Natural Language Processing CS 6320

Lecture 15

Information Extraction

Instructor: Sanda Harabagiu

What is Information Extraction

The NLP task of information extraction (IE), turns the unstructured information embedded in texts into structured data, for example for populating a relational database to enable further processing.

Three IE sub-tasks:

1. Named Entity Recognition (NER)
2. Relation Extraction
3. Event Extraction

Named Entity Recognition

The task of named entity recognition (NER) is to find each mention of a named entity in the text and label its type.

John attends University of Texas at Dallas, which is situated in Richardson.

- *Finding the named entities:*

John attends University of Texas at Dallas, which is situated in Richardson.

- *Labeling their types:*

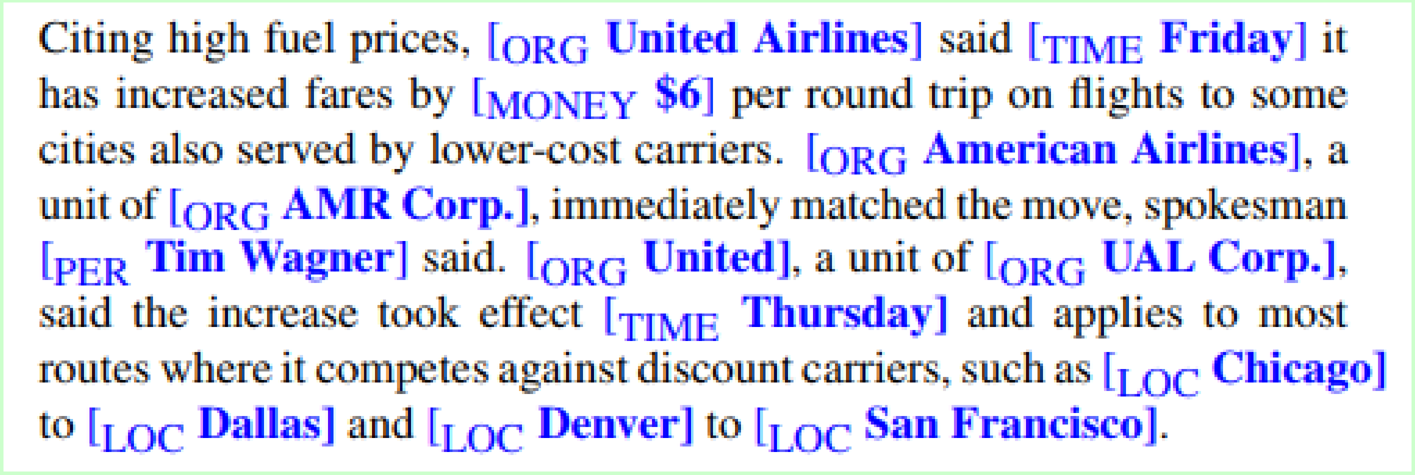
John attends University of Texas at Dallas, which is situated in Richardson.

PERSON **ORGANIZATION** **LOCATION**

What constitutes a named entity type is task specific; people, places, and organizations are common, but gene or protein names or financial asset classes might be relevant for some tasks.

Named Entity Recognition-Definition

A named entity is, roughly speaking, anything that can be referred to with a proper name: a person, a location, an organization. The term is commonly extended to include things that aren't entities per se, including dates, times, and other kinds of temporal expressions, and even numerical expressions like prices. Here's an annotated text:



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**].

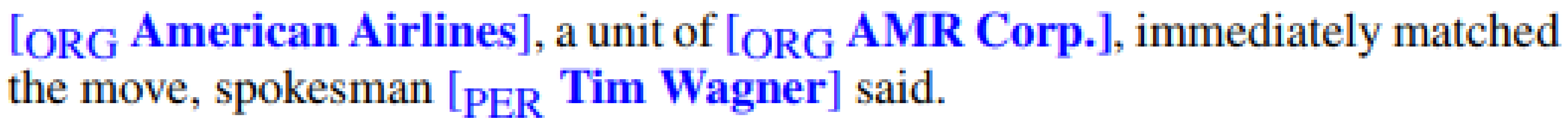
This text contains 13 mentions of named entities including 5 organizations, 4 locations, 2 times, 1 person, and 1 mention of money!

NER as Sequence Labeling

- ❑ *The standard algorithm for named entity recognition is as a word-by-word sequence labeling task, in which:*
- ▪ *the assigned tags capture <u>both the boundary and the type</u>.*

A sequence classifier like an MEMM/CRF or a bi-LSTM can be trained to label the tokens in a text with tags that indicate the presence of particular kinds of named entities.

Consider our running example:


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

How could we recognize these NEs????

- ➢ *Use IOB tagging.*

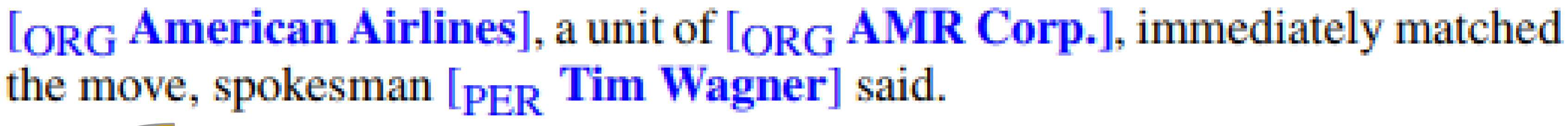
*In IOB tagging we introduce <u>a tag for the beginning</u> (**B**) and <u>inside</u> (**I**) of each entity type, and <u>one for tokens outside</u> (**O**) any entity.*

The number of tags is thus 2n + 1 tags, where n is the number of entity types!

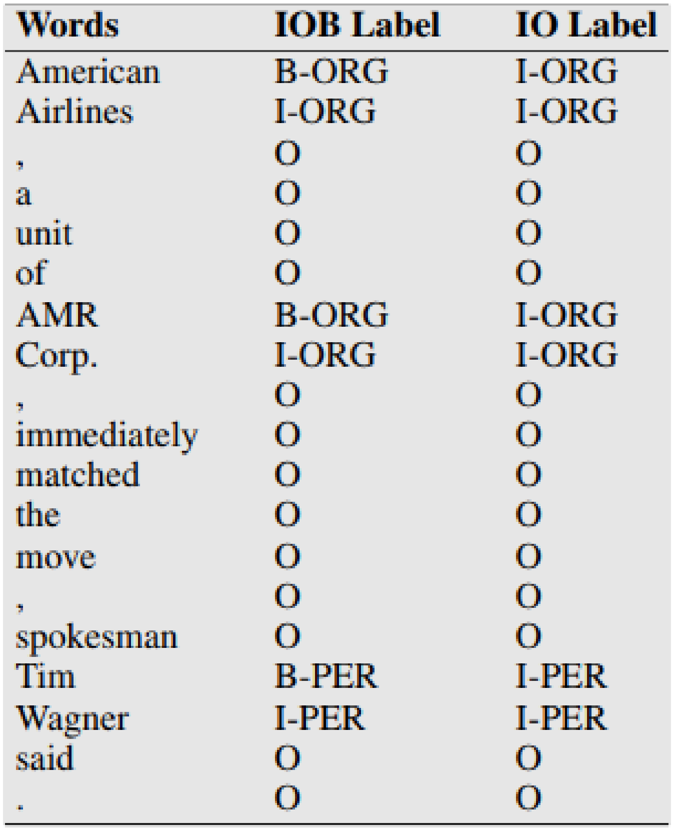
NER as Sequence Labeling

- *Use IOB tagging.*

*In IOB tagging we introduce <u>a tag for the beginning</u> (**B**) and <u>inside</u> (**I**) of each entity type, and <u>one for tokens outside</u> (**O**) any entity.*



[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

<u>The IO tagging</u> loses some information by eliminating the B tag. Without the B tag IO tagging is unable to distinguish between two entities of the same type that are right next to each other. Since this situation doesn't arise very often (usually there is at least some punctuation or other delimitator), IO tagging may be sufficient, and has the advantage of using only n+1 tags.

A feature-based algorithm for NER

- *Extract features and train an MEMM or CRF sequence model*

IMPORTANT: *Word shape features are particularly important in the context of NER!*

word shape features *are used to represent the* *abstract letter pattern of the word* *by:*

- *Mapping lower-case letters to 'x',*
- *Mapping upper-case to 'X',*
- *Mapping numbers to 'd',*
- *and retaining punctuation.*

Examples: I.M.F would map to X.X.X.

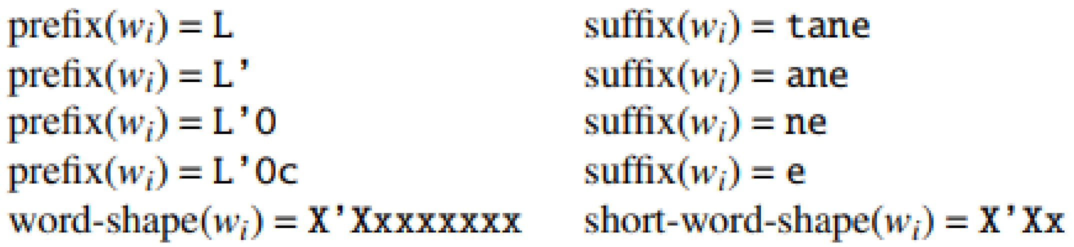
and DC10-30 would map to XXdd-dd.

A second class of shorter word shape features is also used. In these features consecutive character types are removed, so DC10-30 would be mapped to Xd-d but I.M.F would still map to X.X.X.

This feature by itself accounts for a considerable part of the success of feature-based NER systems for English news text. Shape features are also particularly important in recognizing names of proteins and genes in biological texts. For example the named entity token L'Occitane would generate the following non-zero valued feature values:

A feature-based algorithm for NER

- *For example the named entity token L’Occitane would generate the following non-zero valued feature values:*



$\text{prefix}(w_i)$ = L	$\text{suffix}(w_i)$ = tane
$\text{prefix}(w_i)$ = L'	$\text{suffix}(w_i)$ = ane
$\text{prefix}(w_i)$ = L'O	$\text{suffix}(w_i)$ = ne
$\text{prefix}(w_i)$ = L'Oc	$\text{suffix}(w_i)$ = e
$\text{word-shape}(w_i)$ = X'Xxxxxxxx	$\text{short-word-shape}(w_i)$ = X'Xx

- What else we need?????? GAZETEERS

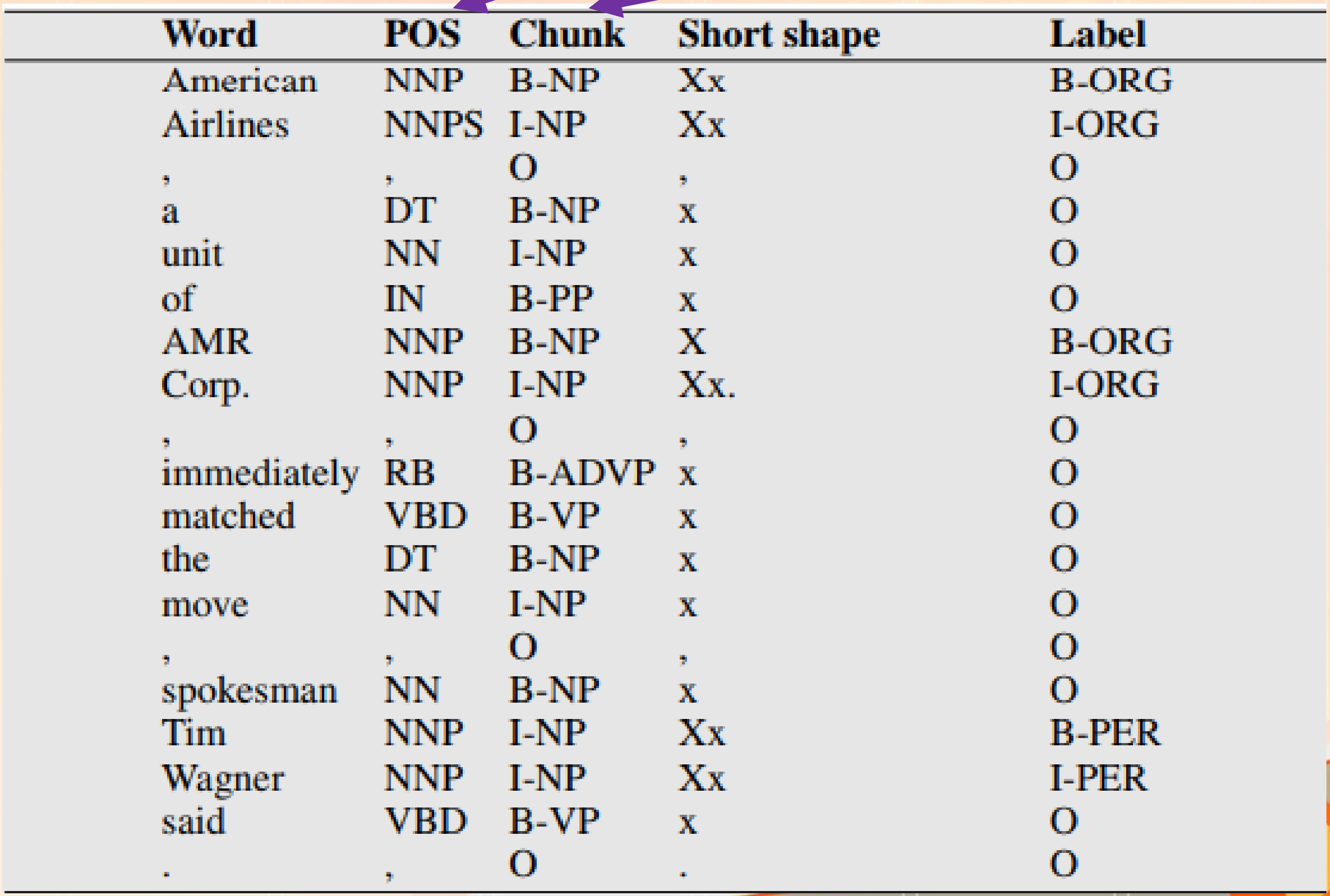
A gazetteer is a list of place names, often providing millions of entries for locations with detailed geographical and political information.

- *A related resource is name-lists; the United States Census Bureau also provides extensive lists of first names and surnames derived from its decadal census in the U.S.*
- *Similar lists of corporations, commercial products, and all manner of things biological and mineral are also available from a variety of sources.*

- *Gazetteer and name features are typically implemented as a binary feature for each name list. Unfortunately, such lists can be difficult to create and maintain, and their usefulness varies considerably. While gazetteers can be quite effective, lists of persons and organizations are not always helpful (Mikheev et al., 1999).*

Features for NER

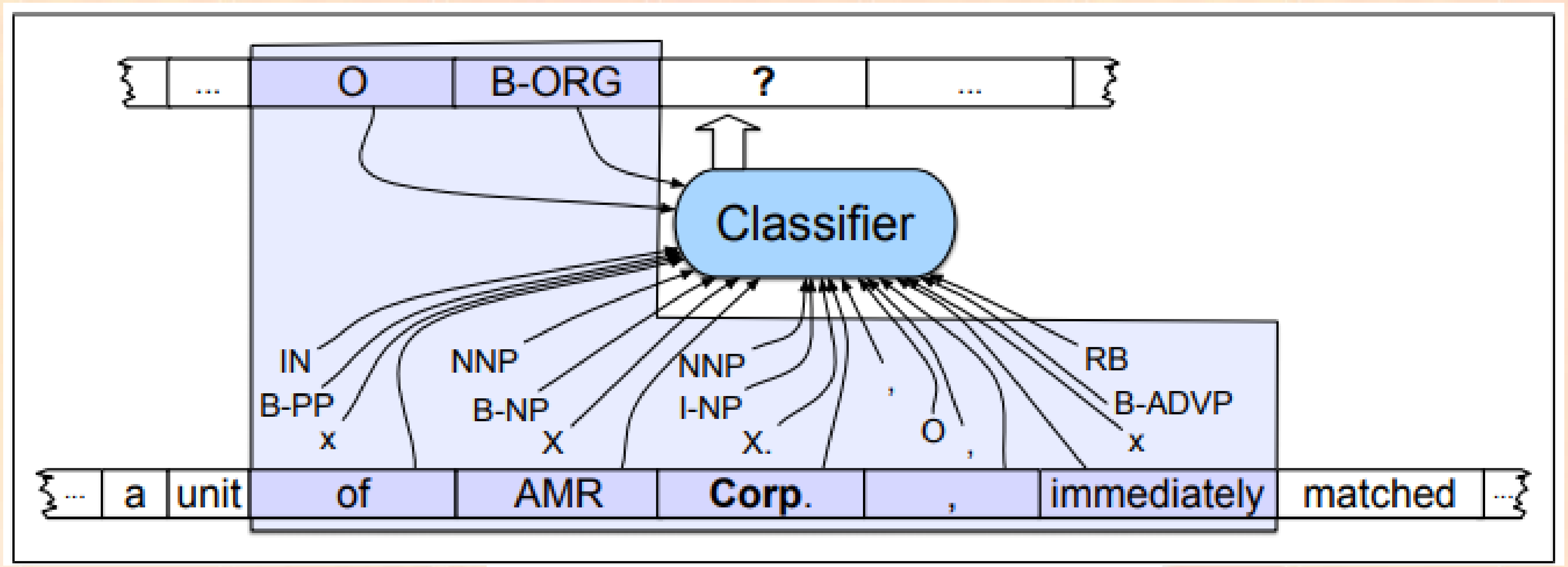
- *Should we consider also some lexical, syntactic features?*

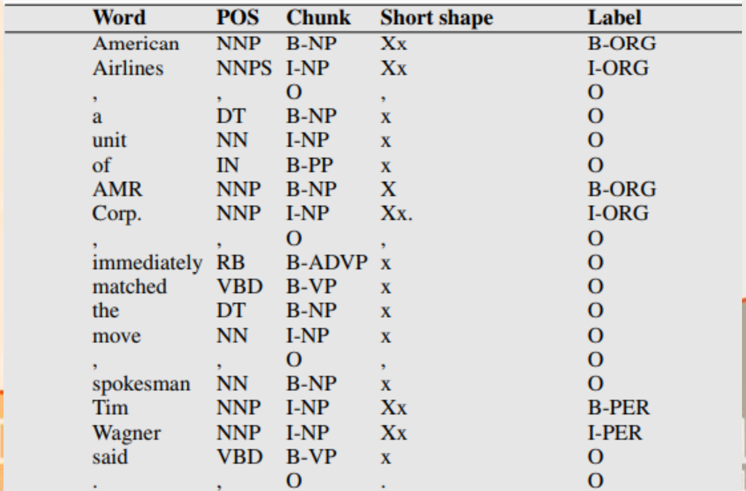


Word	POS	Chunk	Short shape	Label
American	NNP	B-NP	Xx	B-ORG
Airlines	NNPS	I-NP	Xx	I-ORG
,	,	O	,	O
a	DT	B-NP	x	O
unit	NN	I-NP	x	O
of	IN	B-PP	x	O
AMR	NNP	B-NP	X	B-ORG
Corp.	NNP	I-NP	Xx.	I-ORG
,	,	O	,	O
immediately	RB	B-ADVP	x	O
matched	VBD	B-VP	x	O
the	DT	B-NP	x	O
move	NN	I-NP	x	O
,	,	O	,	O
spokesman	NN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

The NER Sequence Classifier

- *Given the features and a training set, a sequence classifier like an MEMM can be trained to label new sentences:*





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
spokesman	NN	B-NP	x	O
Tim	NNP	I-NP	Xx	B-PER
Wagner	NNP	I-NP	Xx	I-PER
said	VBD	B-VP	x	O
.	,	O	.	O

A neural algorithm for NER

The standard neural algorithm for NER is based on the bi-LSTM

Recall that in that model, word and character embeddings are computed for input word w_i . These are passed through a left-to-right LSTM and a right-to-left LSTM, whose outputs are concatenated (or otherwise combined) to produce a single output layer at position $_i$.

- *In the simplest method, this layer can then be directly passed onto a softmax that creates a probability distribution over all NER tags, and the most likely tag is chosen as t_i .*
- *For named entity tagging this greedy approach to decoding <u>is insufficient</u>, since it doesn't allow us to impose the strong constraints neighboring tokens have on each other (e.g., the tag I-PER must follow another I-PER or B-PER). <u>Instead a CRF layer is normally used on top of the bi-LSTM</u> output, and the Viterbi decoding algorithm is used to decode.*

A neural CRF-LSTM model for NER

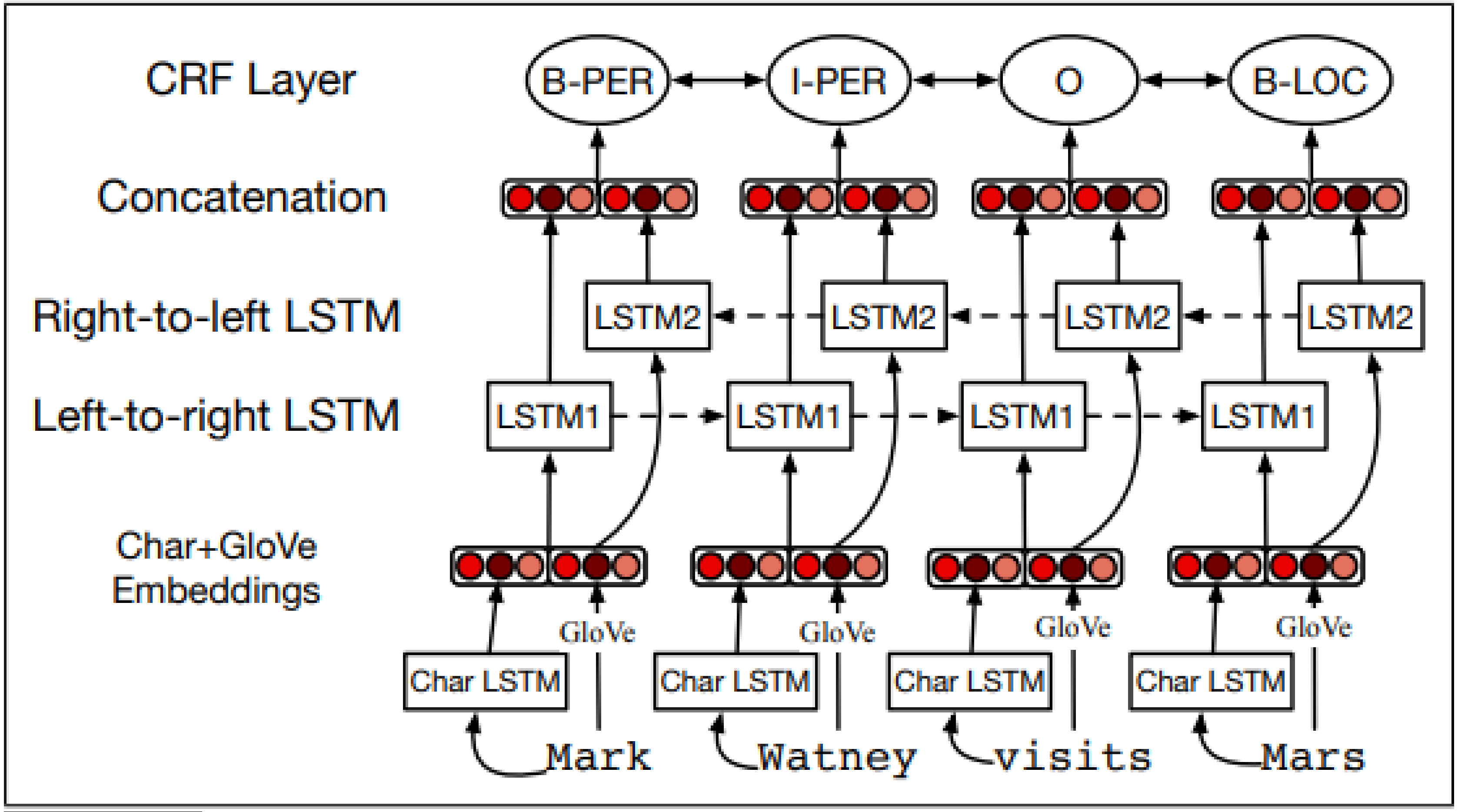


Figure 17.8 Putting it all together: character embeddings and words together a bi-LSTM sequence model. After (Lample et al., 2016)

Rule-based NER

- *While machine learned (neural or MEMM/CRF) sequence models are the norm in academic research, commercial approaches to NER are often based on pragmatic combinations of lists and rules, with some smaller amount of supervised machine learning.*

1. *First, use high-precision rules to tag unambiguous entity mentions.*
2. *Then, search for substring matches of the previously detected names.*
3. *Consult application-specific name lists to identify likely name entity mentions from the given domain.*
4. *Finally, apply probabilistic sequence labeling techniques that make use of the tags from previous stages as additional features.*

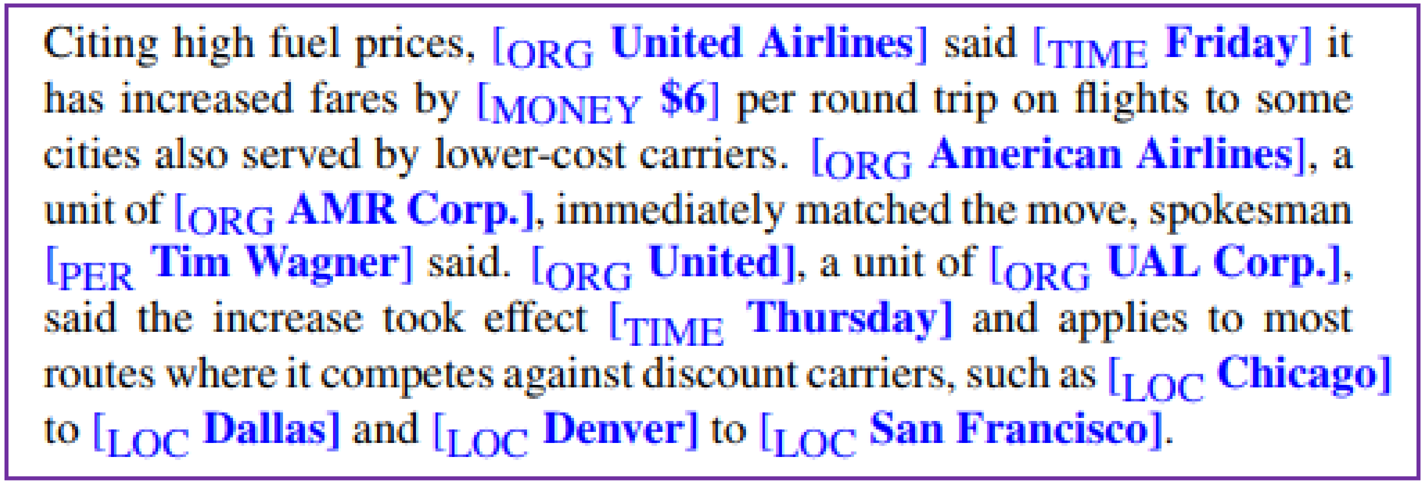
The intuition behind this staged approach is twofold:

a) *First, some of the entity mentions in a text will be more clearly indicative of a given entity's class than others.*

b) *Second, once an unambiguous entity mention is introduced into a text, it is likely that subsequent shortened versions will refer to the same entity (and thus the same type of entity).*

Relation Extraction

- ➢ *Discern the relationships that exist among the detected entities? Let's return to our sample airline text:*



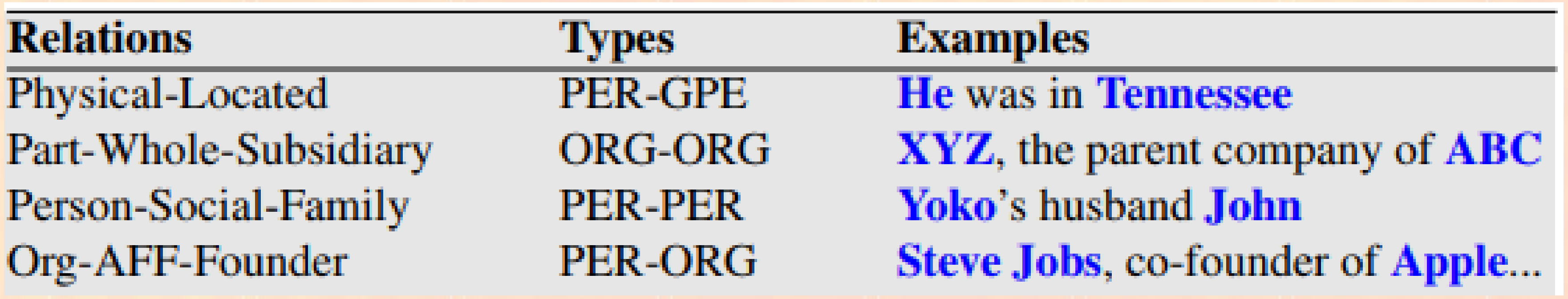
Citing high fuel prices, [$_{ORG}$ **United Airlines**] said [$_{TIME}$ **Friday**] it has increased fares by [$_{MONEY}$ **\$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{ORG}$ **American Airlines**], a unit of [$_{ORG}$ **AMR Corp.**], immediately matched the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

- Relation 1: [Tim Warner]$_{PERSON}$ → Spokesman → [American Airlines]$_{ORG}$
- Relation 2: [American Airlines]$_{ORG}$ → Is-Unit-Of → [AMR Corp.]$_{ORG}$

*These binary relations are instances of more generic relations such as **part-of** or **employs** that are fairly frequent in news-style texts.*

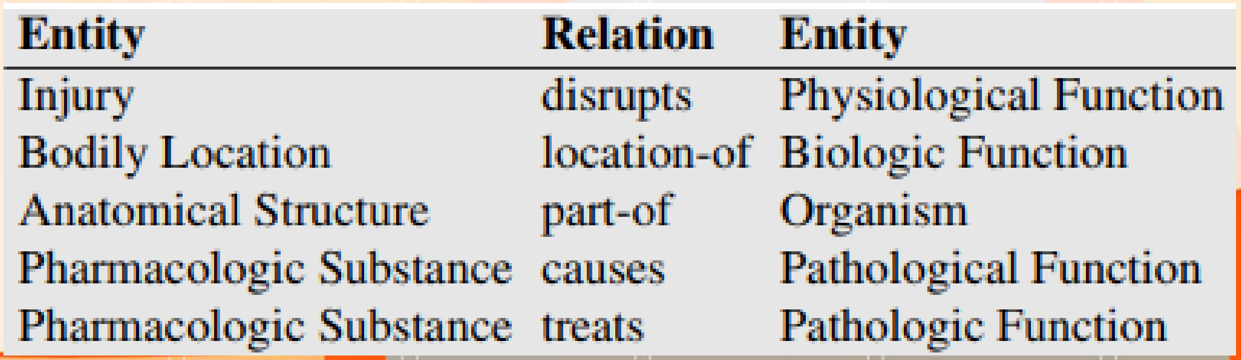
Relation Extraction

- *More examples:*



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**...

- *Sets of relations have been defined for many other domains as well. For example* ***UMLS,*** *the Unified Medical Language System from the US National Library of Medicine has a network that defines 134 broad subject categories, entity types, and 54 relations between the entities, such as the following:*



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

Relation Extraction in Medical Domain

Given a medical sentence like this one:

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

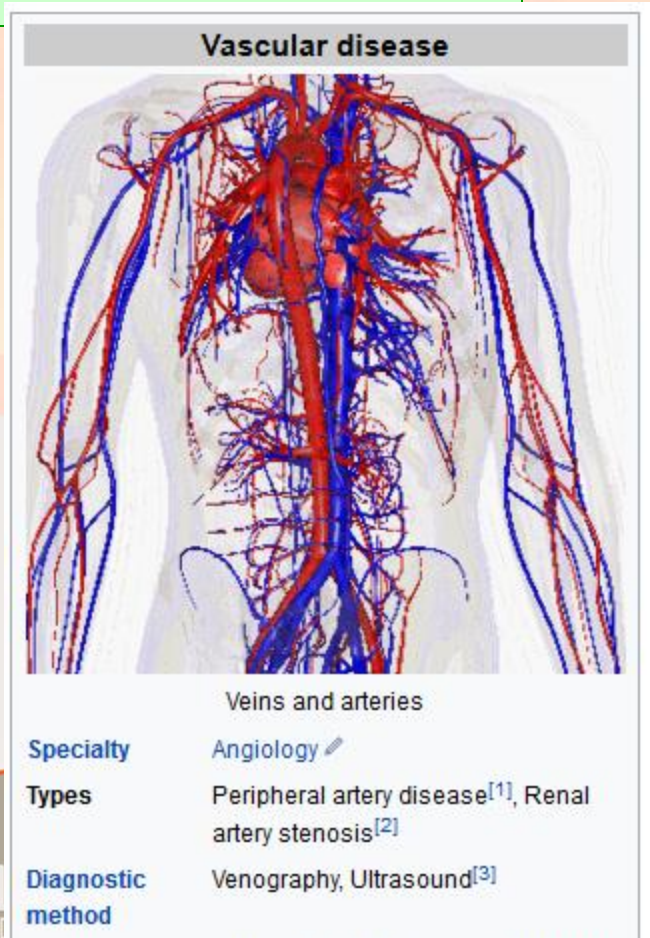
We could thus extract the UMLS relation:

Echocardiography $\rightarrow$ Diagnoses $\rightarrow$ Acquired stenosis

Wikipedia also offers a large supply of relations, drawn from ***infoboxes,*** *i.e. structured tables associated with certain Wikipedia articles.*

Infobox facts can be turned into relations like president-of or located-in or Diagnosis_Method.

We can use a metalanguage called ***RDF (Resource Description Framework)****. An RDF triple is a tuple of entity-relation-entity, called a subject-predicate-object expression.*



Vascular disease

Veins and arteries

Specialty	Angiology
Types	Peripheral artery disease[1], Renal artery stenosis[2]
Diagnostic method	Venography, Ultrasound[3]

RDF extraction

RDF Relations:

UTD → *has-motto* → Disciplina Presidium Civitatis

UTD → *university-type* → Public – Research State University

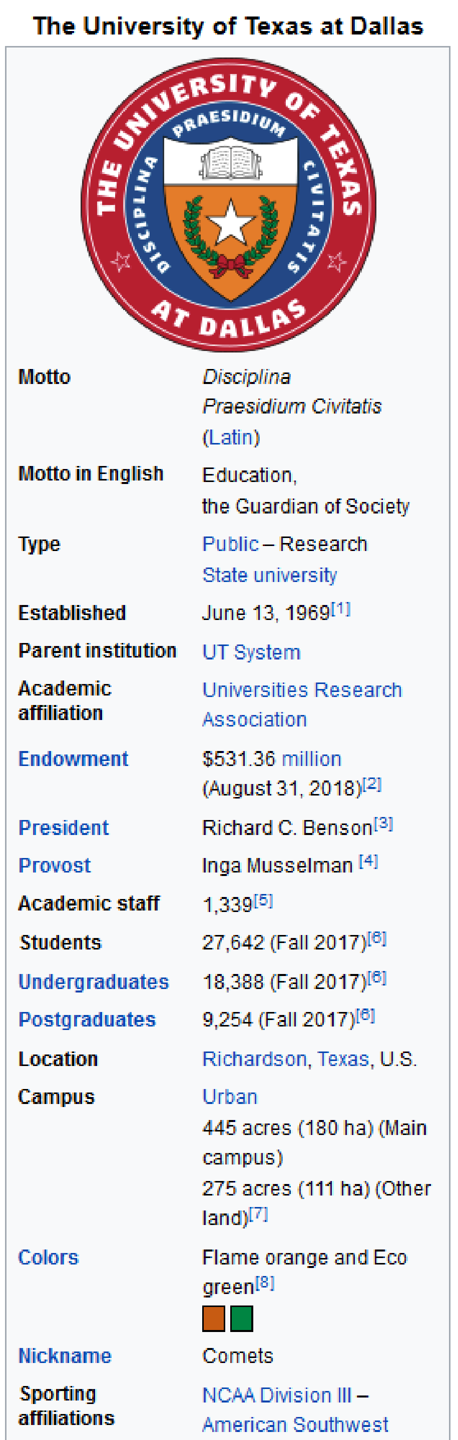
UTD → *part-of* → UT System

UTD → *endowment* → $531.36 million

UTD → *president* → Richard C. Benson

…………………………………………………….

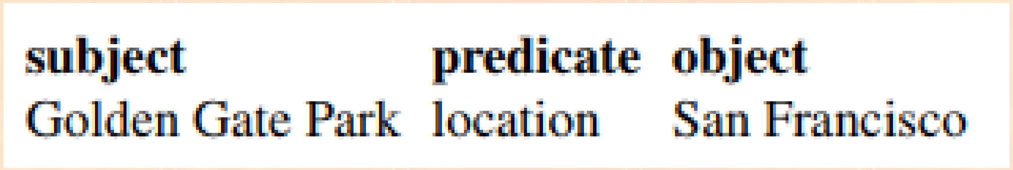
- ❑ Why called *subject-predicate-object expressions?*
- ➢ *UTD is always the subject-argument of the relation*
- ➢ *These are binary relations*
- ➢ *The second argument usually has the role of an object, in sentences like:*
- ▪ *UTD's motto is "*Disciplina Presidium Civitatis".
- ▪ UTD is part of the UT System.
- ▪ UTD's president is Richard C. Benson.



The University of Texas at Dallas

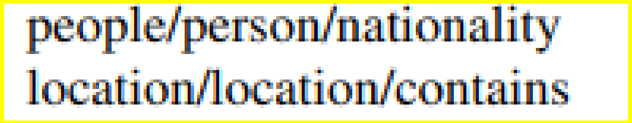
Motto	*Disciplina Praesidium Civitatis* (Latin)
Motto in English	Education, the Guardian of Society
Type	Public – Research State university
Established	June 13, 1969[1]
Parent institution	UT System
Academic affiliation	Universities Research Association
Endowment	$531.36 million (August 31, 2018)[2]
President	Richard C. Benson[3]
Provost	Inga Musselman [4]
Academic staff	1,339[5]
Students	27,642 (Fall 2017)[6]
Undergraduates	18,388 (Fall 2017)[6]
Postgraduates	9,254 (Fall 2017)[6]
Location	Richardson, Texas, U.S.
Campus	Urban 445 acres (180 ha) (Main campus) 275 acres (111 ha) (Other land)[7]
Colors	Flame orange and Eco green[8] 🟧🟩
Nickname	Comets
Sporting affiliations	NCAA Division III – American Southwest

Relations in RDF format



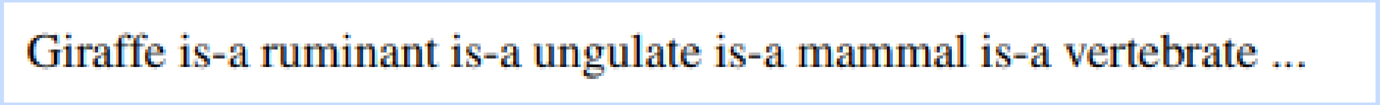
subject	predicate	object
Golden Gate Park	location	San Francisco

- *the crowdsourced* ***DBpedia*** *(Bizer et al., 2009) is an ontology derived from Wikipedia containing over 2 billion RDF triples.*
- *Another dataset from Wikipedia infoboxes is* ***Freebase*** *(Bollacker et al., 2008), having relations like:*



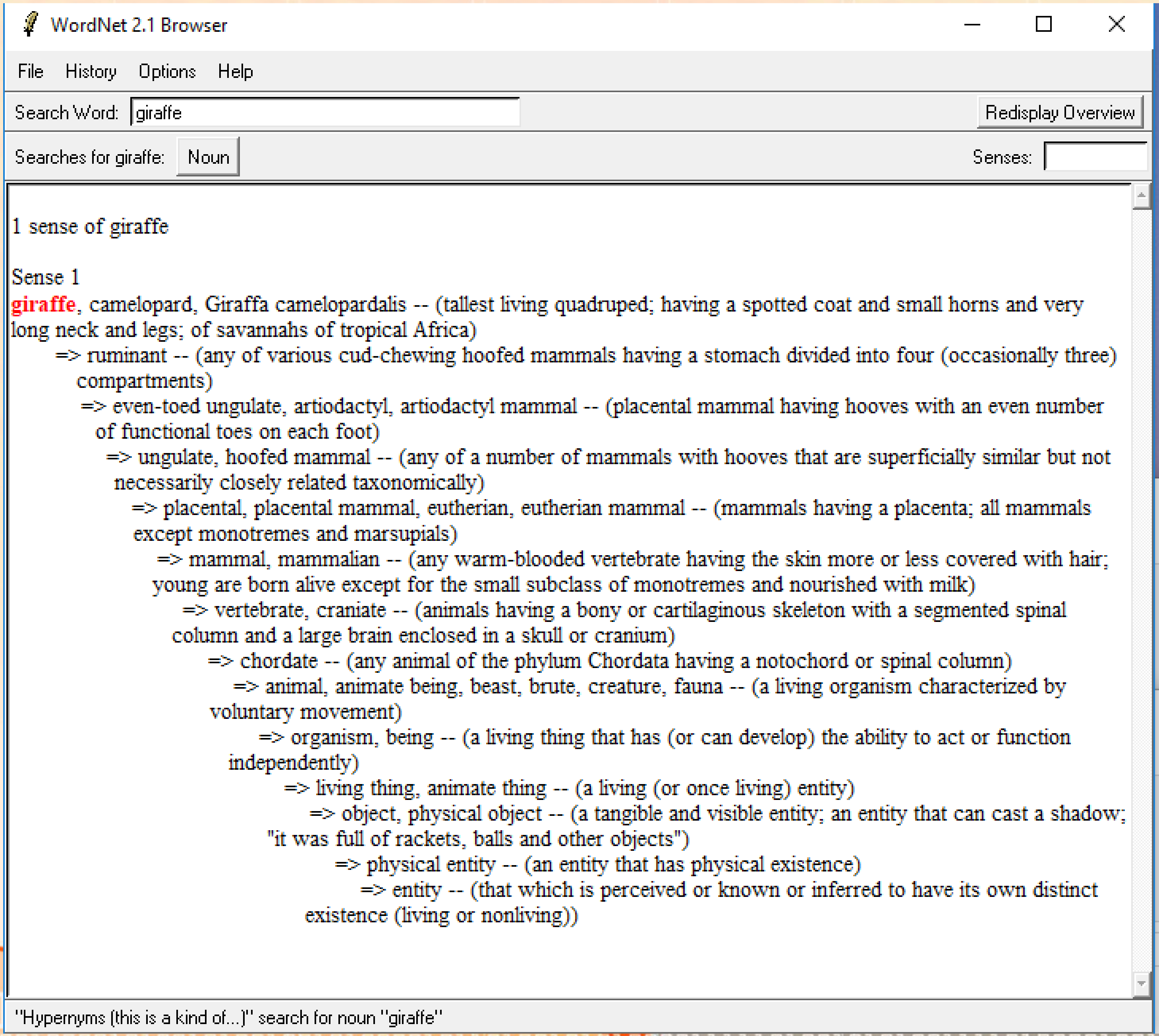
people/person/nationality
location/location/contains

- ***WordNet*** *or other ontologies offer useful ontological relations that express hierarchical relations between words or concepts through* ***Is-A*** *relations, also known as* ***hypernyms****:*



Giraffe is-a ruminant is-a ungulate is-a mammal is-a vertebrate ...

WordNet Relations



1 sense of giraffe

Sense 1
giraffe, camelopard, Giraffa camelopardalis -- (tallest living quadruped; having a spotted coat and small horns and very long neck and legs; of savannahs of tropical Africa)

```
    => ruminant -- (any of various cud-chewing hoofed mammals having a stomach divided into four (occasionally three) compartments)
      => even-toed ungulate, artiodactyl, artiodactyl mammal -- (placental mammal having hooves with an even number of functional toes on each foot)
        => ungulate, hoofed mammal -- (any of a number of mammals with hooves that are superficially similar but not necessarily closely related taxonomically)
          => placental, placental mammal, eutherian, eutherian mammal -- (mammals having a placenta; all mammals except monotremes and marsupials)
            => mammal, mammalian -- (any warm-blooded vertebrate having the skin more or less covered with hair; young are born alive except for the small subclass of monotremes and nourished with milk)
              => vertebrate, craniate -- (animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium)
                => chordate -- (any animal of the phylum Chordata having a notochord or spinal column)
                  => animal, animate being, beast, brute, creature, fauna -- (a living organism characterized by voluntary movement)
                    => organism, being -- (a living thing that has (or can develop) the ability to act or function independently)
                      => living thing, animate thing -- (a living (or once living) entity)
                        => object, physical object -- (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets, balls and other objects")
                          => physical entity -- (an entity that has physical existence)
                            => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
```

"Hypernyms (this is a kind of...)" search for noun "giraffe"

*WordNet also has **Instance-of** relation between individuals and classes, so that for <u>example</u> **San Francisco** is in the Instance-of relation with **city**.*

Extracting relations is an important step in <u>extending or building ontologies</u>.

How do we extract relations automatically???

- *There are four main classes of algorithms for relation extraction:*

1. *hand-written patterns,*
2. *supervised machine learning,*
3. *semi-supervised*
   a. *via bootstrapping*
   b. *via distant supervision*
4. *unsupervised.*

Hand-built patterns have the advantage of high-precision and they can be tailored to specific domains. On the other hand, they are often low-recall, and it's a lot of work to create them for all possible patterns.

Using Patterns to Extract Relations

The earliest and still common algorithm for relation extraction is lexico-syntactic patterns, first developed by Hearst (1992a). Consider the following sentence:

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

Hearst pointed out that most human readers will not know what Gelidium is, but that they can readily infer that it is a kind of (a hyponym of) red algae. She suggested that the following lexico-syntactic pattern:

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

implies the following semantics:

$$\forall NP_i, i \geq 1, \text{hyponym}(NP_i, NP_0) \Rightarrow \text{hyponym}(\text{Gelidium}, \text{red algae})$$

Hearst patterns for inferring the hyponym relation

Hand built lexico-syntactic patterns for finding hypernyms, using {} to mark optionality, NP_H indicates the hyperymy:

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

Modern versions of the pattern-based approach extend it by adding named entity constraints. For example if our goal is to answer questions about "Who holds what office in which organization?", we can use patterns like:

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

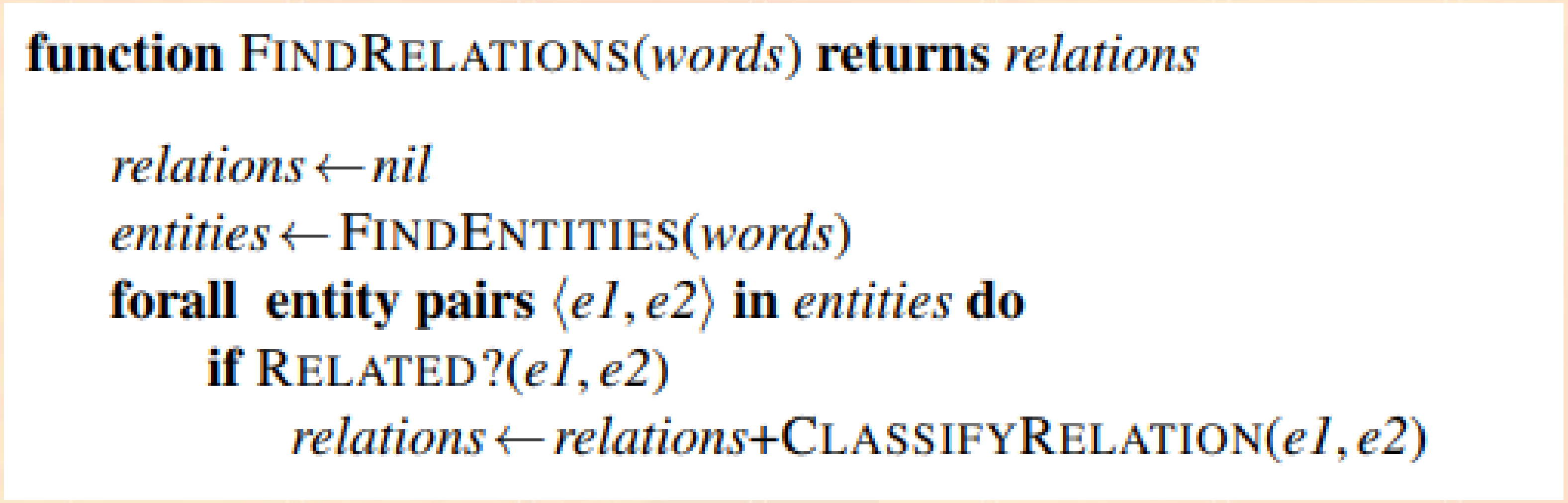
Relation Extraction via Supervised Learning

The most approach has three steps:

1. *Step one: find pairs of named entities (usually in the same sentence).*
2. *Step two, a filtering classifier is trained to make a **binary decision** as to whether a given pair of named entities are related (by any relation).*
   - *Positive examples are extracted directly from all relations in the annotated corpus, and negative examples are generated from within-sentence entity pairs that are not annotated with a relation.*
3. *Step 3, a classifier is trained to assign a label to the relations that were found by step 2.*

The use of the filtering classifier can speed up the final classification and also allows the use of distinct feature-sets appropriate for each task.

Finding and classifying the relations among entities in a text



```
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)
```

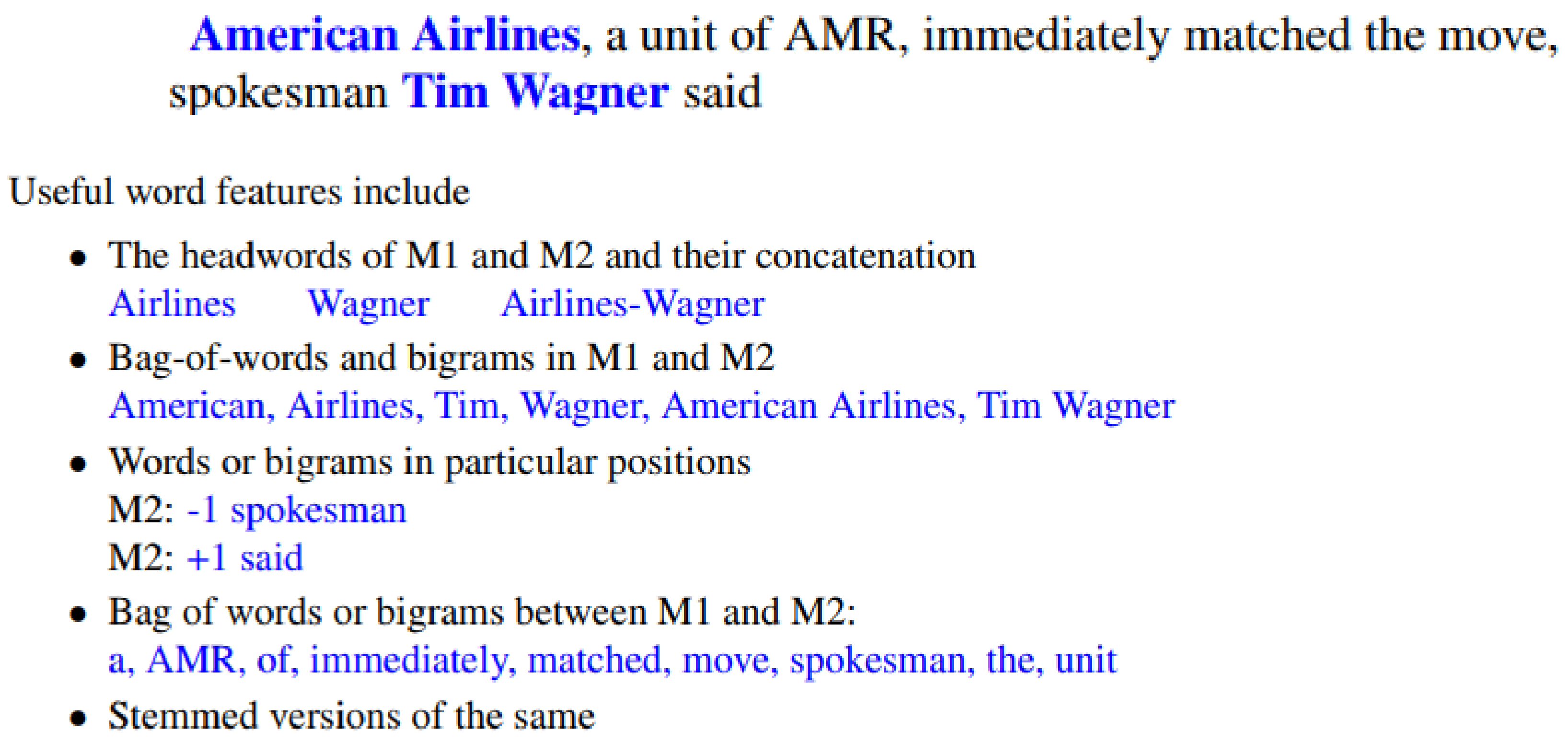
- *What classifiers?*

➢ *any of the standard classification techniques (logistic regression, neural network, SVM, etc)*

❑ *For the feature-based classifiers like logistic regression or random forests the most important step is to identify useful features.*

Considering the Features

Let us consider which features could be useful for recognizing and classifying the relationship between American Airlines (Mention 1, or M1) and Tim Wagner (Mention 2, M2) from this sentence:



American Airlines, a unit of AMR, immediately matched the move, spokesman **Tim Wagner** said

Useful word features include

- The headwords of M1 and M2 and their concatenation
  Airlines Wagner Airlines-Wagner
- Bag-of-words and bigrams in M1 and M2
  American, Airlines, Tim, Wagner, American Airlines, Tim Wagner
- Words or bigrams in particular positions
  M2: -1 spokesman
  M2: +1 said
- Bag of words or bigrams between M1 and M2:
  a, AMR, of, immediately, matched, move, spokesman, the, unit
- Stemmed versions of the same

Using Embeddings, NEs and Syntactic features

American Airlines, a unit of AMR, immediately matched the move, spokesman **Tim Wagner** said

*Embeddings can be used to represent words in any of these features. Useful **named entity features** include:*

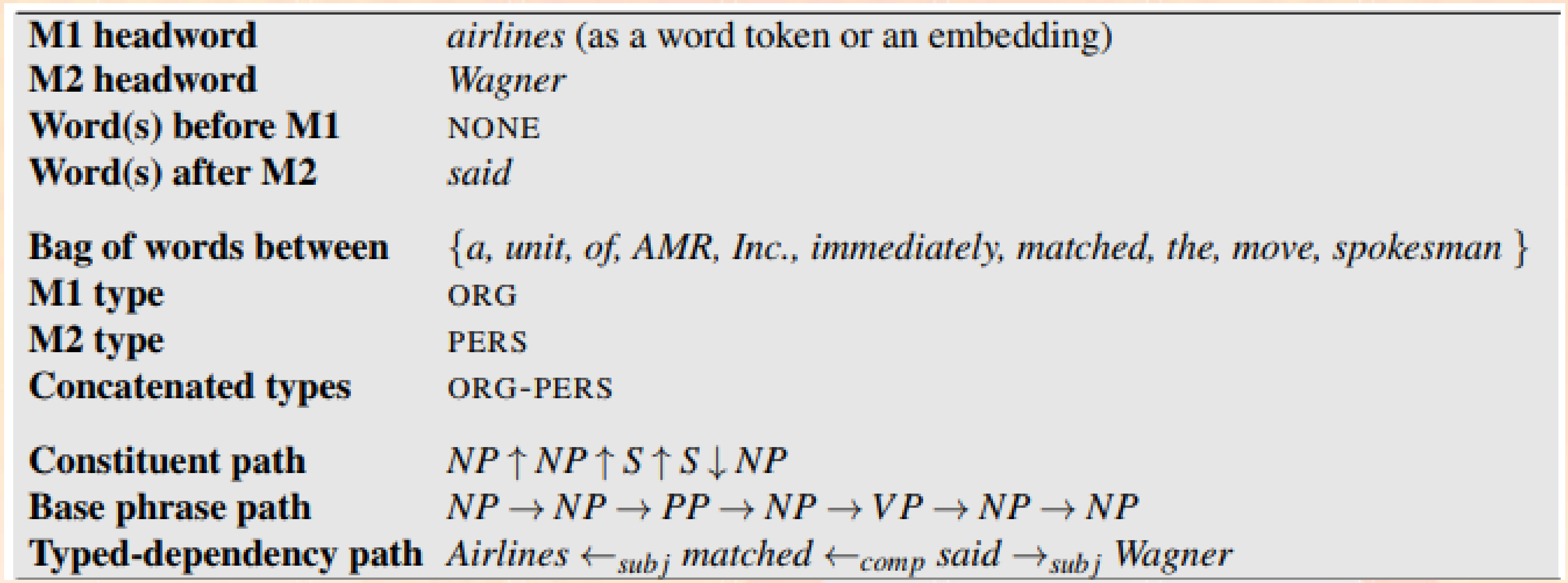
- Named-entity types and their concatenation
  (M1: ORG, M2: PER, M1M2: ORG-PER)
- Entity Level of M1 and M2 (from the set NAME, NOMINAL, PRONOUN)
  M1: NAME [it or he would be PRONOUN]
  M2: NAME [the company would be NOMINAL]
- Number of entities between the arguments (in this case 1, for AMR)

The syntactic structure of a sentence can also signal relationships among its entities. Syntax is often featured by using <u>strings representing syntactic paths</u>: the (dependency or constituency) path traversed through the tree in getting from one entity to the other:

- Base syntactic chunk sequence from M1 to M2
  NP NP PP VP NP NP
- Constituent paths between M1 and M2
  $NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
- Dependency-tree paths
  $Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$

Feature Engineering for Relation Extraction

- *Sample of features extracted during classification of the tuple; M1 is the first mention, M2 the second*

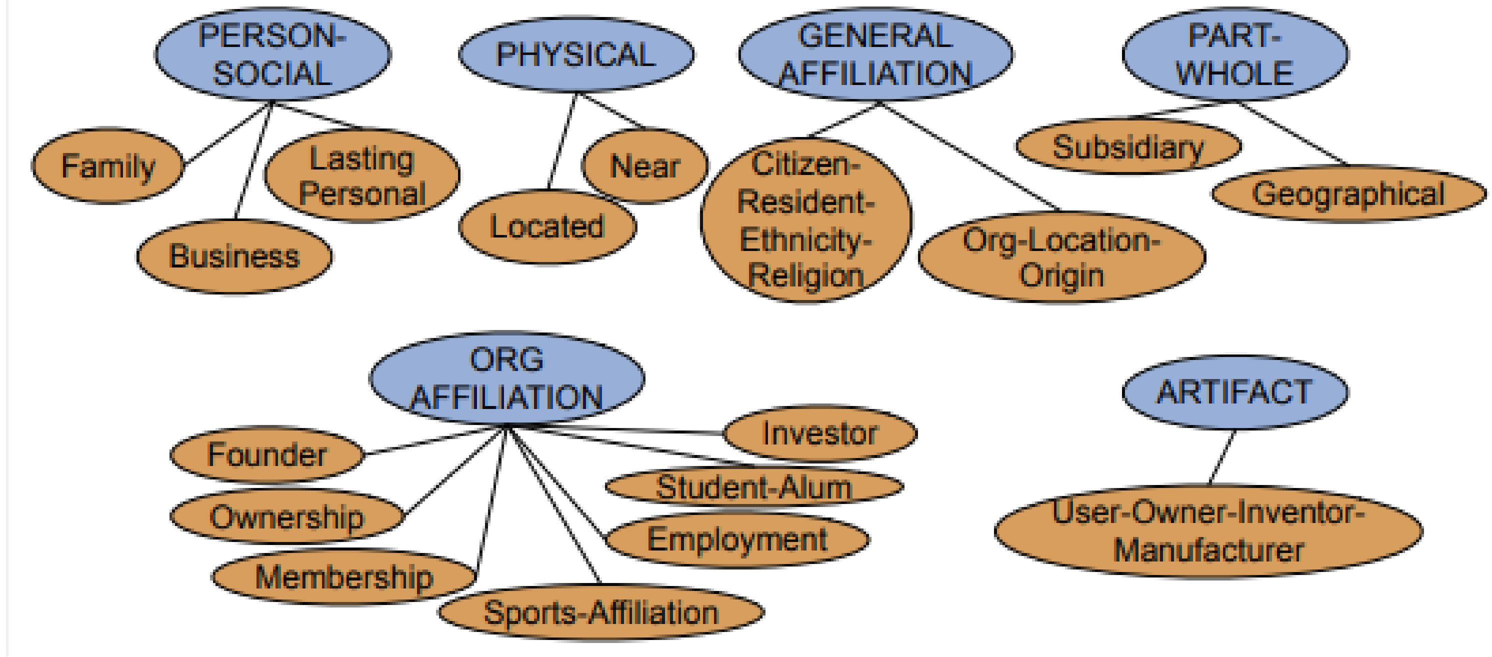


M1 headword	*airlines* (as a word token or an embedding)
M2 headword	*Wagner*
Word(s) before M1	NONE
Word(s) after M2	*said*
Bag of words between	{*a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman* }
M1 type	ORG
M2 type	PERS
Concatenated types	ORG-PERS
Constituent path	$NP \uparrow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

- https://nlp.stanford.edu/software/relationExtractor.html

What kind of relations?

Most supervised method were evaluated on the 17 ACE relations:

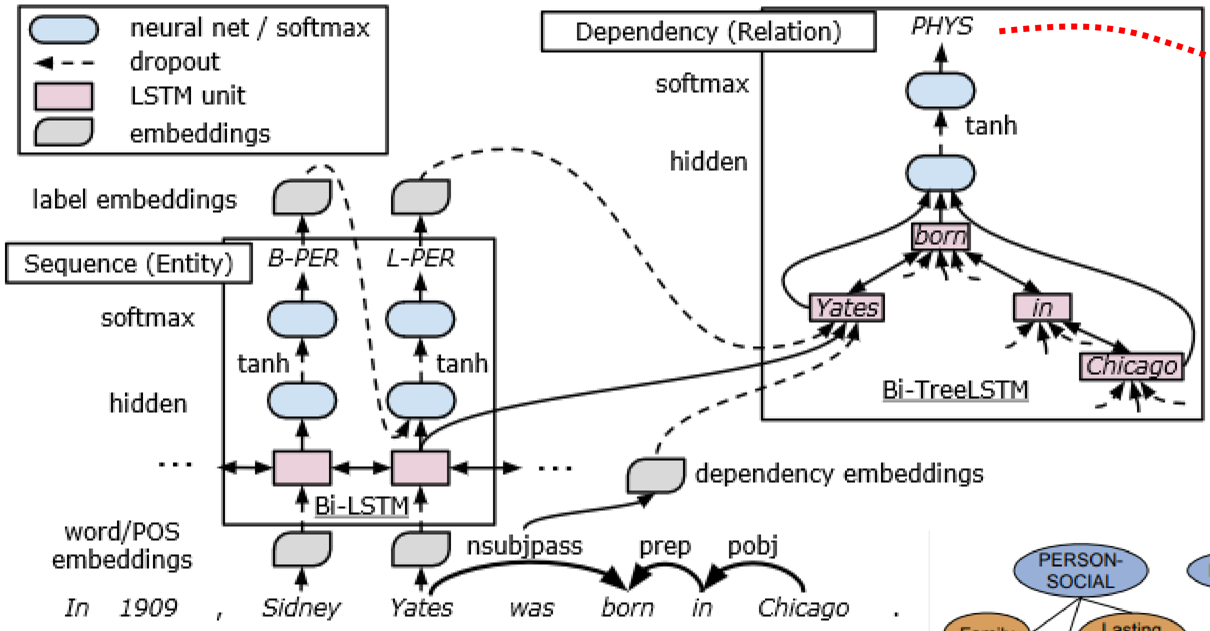


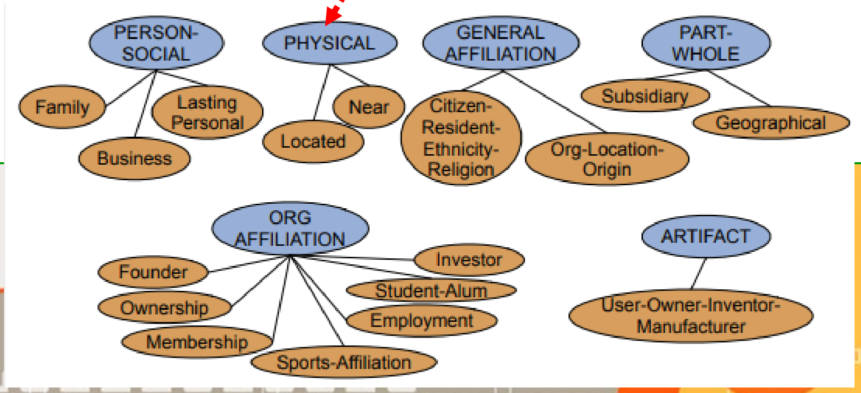
Neural Architectures for Relation Extraction

End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures (Miwa and Bansal 2016)

- https://www.aclweb.org/anthology/P16-1105

➢ *recurrent neural network based model that captures both word sequence and dependency tree substructure information by stacking bidirectional tree-structured LSTM-RNNs on bidirectional sequential LSTM-RNNs.*





Distant Supervision for Relation Extraction

- text that has been hand-labeled with relation labels is extremely expensive to produce!
- *there are ways to find indirect sources of training data!!!*

- *The distant supervision method of Mintz et al. (2009) combines the advantages of <u>bootstrapping</u> with supervised learning.*

IDEEA: *Instead of just a handful of seeds, distant supervision:*

1. *uses a large database to acquire a <u>huge number of seed examples</u>,*
2. *creates <u>lots of noisy pattern features</u> from all these examples and*
3. *then combines them in a <u>supervised classifier</u>.*

Walk-through example

- *we are trying to learn the place-of-birth relationship between people and their birth cities.*
- *In the seed-based approach, we might have only 5 examples to start with!!!*

But Wikipedia-based databases like DBPedia or Freebase have tens of thousands of examples of many relations; including over <u>100,000 examples of place-of-birth</u>.

The next step is to <u>run named entity taggers</u> on large amounts of text— Mintz et al. (2009) used 800,000 articles from Wikipedia— and extract all sentences that have two named entities that match the tuple, like the following:

...Hubble was born in Marshfield...
...Einstein, born (1879), Ulm...
...Hubble's birthplace in Marshfield...

Walk-through example 2

Now we have a large set of sentence examples:

...Hubble was born in Marshfield...
...Einstein, born (1879), Ulm...
...Hubble's birthplace in Marshfield...

Training instances can now be extracted from this data, one training instance for each identical tuple . Thus there will be one training instance for each of:

```
<born-in, Edwin Hubble, Marshfield>
<born-in, Albert Einstein, Ulm>
<born-year, Albert Einstein, 1879>
```

We can then apply feature-based or neural classification.

For feature-based classification, standard supervised relation extraction features like the named entity labels of the two mentions, the words and dependency paths in between the mentions, and neighboring words.

- Each tuple will have features collected from many training instances; the feature vector for a single training instance like <born-in, Albert Einstein, Ulm> will have lexical and syntactic features from many different sentences that mention Einstein and Ulm!!!!

Features in Distant Learning

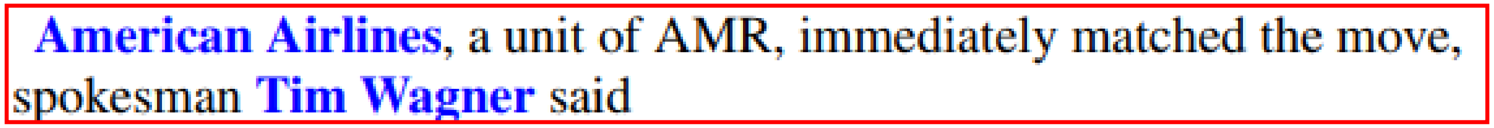
- ❑ Note: *Because distant supervision has* *very large training sets**, it is also able to use* ***very rich features*** *that are conjunctions of these individual features.*

- ➢ *So we will extract thousands of patterns that conjoin the entity types with the intervening words or dependency paths like these:*

  PER was born in LOC

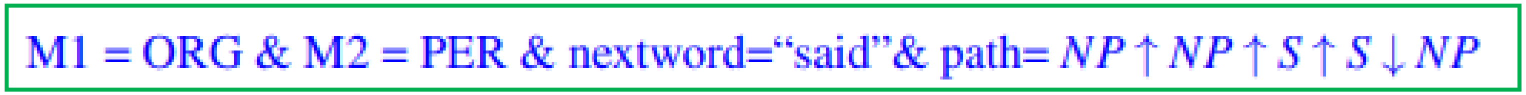
  PER's birthplace in LOC

or for the relation example:



American Airlines, a unit of AMR, immediately matched the move, spokesman **Tim Wagner** said

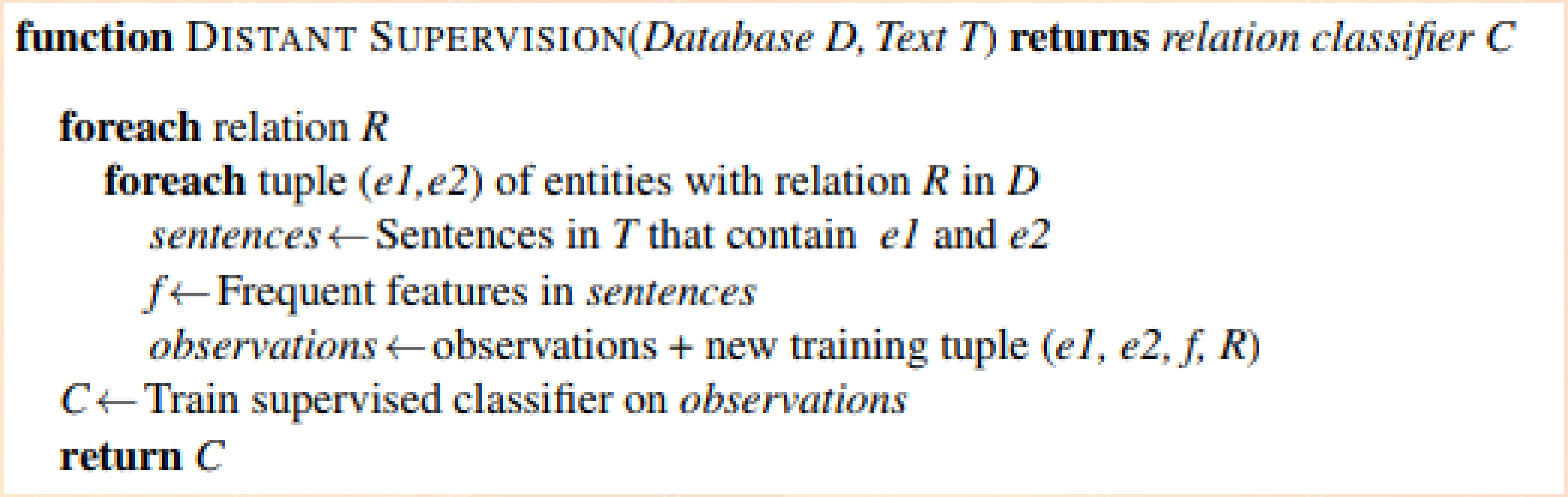
we would learn rich conjunction features like this one:



M1 = ORG & M2 = PER & nextword="said"& path= $NP \uparrow NP \uparrow S \uparrow S \downarrow NP$

❑ Note: *Since not every test sentence will have one of the training relations, the classifier will also need to be able to label an example as* ***no-relation****. This label is trained by randomly selecting entity pairs that do not appear in any Freebase relation, extracting features for them, and building a feature vector for each such tuple.*

The distant supervision algorithm for relation extraction



```
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
```

The Neural classifier might not need to use the feature set f !!!!

Unsupervised Relation Extraction

- *The goal of unsupervised relation extraction is to extract relations when we have no labeled training data*
- *When we do not even any list of relations – it is OPEN-IE*
  - *the relations open information extraction are simply strings of words (usually beginning with a verb).*

EXAMPLE: the ReVerb system (Fader et al., 2011) extracts a relation from a sentence *s* in 4 steps:

1. Run a part-of-speech tagger and entity chunker over *s*
2. 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.
3. For each phrase *w*, find the nearest noun phrase *x* to the left which is not a relative pronoun, wh-word or existential “there”. Find the nearest noun phrase *y* to the right.
4. Assign confidence **c** to the relation *r = (x,w, y)* using a confidence classifier and return it.

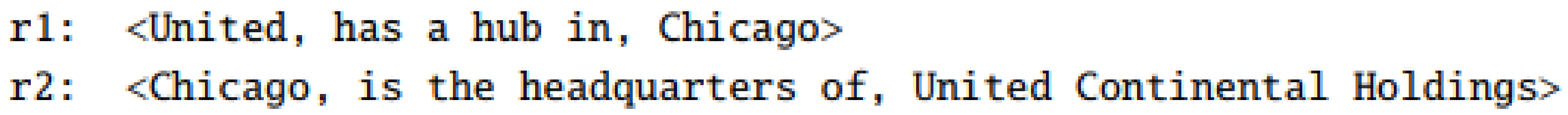
Example of ReVerb processing:

- *Given the sentence:*

United has a hub in Chicago, which is the headquarters of United Continental Holdings.

- *It has 2 relation phrases "*has a hub in*" and "*is the headquarters of*"*

*Step 3 of ReVerb will find "*United*" to the left and "*Chicago*" to the right of "*has a hub in*", and skips over "*which*" to find "*Chicago*" to the left of "*is the headquarters of*". The final output is:*



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

Extracting Time

- *Times and dates are a particularly important <u>kind of named entity </u>that play a role in question answering, in calendar and personal assistant applications.*

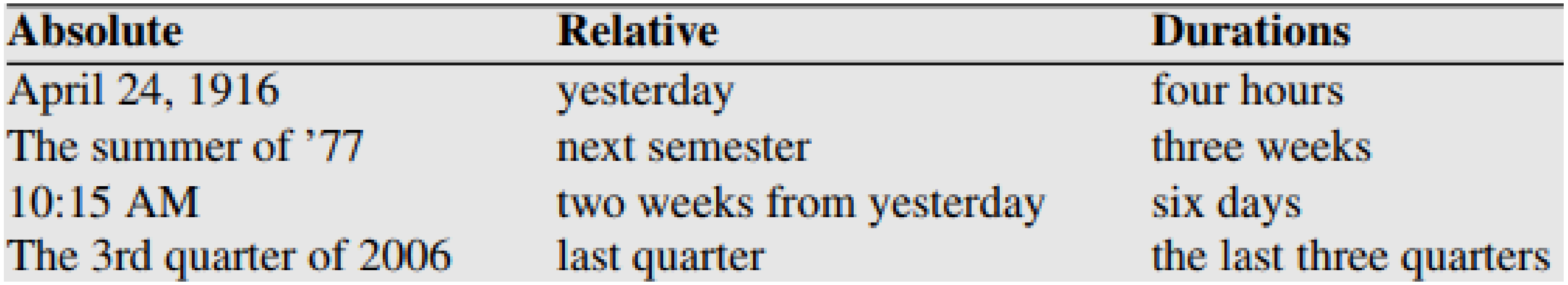
*In order to reason about times and dates, after we **extract** these temporal expressions they **must be normalized**—converted to a standard format so we can reason about them.*

Temporal Expression Extraction

- Temporal expressions are those that refer to:
  - absolute points in time,
  - relative times,
  - absolute durations,
  - and sets of these.

- Absolute temporal expressions are those that can be mapped directly to *calendar dates*, *times of day*, *or both*.
- Relative temporal expressions map to particular times through some *other reference point* (as in a week from last Tuesday).
- Durations denote spans of time at varying levels of granularity (seconds, minutes, days, weeks, centuries, etc.).

Examples of absolute, relational and durational temporal expressions.



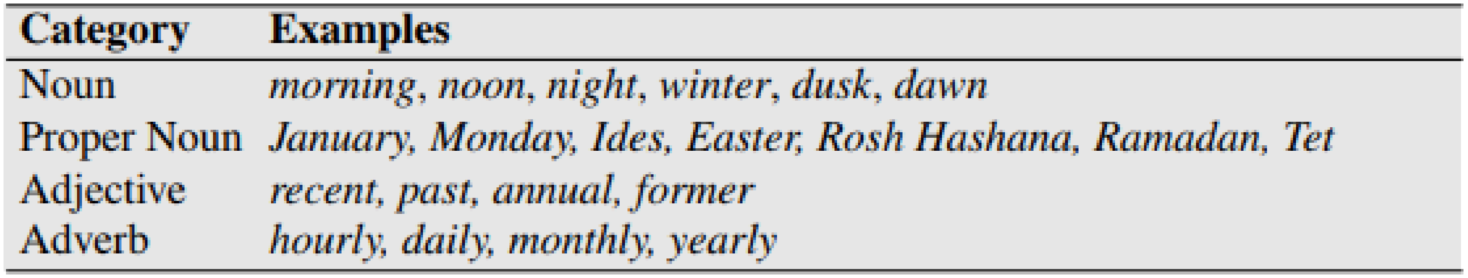
Absolute	Relative	Durations
April 24, 1916	yesterday	four hours
The summer of '77	next semester	three weeks
10:15 AM	two weeks from yesterday	six days
The 3rd quarter of 2006	last quarter	the last three quarters

- *Important Observation:* Temporal expressions are grammatical constructions that have **temporal lexical triggers** as their heads.

Lexical triggers might be nouns, proper nouns, adjectives, and adverbs;

Full temporal expressions consist of their (lexical triggers) phrasal projections: noun phrases, adjective phrases, and adverbial phrases.

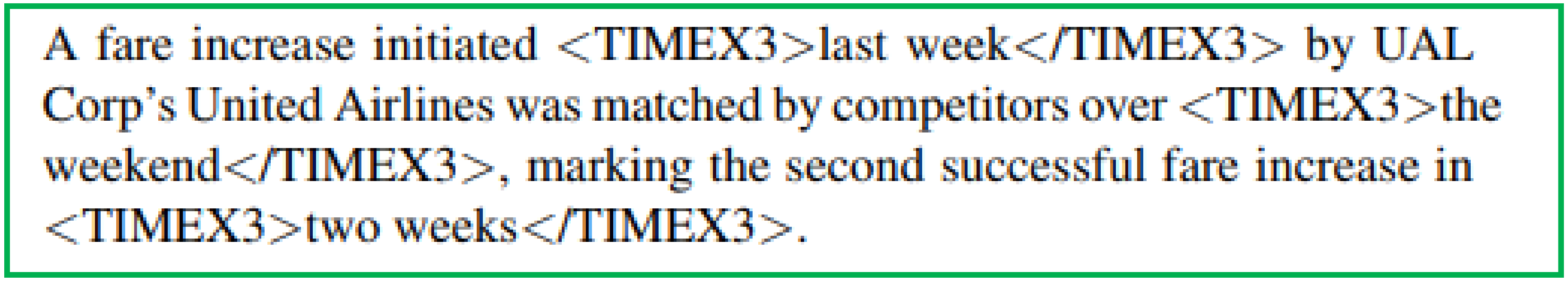
Examples of lexical triggers:



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*

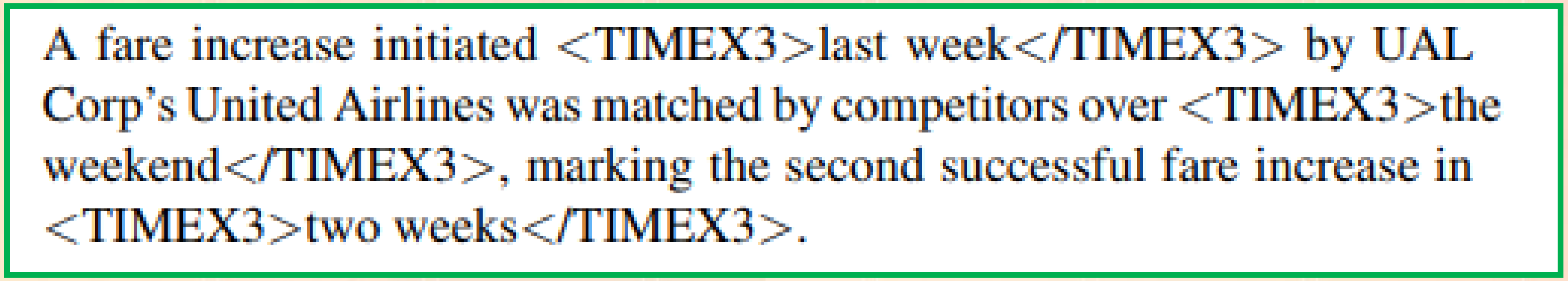
The TimeML annotation scheme

- *The TimeML annotation scheme annotates temporal expressions with an XML tag, TIMEX3, and various attributes to that tag (Pustejovsky et al. 2005, Ferro et al. 2005).*


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

The temporal expression recognition task

- *The temporal expression recognition task consists of finding the start and end of all of the text spans that correspond to such temporal expressions.*



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

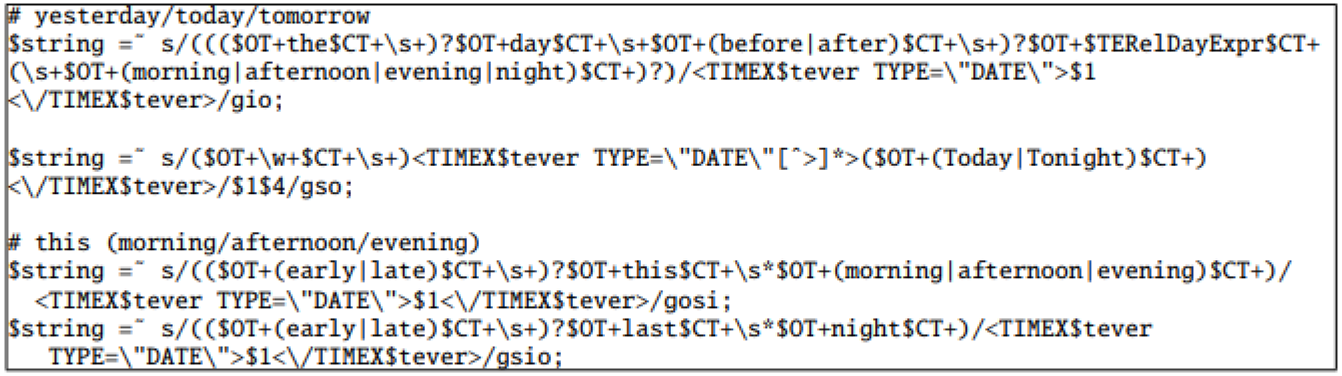
- *Rule-based approaches*
- *Sequence-labeling approaches*

Rule-based Approaches to Temporal Expression Extraction

Rule-based approaches to temporal expression recognition use cascades of automata to recognize ***patterns*** *at increasing levels of complexity.*

- *Tokens are first part-of-speech tagged, and then larger and larger chunks are recognized from the results from previous stages, based on <u>patterns containing trigger words</u> (e.g., February) or classes (e.g., MONTH).*

Perl fragment from the GUTime temporal tagging system in Tarsqi (Verhagen et al., 2005).



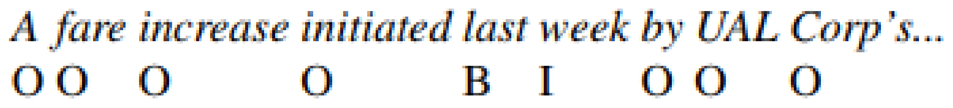
```
# yesterday/today/tomorrow
$string =~ s/((($OT+the$CT+\s+)?$OT+day$CT+\s+$OT+(before|after)$CT+\s+)?$OT+$TERelDayExpr$CT+
(\s+$OT+(morning|afternoon|evening|night)$CT+)?)/<TIMEX$tever TYPE=\"DATE\">$1
<\/TIMEX$tever>/gio;

$string =~ s/($OT+\w+$CT+\s+)<TIMEX$tever TYPE=\"DATE\"[^>]*>($OT+(Today|Tonight)$CT+)
<\/TIMEX$tever>/$1$4/gso;

# this (morning/afternoon/evening)
$string =~ s/(($OT+(early|late)$CT+\s+)?$OT+this$CT+\s*$OT+(morning|afternoon|evening)$CT+)/
  <TIMEX$tever TYPE=\"DATE\">$1<\/TIMEX$tever>/gosi;
$string =~ s/(($OT+(early|late)$CT+\s+)?$OT+last$CT+\s*$OT+night$CT+)/<TIMEX$tever
   TYPE=\"DATE\">$1<\/TIMEX$tever>/gsio;
```

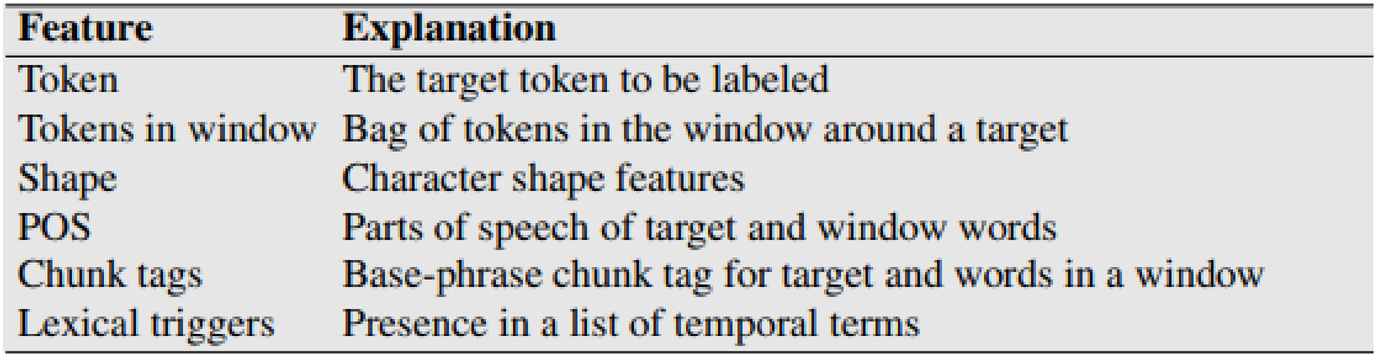
Sequence-labeling approaches to TIMEX

Sequence-labeling approaches follow the same IOB scheme used for named entity tags, marking words that are either inside, outside or at the beginning of a TIMEX3-delimited temporal expression with the I, O, and B tags as follows:



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

Features are extracted from the token and its context, and a statistical sequence labeler is trained! Typical features are:



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

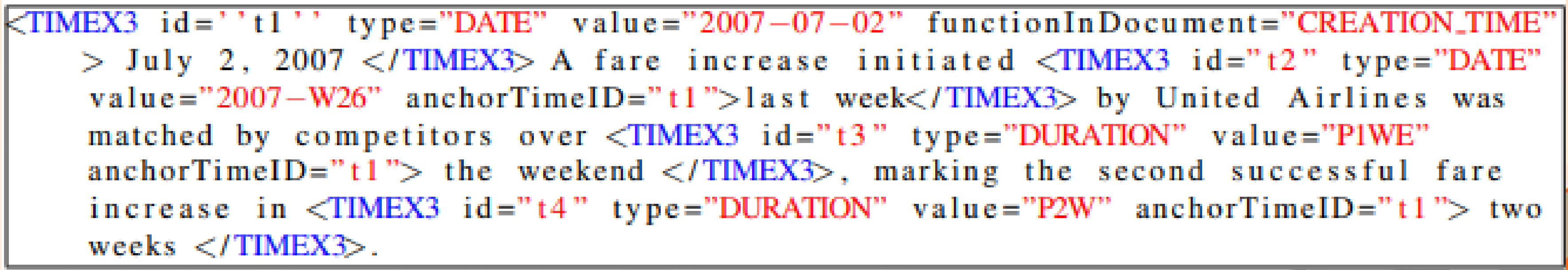
Temporal Normalization

- *Temporal normalization is the process of* ***mapping*** *a temporal expression to either:*
- *temporal normalization of a specific point in time or to*
- *a duration.*

Points in time *correspond to calendar dates, to times of day, or both.*

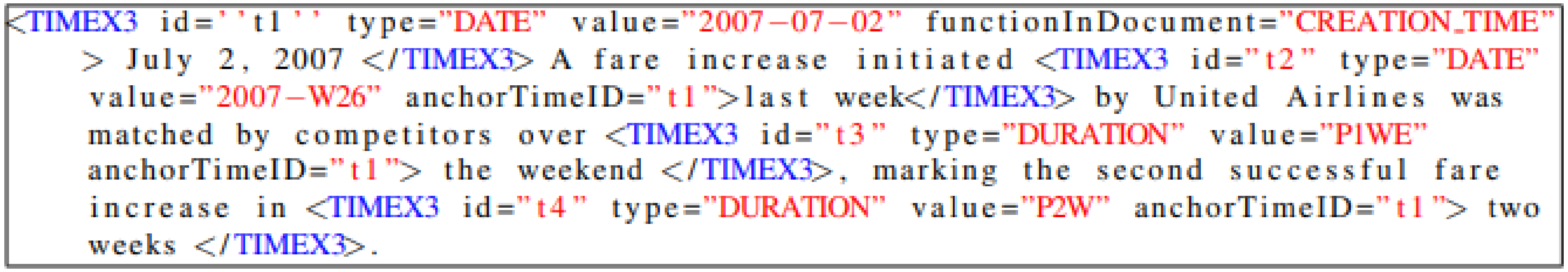
Durations primarily consist of lengths of time but may also include information about start and end points. Normalized times are represented with the VALUE attribute from the ISO 8601 standard for encoding temporal values (ISO8601, 2004).

Example:



```
<TIMEX3 id=''t1'' type="DATE" value="2007-07-02" functionInDocument="CREATION_TIME"
    > July 2, 2007 </TIMEX3> A fare increase initiated <TIMEX3 id="t2" type="DATE"
    value="2007-W26" anchorTimeID="t1">last week</TIMEX3> by United Airlines was
    matched by competitors over <TIMEX3 id="t3" type="DURATION" value="P1WE"
    anchorTimeID="t1"> the weekend </TIMEX3>, marking the second successful fare
    increase in <TIMEX3 id="t4" type="DURATION" value="P2W" anchorTimeID="t1"> two
    weeks </TIMEX3>.
```

Explaining the Normalization



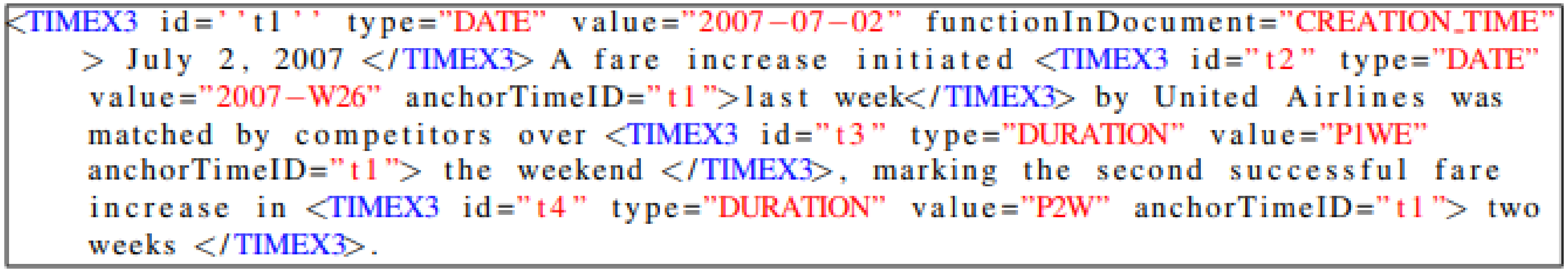
```
<TIMEX3 id=''t1'' type="DATE" value="2007-07-02" functionInDocument="CREATION_TIME"
    > July 2, 2007 </TIMEX3> A fare increase initiated <TIMEX3 id="t2" type="DATE"
    value="2007-W26" anchorTimeID="t1">last week</TIMEX3> by United Airlines was
    matched by competitors over <TIMEX3 id="t3" type="DURATION" value="P1WE"
    anchorTimeID="t1"> the weekend </TIMEX3>, marking the second successful fare
    increase in <TIMEX3 id="t4" type="DURATION" value="P2W" anchorTimeID="t1"> two
    weeks </TIMEX3>.
```

Temporal expression t_1 is the dateline, or document date, for this text, which was July 2, 2007. The ISO representation for this kind of expression is YYYY-MM-DD, or in this case, 2007-07-02

Temporal expression t_2 is a date, "last week", referring to a particular week of the year. In the ISO standard, weeks are numbered from 01 to 53, with the first week of the year being the one that has the first Thursday of the year. These weeks are represented with the template YYYY-Wnn. The ISO week for our document date is week 27; thus the value for last week is represented as "2007-W26".

Temporal expression t_3 is a duration, "the weekend". ISO weeks begin on Monday; thus, weekends occur at the end of a week and are fully contained within a single week. Weekends are treated as durations, so the value of the VALUE attribute has to be a ***length****. Durations are represented according to the pattern Pnx, where n is an integer denoting the length and x represents the unit, as in P3Y for three years or P2D for two days. In this example, one weekend is captured as P1WE. In this case, there is also sufficient information to anchor this particular weekend as part of a particular week. Such information is encoded in the ANCHORTIMEID attribute.*

Explaining the Normalization

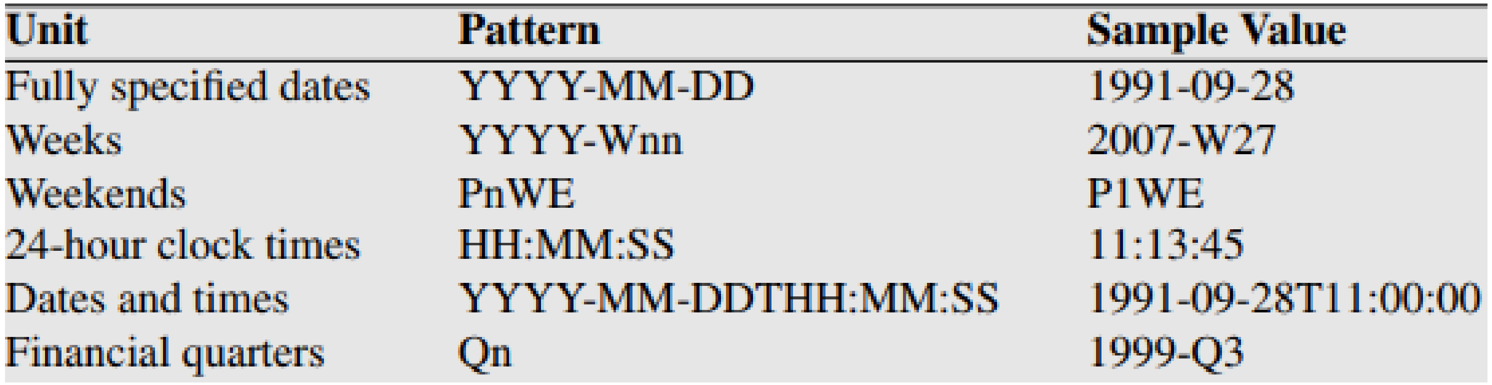


```
<TIMEX3 id=''t1'' type="DATE" value="2007-07-02" functionInDocument="CREATION_TIME"
   > July 2, 2007 </TIMEX3> A fare increase initiated <TIMEX3 id="t2" type="DATE"
   value="2007-W26" anchorTimeID="t1">last week</TIMEX3> by United Airlines was
   matched by competitors over <TIMEX3 id="t3" type="DURATION" value="P1WE"
   anchorTimeID="t1"> the weekend </TIMEX3>, marking the second successful fare
   increase in <TIMEX3 id="t4" type="DURATION" value="P2W" anchorTimeID="t1"> two
   weeks </TIMEX3>.
```

Temporal expression t_4 is the phrase "two weeks" also denotes <u>a duration</u> captured as P2W

- ➢ *Consult ISO8601 (2004), Ferro et al. (2005), and Pustejovsky et al. (2005) for more details.*

- ❑ Sample ISO patterns for representing various times and durations:

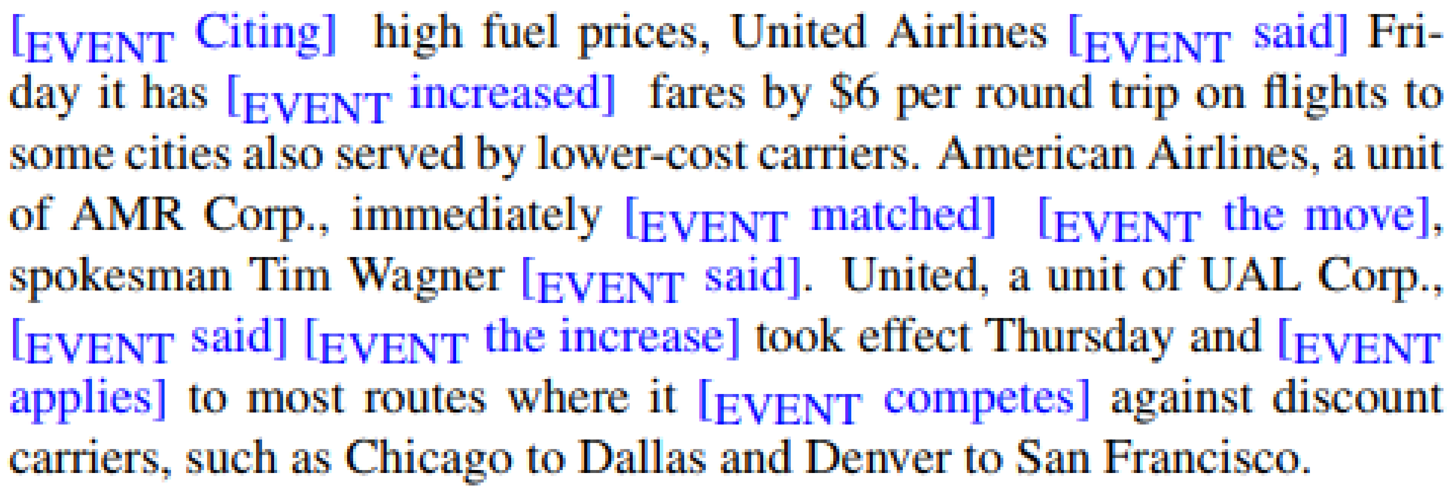


Unit	**Pattern**	**Sample Value**
Fully specified dates	YYYY-MM-DD	1991-09-28
Weeks	YYYY-Wnn	2007-W27
Weekends	PnWE	P1WE
24-hour clock times	HH:MM:SS	11:13:45
Dates and times	YYYY-MM-DDTHH:MM:SS	1991-09-28T11:00:00
Financial quarters	Qn	1999-Q3

- ❑ *https://nlp.stanford.edu/software/sutime.html*

Extracting Events and their Times

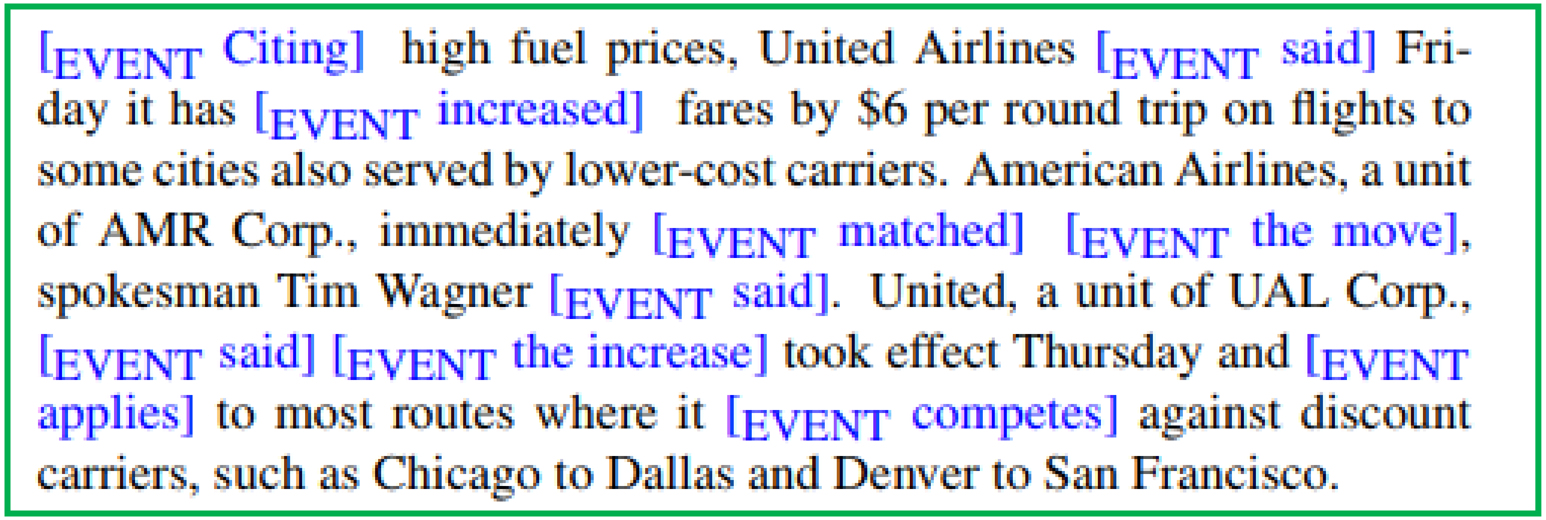
- *The task of event extraction is to identify mentions of events in texts. For the event extraction purposes of this task, an event mention is* ***any expression denoting an event or state*** *that can be assigned to a particular point, or interval, in time.*



[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.

Event Extraction

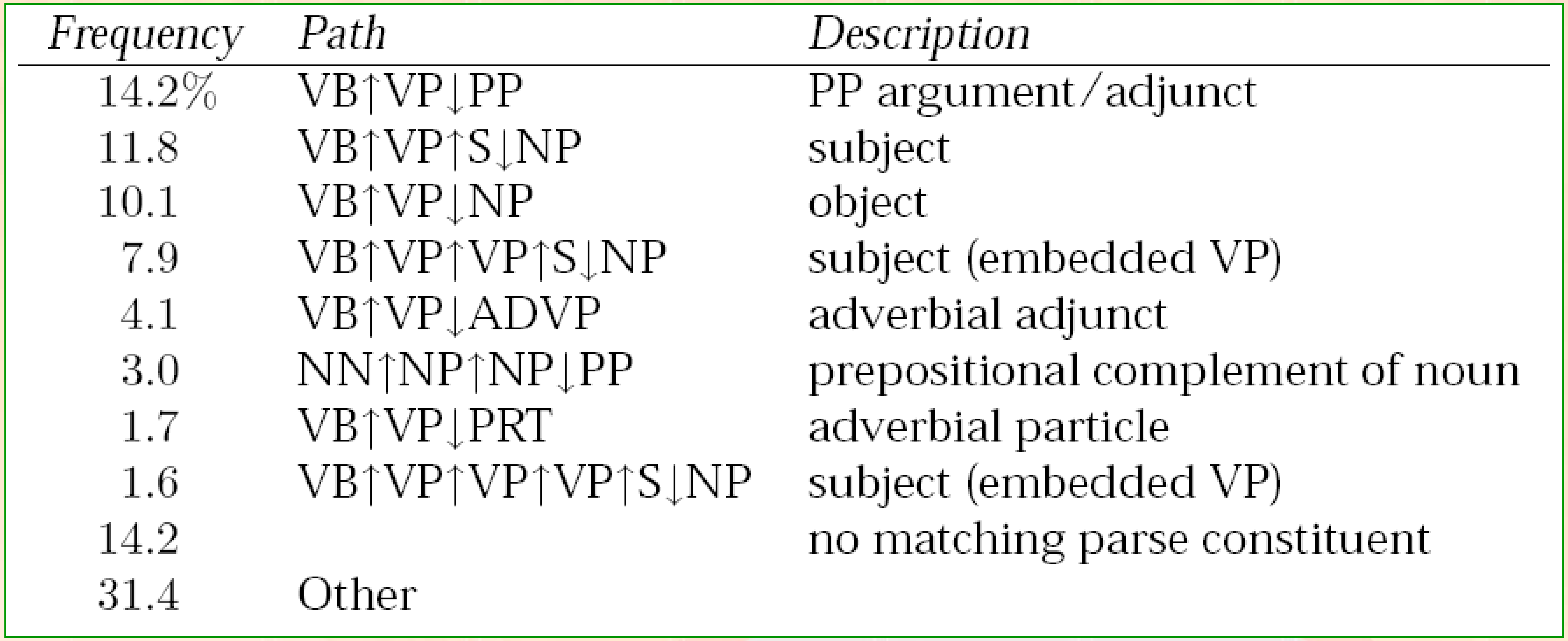
- *In English, <u>most</u> event mentions correspond to **verbs**, and most verbs introduce events.*



[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.

- However, as we can see from our example, this is not always the case! Events can be introduced by **noun phrases**, as in the move and the increase, and <u>some verbs fail to introduce events</u>, as in the phrasal verb took effect, which refers to when the event began rather than to the event itself. Similarly, **<u>light verbs</u>** such as make, take, and have often fail to denote events; for light verbs the event is often expressed by the nominal direct object (took a flight), and these light verbs just provide a syntactic structure for the noun's arguments.

Most Frequent Values of the Parse Tree Path Feature



Frequency	*Path*	*Description*
14.2%	VB↑VP↓PP	PP argument/adjunct
11.8	VB↑VP↑S↓NP	subject
10.1	VB↑VP↓NP	object
7.9	VB↑VP↑VP↑S↓NP	subject (embedded VP)
4.1	VB↑VP↓ADVP	adverbial adjunct
3.0	NN↑NP↑NP↓PP	prepositional complement of noun
1.7	VB↑VP↓PRT	adverbial particle
1.6	VB↑VP↑VP↑VP↑S↓NP	subject (embedded VP)
14.2		no matching parse constituent
31.4	Other	

Various Versions of Event Extraction

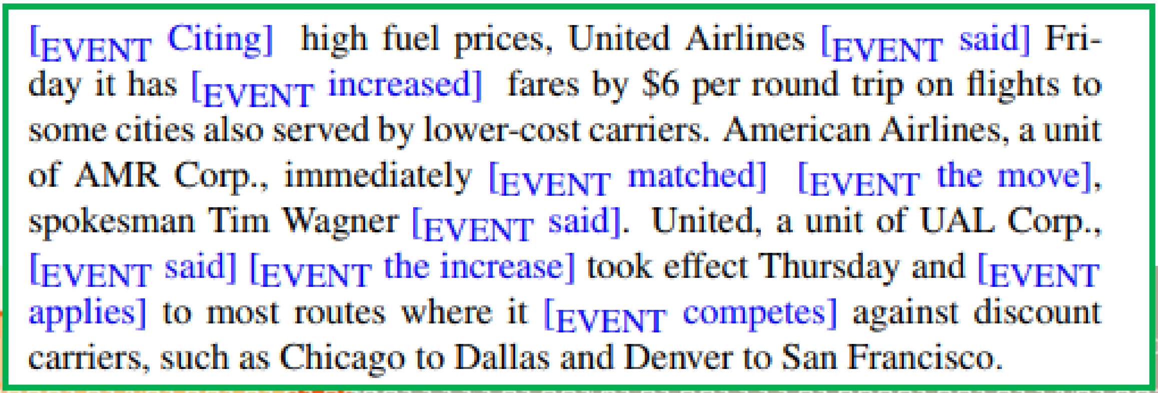
- *Various versions of the event extraction task exist, depending on the goal.*

- *For example in the **TempEval shared tasks** (Verhagen et al. 2009) the goal is <u>to extract events and aspects like their aspectual and temporal properties</u>.*

Events are to be classified as:

- *actions,*
- *states,*
- *reporting events (say, report, tell, explain),*
- *perception events, and so on.*

The aspect, tense, and modality of each event also needs to be extracted. Thus for example the various said events in the sample text would be annotated as (class=REPORTING, tense=PAST, aspect=PERFECTIVE).



[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.

Event Classes in TimeML

- *Events can be punctual or last for a period of time. TimeML also considers as events those predicates describing <u>states</u> or <u>circumstances</u> in which something obtains or holds true. Not all stative predicates are marked up, however, as only those states which participate in an opposition structure in a given text are marked up.*
- *Events are generally expressed by means of tensed or untensed verbs, nominalizations, adjectives, predicative clauses, or prepositional phrases.*

The specification of EVENT classes in TimeML is shown below:

attributes ::= eid class tense aspect

eid ::= ID {eid ::= EventID EventID ::= e}

class ::= 'OCCURRENCE' | 'PERCEPTION' | 'REPORTING' | 'ASPECTUAL' | 'STATE' | 'I_STATE' | 'I_ACTION' | 'MODAL'

tense ::= 'PAST' | 'PRESENT' | 'FUTURE' | 'NONE' aspect ::= 'PROGRESSIVE' | 'PERFECTIVE' | 'PERFECTIVE_PROGRESSIVE' | 'NONE'

Event Classes in TimeML

Examples of each of event types in TimeML are given below:

*1. **Occurrence**: die, crash, build, merge, sell*

*2. **State**: on board, kidnapped, love, ..*

*3. **Reporting**: say, report, announce, 4. I-Action: attempt, try, promise, offer*

*5. **I-State**: believe, intend, want*

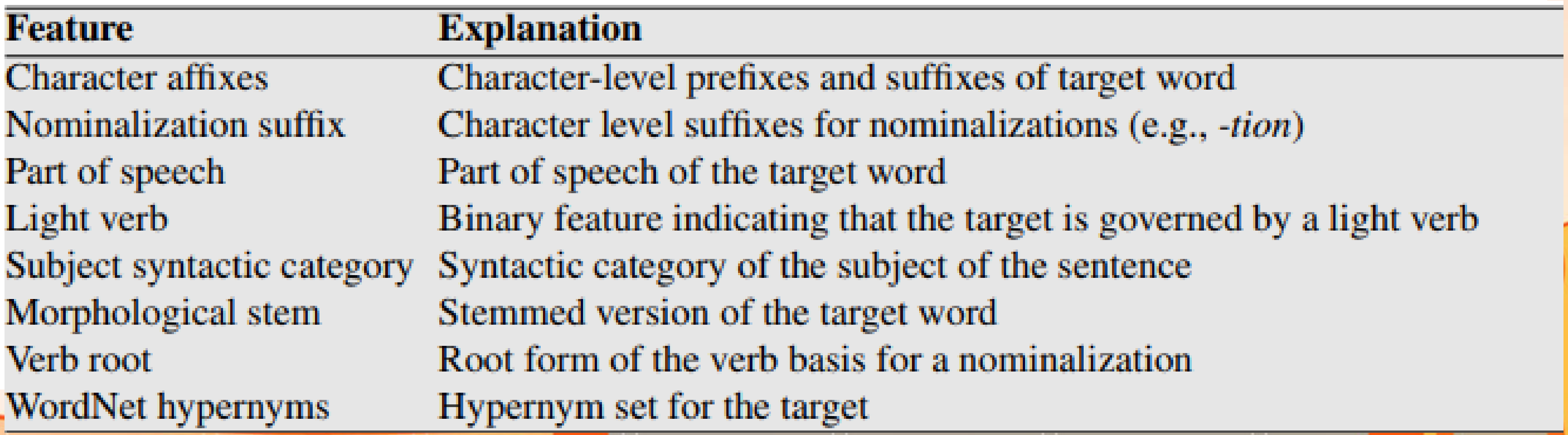
*6. **Aspectual**: begin, finish, stop, continue.*

*7. **Perception:** See, hear, watch, feel*

Modeling Event Extraction

- *Event extraction is generally modeled via <u>supervised learning</u>,*
  - *detecting events via* ***sequence models with IOB tagging****, and*
  - *assigning event classes and attributes with* ***multi-class classifiers.***

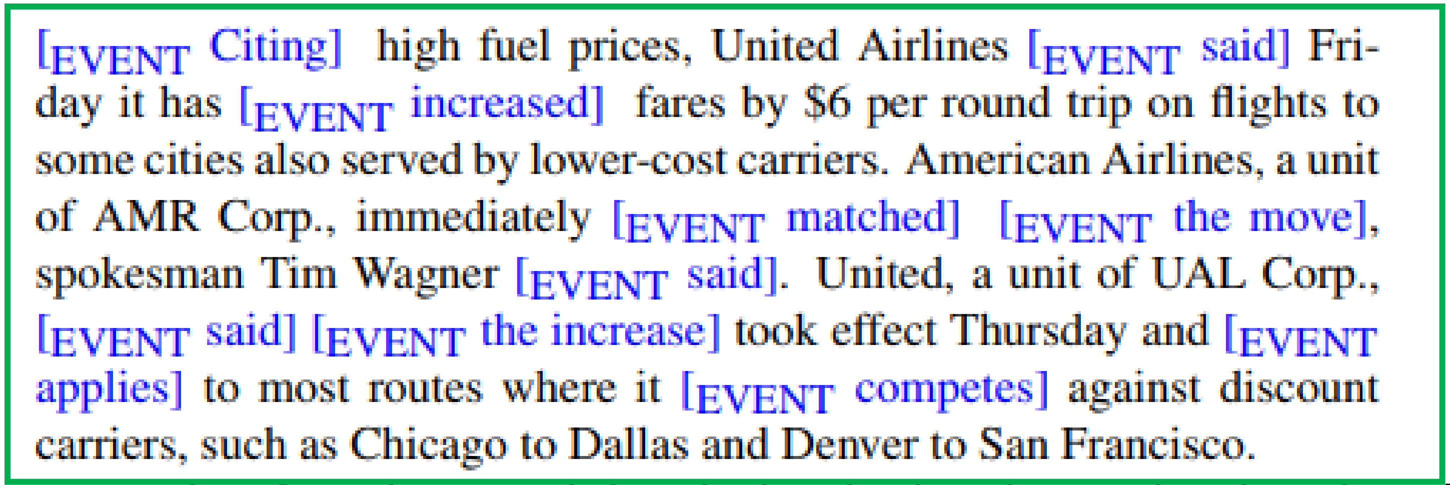
  Common features include surface information like parts of speech, lexical items, and verb tense information;



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

- *With both the events and the temporal expressions in a text having been detected, the next logical task is to use this information to fit the events into a complete timeline.*
  - *Such a timeline would be useful for applications such as question answering and summarization.*
- *A somewhat simpler, but still useful, task is to impose a partial ordering on the events and temporal expressions mentioned in a text. Such an ordering can provide many of the same benefits as a true timeline.*



[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.

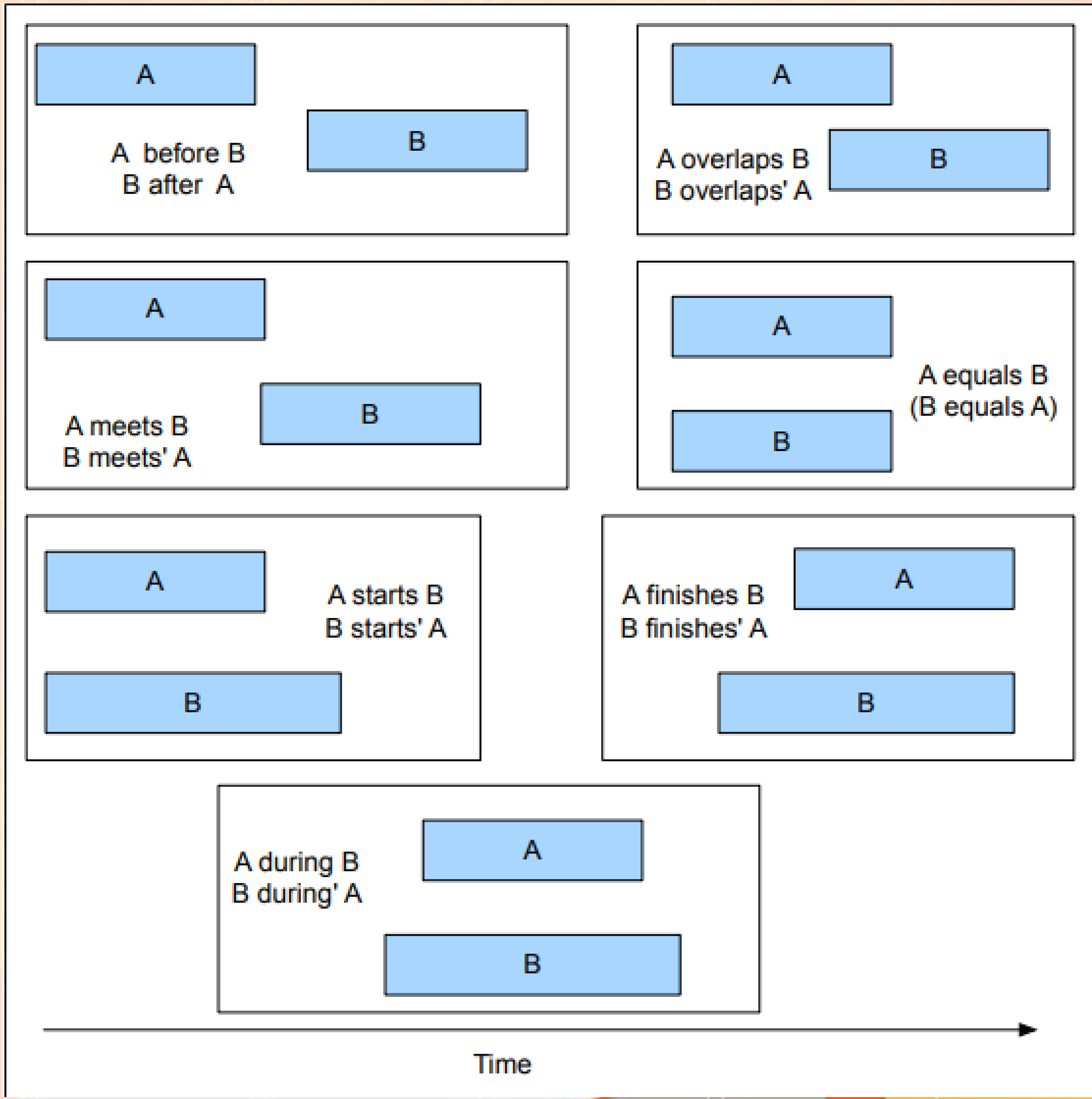
- *An example of such a partial ordering is the determination that the fare increase by American Airlines came after the fare increase by United in our sample text*

Allen's Temporal Relations

The 13 temporal relations from Allen (1984).

The temporal relations between events can be classified into one of the Allen standard set of relations!!!

How??? *using feature-based classifiers as in trained on* ***the TimeBank corpus*** *with features like words/embeddings, parse paths, tense and aspect.*



The TimeBank Corpus

- *The TimeBank corpus consists of 183 news articles selected from a variety of sources, including the Penn TreeBank and PropBank collections.*

- *Each article in the TimeBank corpus has had the temporal expressions and event mentions in them explicitly annotated in the TimeML annotation (Pustejovsky et al., 2003). In addition to temporal expressions and events, the TimeML annotation provides temporal links between events and temporal expressions that specify the nature of the relation between them.*

Example of TimeBank annotations

Given the sentence:

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

- *The TimeBank annotations are:*

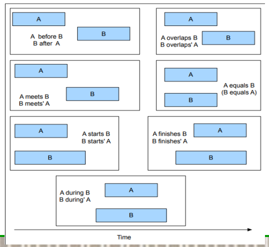
```
<TIMEX3 tid="t57" type="DATE" value="1989-10-26"  functionInDocument="CREATION_TIME">
10/26/89  </TIMEX3>

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.
```

➢ *this text includes three events (e$_1$, e$_3$ and e$_4$) and two temporal expressions (t$_{57}$ and t$_{58}$) . The events are all in the occurrence class and are given unique identifiers for use in further annotations. The temporal expressions include the creation time of the article, which serves as the document time, and a single temporal expression within the text.*

➢ *There are within-sentence temporal relations annotated for this example.*

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

Many texts contain reports of events, and possibly sequences of events, that often correspond to fairly common, stereotypical situations in the world. These abstract scripts situations or stories, related to what have been called ***scripts*** *(Schank and Abelson, 1977), consist of prototypical sequences of sub-events, participants, and their roles.*

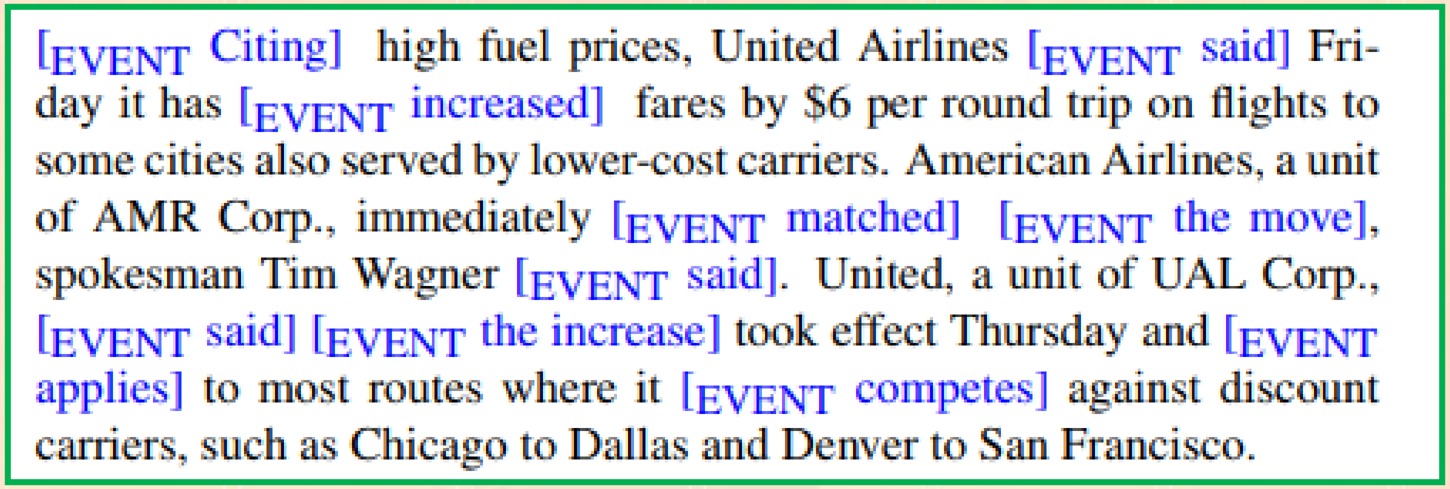
The strong expectations provided by these scripts can facilitate the proper classification of entities, the assignment of entities into roles and relations, and most critically, the drawing of inferences that fill in things that have been left unsaid.

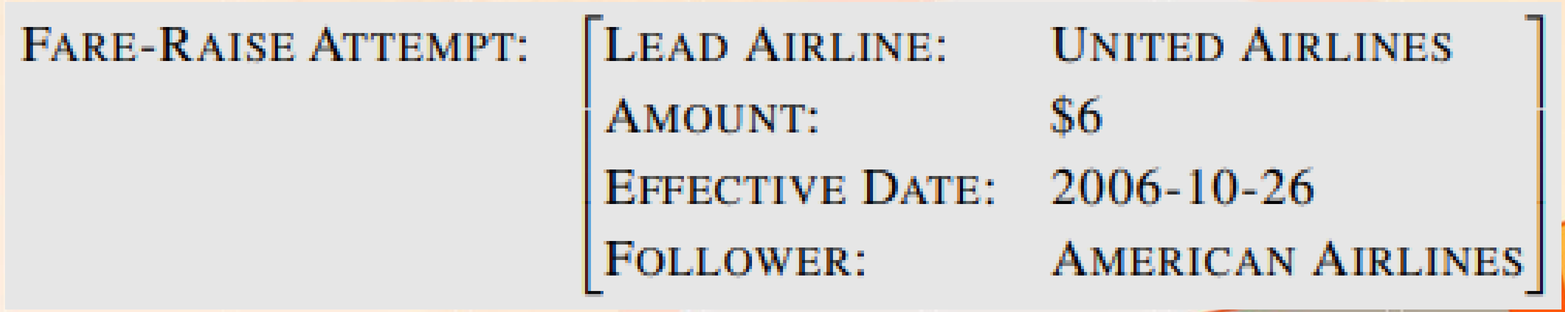
In their simplest form, such scripts can be represented as templates consisting of ***fixed sets of slots*** *that take as values* ***slot-fillers*** *belonging to particular classes.*

- *The task template filling of template filling is to find documents that invoke particular scripts and then fill the slots in the associated templates with fillers extracted from the text. These slot-fillers may consist of text segments extracted directly from the text, or they may consist of concepts that have been inferred from text elements through some additional processing.*

Example of Template Filling

- *From the text:*


[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.



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

Machine Learning Approaches to Template Filling

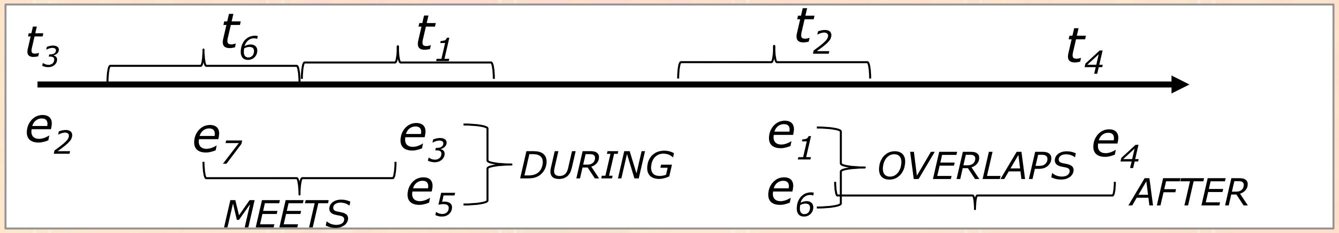
The task is generally modeled by training two separate supervised systems.

- *The first system decides whether the template is present in a particular sentence. This task is called template recognition. Template recognition can be treated as a text classification task, with features extracted from every sequence of words that was labeled in training documents as filling any slot from the template being detected. The usual set of features can be used: tokens, embeddings, word shapes, part-of-speech tags, syntactic chunk tags, and named entity tags.*
- *The second system has the job of role-filler extraction. A separate classifier is trained to detect each role (LEAD-AIRLINE, AMOUNT, and so on). This can be a binary classifier that is run on every noun-phrase in the parsed input sentence, or a sequence model run over sequences of words. Each role classifier is trained on the labeled data in the training set. Again, the usual set of features can be used, but now trained only on an individual noun phrase or the fillers of a single slot.*

➢ *Problem: Multiple non-identical text segments might be labeled with the same slot label. Need to perform coreference resolution.*

Information Extraction - Conclusions

- *We explored techniques for extracting information from texts.*
- *Named entities can be recognized and classified by featured-based or neural sequence labeling techniques.*
- *Relations among entities can be extracted by pattern-based approaches, supervised learning methods when annotated training data is available, lightly supervised bootstrapping methods when small numbers of seed tuples or seed patterns are available, distant supervision when a database of relations is available, and unsupervised or Open IE methods.*
- *Reasoning about time can be facilitated by detection and normalization of temporal expressions through a combination of statistical learning and rule-based methods.*
- *Events can be detected and ordered in time using sequence models and classifiers trained on temporally- and event-labeled data like the TimeBank corpus.*
- *Template-filling applications can recognize stereotypical situations in texts and assign elements from the text to roles represented as fixed sets of slots.*


t_3
t_6
t_1
t_2
t_4
e_2
e_7
e_3
e_5
DURING
MEETS
e_1
e_6
OVERLAPS
e_4
AFTER