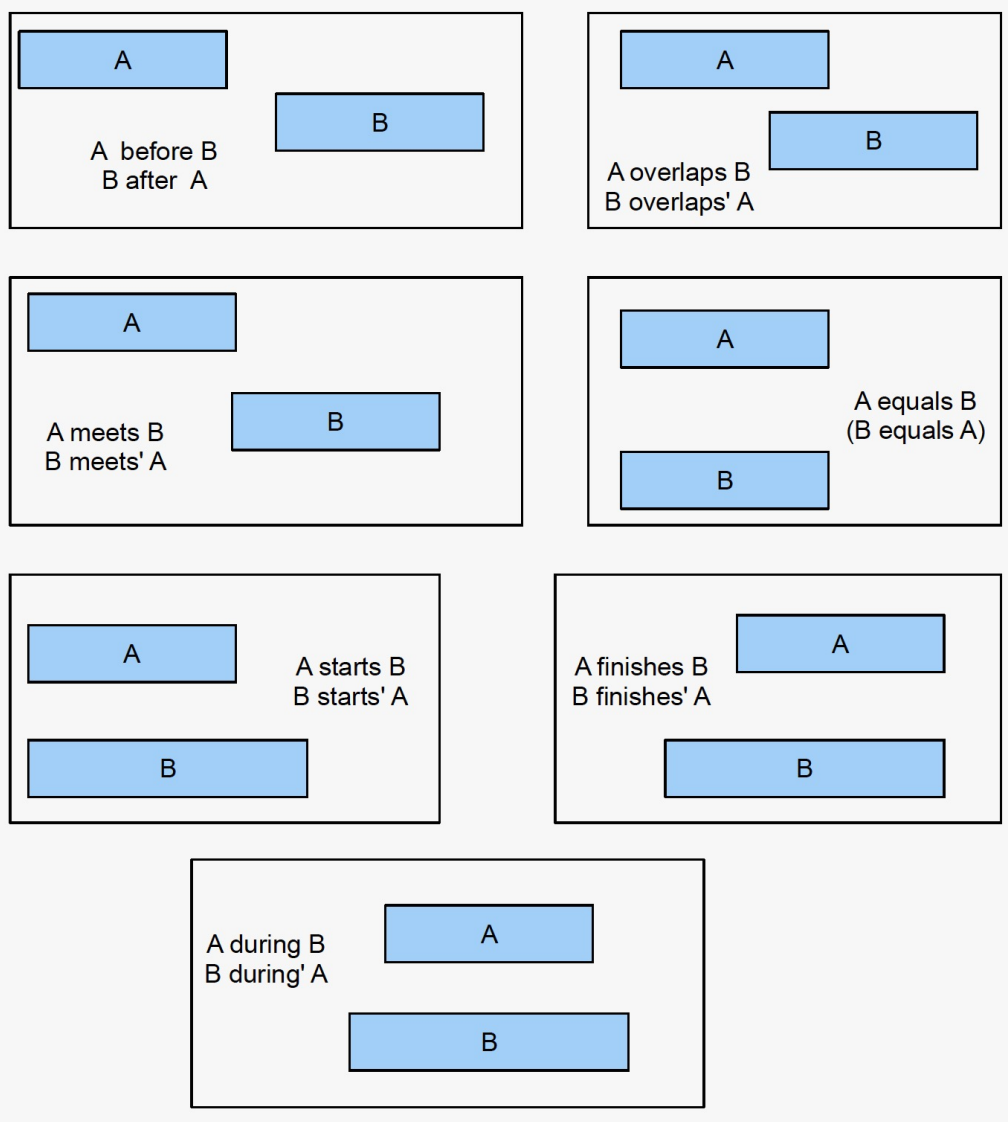
- classes of events: actions, states, reporting events (say, report, tell, explain), perception events etc.
  and further sub-classes: example: said events in example text:
  (class=REPORTING, tense=PAST, aspect=PERFECTIVE)

- approach: supervised ML with IOB tagging; features:

Feature	Explanation
Character affixes	Character-level prefixes and suffixes of target word
Nominalization suffix	Character level suffixes for nominalizations (e.g., *-tion*)
Part of speech	Part of speech of the target word
Light verb	Binary feature indicating that the target is governed by a light verb
Subject syntactic category	Syntactic category of the subject of the sentence
Morphological stem	Stemmed version of the target word
Verb root	Root form of the verb basis for a nominalization
WordNet hypernyms	Hypernym set for the target

Temporal Ordering of Events

- absolute positioning of events in anchored timeline or partial ordering of events (after, before etc.) (useful in e.g. question answering)
- example for partial ordering: determining that fare increase by American Airlines came after fare increase by United
- partial ordering: binary relation detection and classification task; target relations: Allen temporal logic relations



Temporal Ordering of Events

- TimeBank corpus: 183 news articles

```
<TIMEX3 tid="t57" type="DATE" value="1989-10-26"  functionInDocument="CREATION_TIME">
10/26/89  </TIMEX3>

Delta Air Lines earnings <EVENT eid="e1" class="OCCURRENCE"> soared </EVENT> 33% to a
record in  <TIMEX3 tid="t58" type="DATE" value="1989-Q1" anchorTimeID="t57"> the
fiscal first quarter </TIMEX3>, <EVENT eid="e3"  class="OCCURRENCE">bucking</EVENT>
the industry trend toward <EVENT eid="e4" class="OCCURRENCE">declining</EVENT>
profits.
```

- in addition to events and temporal expressions, corpus also includes Allen relations btw these

Delta Air Lines earnings soared 33% to a record in the fiscal first quarter, bucking the industry trend toward declining profits

- Soaring$_{e1}$ is **included** in the fiscal first quarter$_{t58}$
- Soaring$_{e1}$ is **before** 1989-10-26$_{t57}$
- Soaring$_{e1}$ is **simultaneous** with the bucking$_{e3}$
- Declining$_{e4}$ **includes** soaring$_{e1}$

Template Filling

- template: “scripts” of common stereotypical situations / event sequences with fixed set of slots
- template filling task: detect presence of template and fill slots with slot filler values

FARE-RAISE ATTEMPT:	LEAD AIRLINE:	UNITED AIRLINES
	AMOUNT:	$6
	EFFECTIVE DATE:	2006-10-26
	FOLLOWER:	AMERICAN AIRLINES

  - template recognition: supervised text classification task; usual set of features: tokens, word shapes, part-of-speech tags, syntactic chunk tags, NE tags etc.
  - role filler extraction: detect roles (either via supervised classifier on NPs or as sequence labelling task) and fill roles (again via supervised ML)

Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Oct 2019); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2019); this slide-set is especially based on chapter 18

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

  == standard approach