School of CIT –
Social Computing
Research Group

Technical
University
of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

Chapter 17
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 → see previous chapter

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

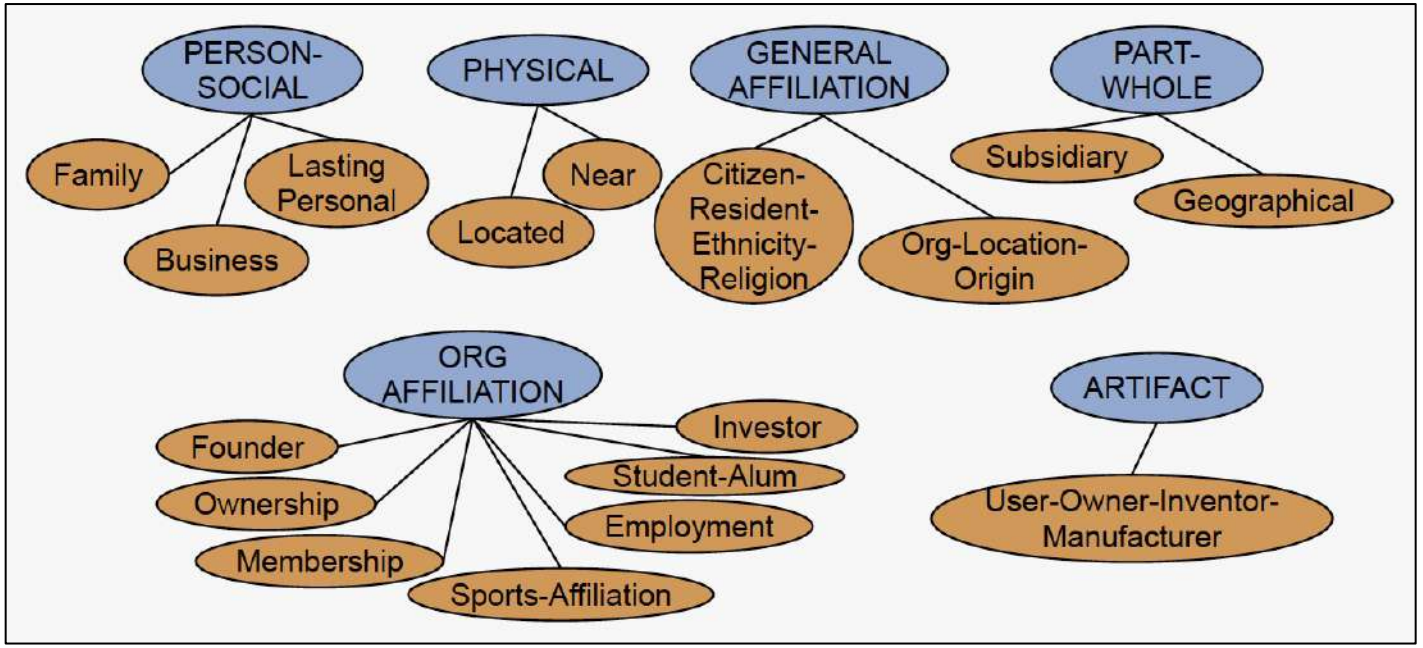
> Citing high fuel prices, [ORG **United Airlines**] said [TIME **Friday**] it has increased fares by [MONEY **$6**] per round trip on flights to some cities also served by lower-cost carriers. [ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim 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**.
...			

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:
  - hyponym 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

Relational / Ontological Databases

- TACRED (2017): 41 general relation types, hand-labeled

high-quality:
high precision, low recall

Example	Entity Types & Label
Carey will succeed Cathleen P. Black, who held the position for 15 years and will take on a new role as chairwoman of Hearst Magazines, the company said.	PERSON/TITLE Relation: *per:title*
Irene Morgan Kirkaldy, who was born and reared in Baltimore, lived on Long Island and ran a child-care center in Queens with her second husband, Stanley Kirkaldy.	PERSON/CITY Relation: *per:city_of_birth*
Baldwin declined further comment, and said JetBlue chief executive Dave Barger was unavailable.	Types: PERSON/TITLE Relation: *no_relation*

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 (Hearst 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)

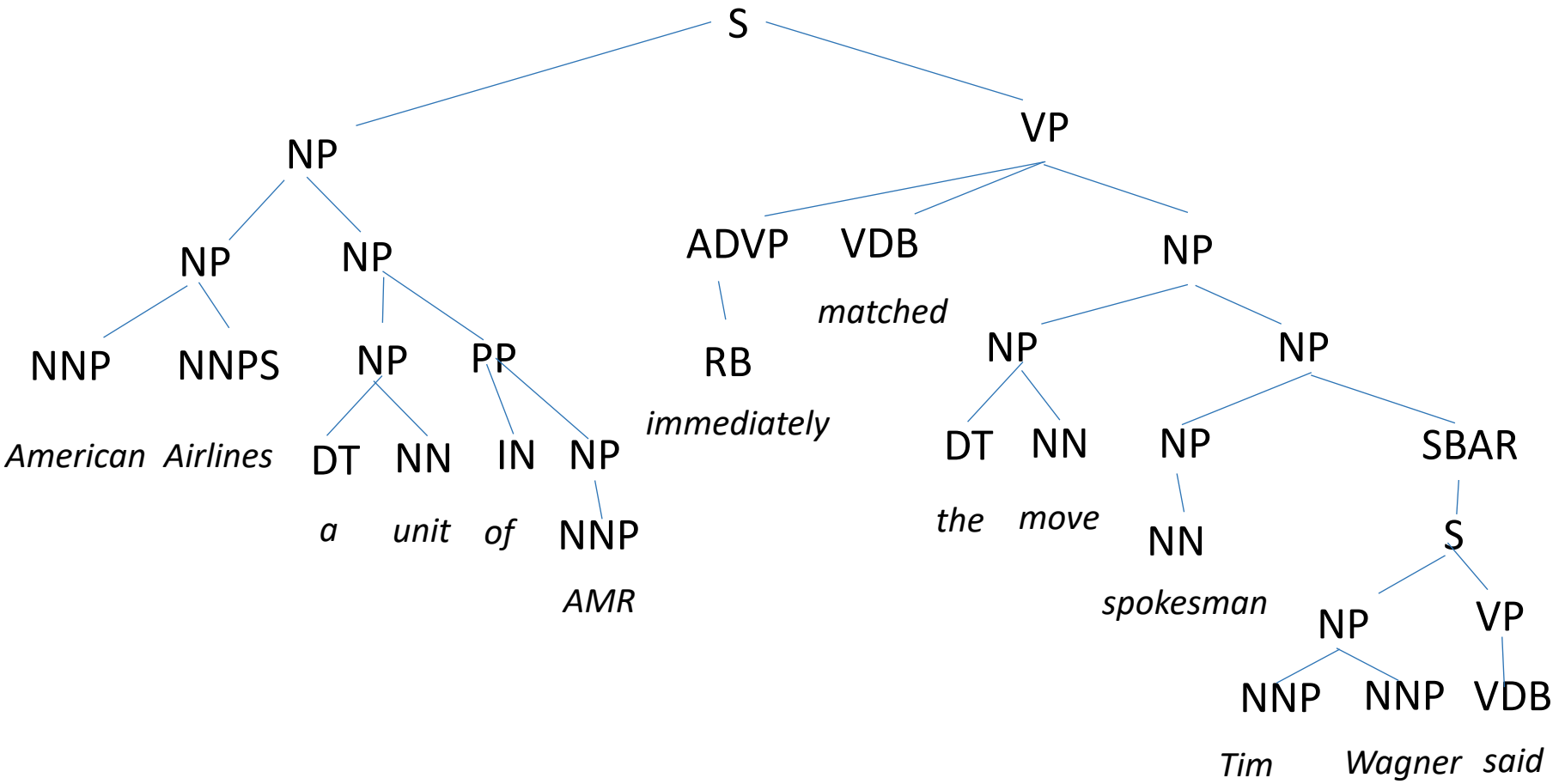
```
function FINDRELATIONS(words) returns relations

    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		$Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$

Neural Relation Extraction

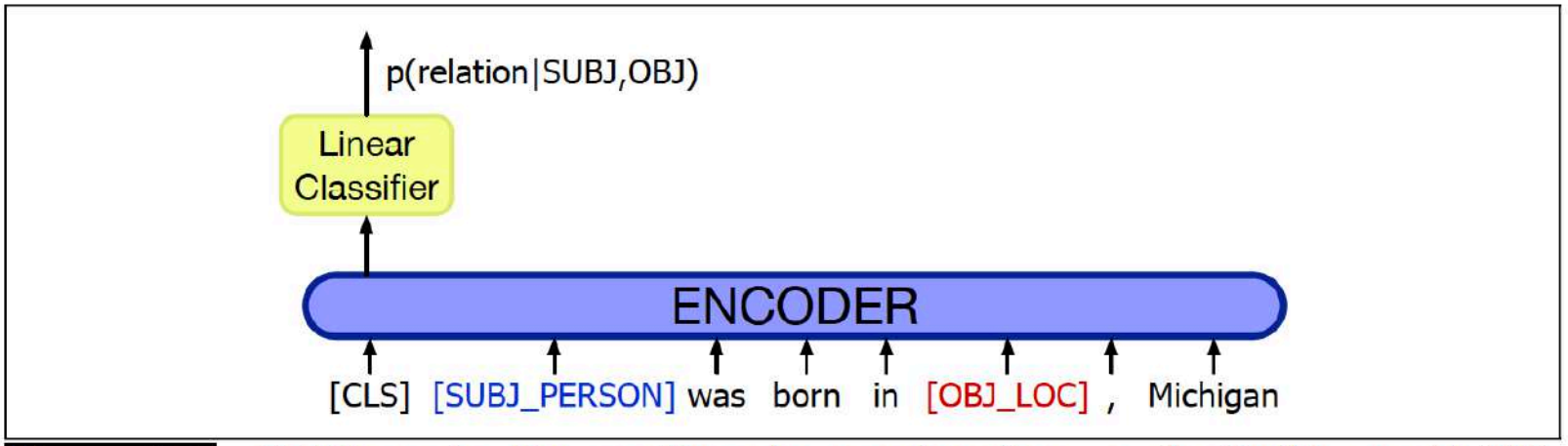


Figure 17.7 Relation extraction as a linear layer on top of an encoder (in this case BERT), with the subject and object entities replaced in the input by their NER tags (Zhang et al. 2017, Joshi et al. 2020).

Other modern idea:

- use Hearst rules to formulate „queries“ to language models or encoders (BERT):

„Man, I was so full after our visit to Hofbräuhaus“ artificially expand:

„Man, I was so full after our visit to Hofbräuhaus. <X_h> such as Hofbräuhaus are…“

let BERT predict <X_h>

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*

↳ avoid production of non-sence pairs

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

confidence value

$$Conf_{RlogF}(p) = \frac{|hits(p)|}{|finds(p)|} log(|finds(p)|)$$

("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 T 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)

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

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 Jan 2022); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2022); this slide-set is especially based on chapter 17

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