CMPE 297, INSTRUCTOR: JORJETA JETCHEVA

INFORMATION EXTRACTION

INFORMATION EXTRACTION OVERVIEW

- Information Extraction (IE) turns unstructured information from text (semantic content) into structured data that can be used to populate a database or knowledge graph, e.g.,
  - We can extract semantic relations, e.g., X is the *child-of* Y (we will cover these today)
  - Events and event coreference
  - Temporal expressions, e.g., dates, days of the week, temporal reference like "the last time"
- We can also fill in templates for stereotypical events by extracting the information from text, e.g., in the example below:

Citing high fuel prices, United Airlines said Friday it has 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 matched the move, spokesman Tim Wagner said. United, a unit of UAL Corp., said the increase took effect Thursday and applies to most routes where it 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

RELATION EXTRACTION OVERVIEW

- Typically we would extract named entities using an NER approach
- Then we would try to discover relationships, e.g., based on
  - General (known) relations
  - Relations defined for a particular domain
  - Detecting new relationships in the text

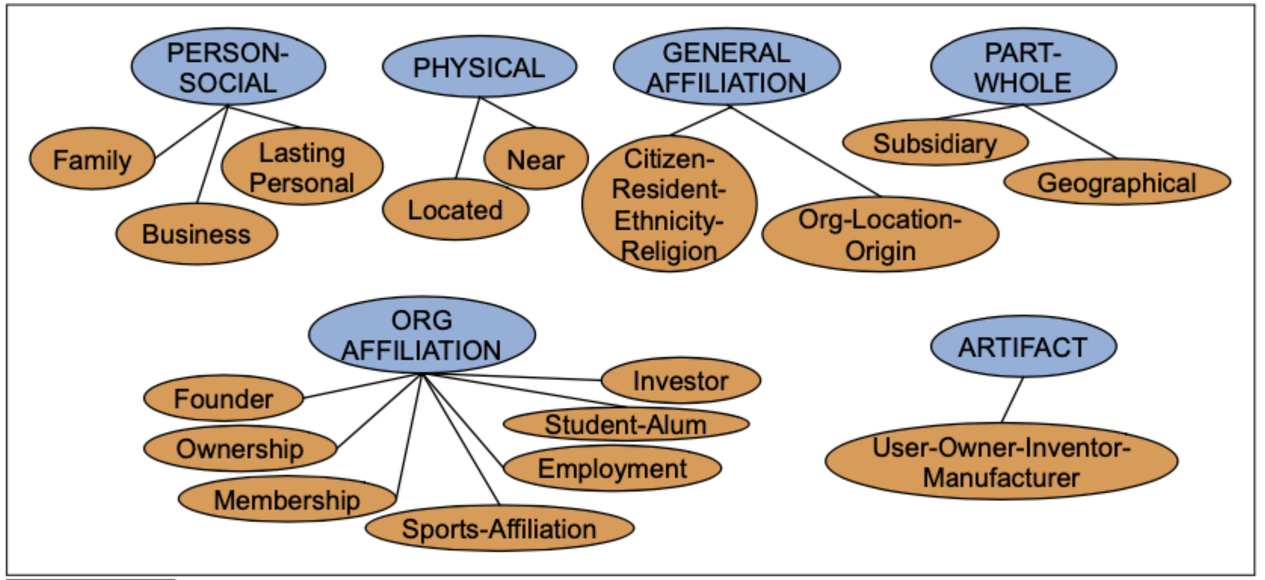


Figure 17.1 The 17 relations used in the ACE relation extraction task.

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**...

Figure 17.2 Semantic relations with examples and the named entity types they involve.

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

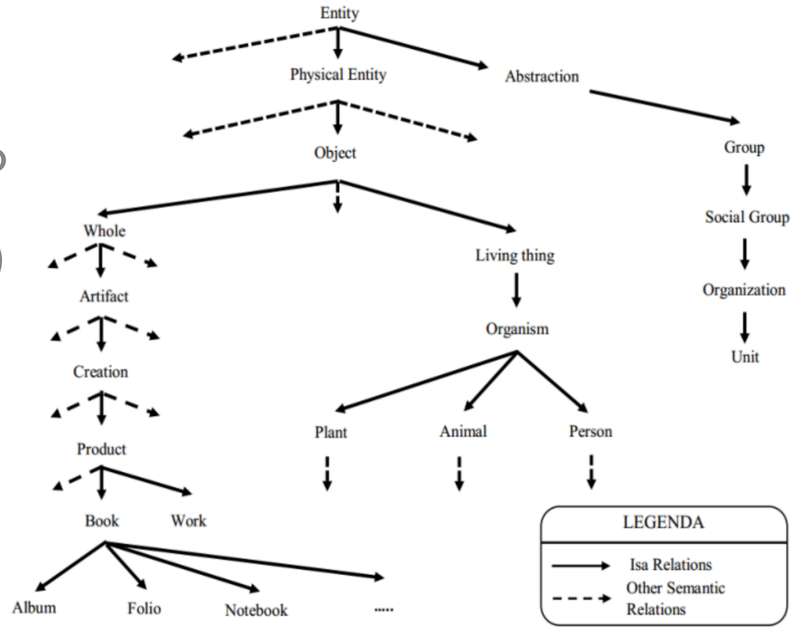
EXAMPLE SOURCES OF GENERIC RELATIONS

- Wikipedia
  - Infoboxes (tables associated with Wikipedia articles)
  - RDF triples (DBPedia contains over 2 billion RDF triples)

    Example RDF triple:

subject	predicate	object
Golden Gate Park	location	San Francisco

- Wordnet (lexical database used as a knowledge base)
  - Is-a relation
  - Instance-of relation



San José State University

Former name	Minns' Evening Normal School (1857–1862) California State Normal School (1862–1921) San Jose State Teachers College (1921–1935) San Jose State College (1935–1972) California State University, San Jose (1972–1974)
Motto	Powering Silicon Valley
Type	Public university
Established	1857; 165 years ago
Parent institution	California State University
Academic affiliations	Space-grant
Endowment	$197.1 million (2021)[1]
Budget	$405.2 million (2021)[2]
President	Stephen Perez (Interim)
Provost	Vincent Del Casino[3]

RELATION EXTRACTION ALGORITHMS

MAIN CLASSES OF RELATION EXTRACTION ALGORITHMS

- Hand-written patterns
- Supervised machine learning
- Semi-supervised machine learning via bootstrapping
- Semi-supervised machine learning via distant supervision
- Unsupervised

RELATION EXTRACTION USING HANDWRITTEN PATTERNS

RELATION EXTRACTION USING PATTERNS

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

Figure 17.5 Hand-built lexico-syntactic patterns for finding hypernyms, using {} to mark optionality (Hearst 1992a, Hearst 1998).

- Classical approach (Hearst 1992), is to extract lexico-syntactic patterns (or Hearst patterns) similarly to how humans do, e.g.,

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

  We can figure out that Gelidium is an example of red algae (red algae are a hypernym for Gelidium)

- This can be represented by the following lexicon-syntactic pattern: $NP_0 \text{ such as } NP_1\{,NP_2 \ldots, (and|or)NP_i\}, i \geq 1$

  implies the following semantics

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

  allowing us to infer

  $$\text{hyponym}(\text{Gelidium}, \text{red algae})$$

- More recently, patterns have been extended with NER constraints:

  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

- Patterns work very well but are difficult to define (manual process) and thus tend to cover only a limited number of the possible relations that might exist in text

RELATION EXTRACTION VIA SUPERVISED LEARNING

RELATION EXTRACTION VIA SUPERVISED LEARNING: OVERVIEW

Given an annotated corpus:

- Find pairs of named entities (typically in the same sentence)
- Intermediate step is often used where we determine whether there is a relation at all or not
  - Typically we generate positive examples based on an annotated corpus
  - Negative examples can be generated from each sentence between pairs of entities that are not annotated with a relation
- Apply a relation classification on each pair: identify specific relationship among a set of possible relations

FEATURE-BASED CLASSIFIERS

- Any feature-based classifier can be used, e.g., logistic regression, Random Forest

Sample features:

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

- Word features
  - Headwords of the two potential mentions, and their concatenations, e.g., Airlines, Wagner, and Airlines-Wagner
  - Unigrams and bigrams
  - Unigrams and bigrams in specific positions within each mention
- Named entity features
  - Named entity types and their concatenation, e.g., ORG, PER, and ORG-PER
  - Type of entity
- Syntactic structure - dependencies or constituency syntactic path traversed through the tree between the entities, e.g., below is the dependency path between the entities in our example:

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

NEURAL CLASSIFIERS

- We can use a pretrained encoder (e.g., BERT) to encode the input
- Then a linear classifier to output class probabilities across all possible relation types (multi-class classification)
  - Output is the highest probability relation between the pair of entities (e.g, SUBJ and OBJ in the example)
- To make the model more general, the entities are replaced by their NER tags

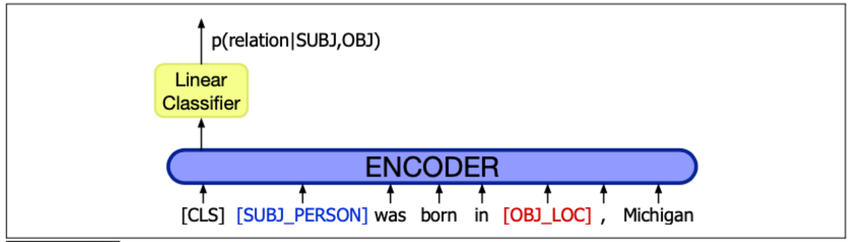


Figure 17.7 Relation extraction as a linear layer on top of an encoder (in this case BERT), with the subject and object entities replaced in the input by their NER tags (Zhang et al. 2017, Joshi et al. 2020).

SEMI-SUPERVISED RELATION EXTRACTION USING BOOTSTRAPPING

SEMISUPERVISED RELATION EXTRACTION VIA BOOTSTRAPPING: OVERVIEW

- Given a few high-precision seed patterns (e.g., handwritten patterns) or tuples of entities
  - Find sentences that contain both entities, e.g., online or across other text corpuses
  - Then learn new patterns from those
- For example, we can create a list of airline/hub pairs, using the seed fact that Ryanair has a hub at Charleroi
  - First we search for sentences that have the terms Ryanair, Charleroi and hub

> 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 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.

  - We then extract words in between the mentions (of the entities) along with the NER tags for the entities, and use the resulting patterns to repeat the search using those new patterns in order to detect new tuples

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

SEMANTIC DRIFT

- Bootstrapping systems need to “rate” the relevance of new tuples (by assigning a confidence score) in order to avoid semantic drift
- Semantic drift is the progressive inclusion of patterns that are erroneous leading to a growing drift in (or departure from) the original meaning
  - An erroneous pattern leads to the creation of problematic patterns, which then result in even more divergent patterns
  - For example:

    if we were to accept the pattern “Sydney has a ferry hub in Circular Quay”

    we will generate the tuple <Sydney, CircularQuay>

    but Sydney is not an airline and CircularQuay is not an airline hub

SEMI-SUPERVISED RELATION EXTRACTION BASED ON DISTANT SUPERVISION

DISTANT SUPERVISION FOR RELATION EXTRACTION: OVERVIEW

- The distant supervision method combines bootstrapping with supervised learning
  - The algorithm generates a lot of noisy pattern features from a large number of seed examples, e.g.,
    - E.g., there are 100K place-of-birth examples across DBPedia and Freebase
    - We can then run NER on a large dataset (e.g., Wikipedia) to find entitities that match the entity types extracted from the above
    - And use the seed examples to extract patterns
  - Then uses supervised classification (feature-based or neural) to differentiate between extracted useful patterns and erroneous patterns
- Pros
  - Uses large unlabeled corpuses of text
  - No semantic drift because the large number of extracted features help constrain the generated patterns
- Cons
  - Generates low precision results (open area of research)
  - Requires large datasets

UNSUPERVISED RELATION EXTRACTION

UNSUPERVISED RELATION EXTRACTION: OVERVIEW

- Extracting new relations (unlabeled data, no existing list of relations) is called open information extraction (aka Open IE) and requires unsupervised learning
- The extracted relations are strings of words, typically starting with a verb (and therefore miss many relations that do not start with a verb)
- Simplified ReVerb algorithm:
  - Run a part-of-speech tagger and entity recognizer over 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
  - For each phrase w, find the nearest noun phrase x to the left which is not a relative pronoun (e.g., who, whom, whose, which and that), wh-word, or the existential "there". Find the nearest noun phrase y to the right.
  - Assign confidence c to the relation $r = (x, w, y)$ using a confidence classifier and return it.

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

- In the original paper, the confidence level is computed using logistic regression trained on 1000 hand labeled examples where each extracted relation is labeled as correct or incorrect

  (+ a number of manually defined features were used)

(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

Figure 17.10 Features for the classifier that assigns confidence to relations extracted by Open Information Extraction system REVERB (Fader et al., 2011).

EVALUATION OF RELATION EXTRACTION

EVALUATION OF RELATION EXTRACTION

- Supervised relation extraction systems are evaluated using human-annotated texts with relations (where we use precision, recall and F-score)
- Semi-supervised and unsupervised methods over large corpuses are evaluated approximately, by sampling the extracted relations & having a human review them

EXTRACTION OF TEMPORAL EXPRESSIONS

TEMPORAL EXPRESSIONS

- Temporal expressions refer to
  - Absolute points in time - can be mapped to calendar dates, times of day, or both
  - Relative times - map to particular times through a reference point, e.g., a week from last Tuesday
  - Durations - spans of time, e.g., seconds, days, weeks, centuries, etc.
  - Sets of the above
- Temporal expressions have temporal lexical triggers as their head word (e.g., example in Figure 17.12)
- The below example uses the XML tag <TIMEX3> used by the TimeML annotation scheme
  - TimeML is a markup language (ML) for temporal and event 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

Figure 17.11 Examples of absolute, relational and durational temporal expressions.

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*

Figure 17.12 Examples of temporal lexical triggers.

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>.

TEMPORAL EXPRESSION EXTRACTION

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 consists of finding the start and end of text spans that constitute temporal expressions
- Approaches
  - Rule-based - first tokens are tagged with PoS tags, then tokens are merged and checked for matching more complex patterns usually involving trigger words (e.g., February) or classes (e.g., MONTH)
  - Sequence labeling - use the IOB scheme utilized by NER, then a sequence model is trained on annotated data

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

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

Figure 17.14 Typical features used to train IOB-style temporal expression taggers.

- Major challenge: avoiding expressions that trigger false positives, e.g.,

1984 tells the story of Winston Smith...

...U2's classic *Sunday* Bloody *Sunday*

TEMPORAL NORMALIZATION: OVERVIEW

- Temporal normalization maps a temporal expression to
  - A specific point in time
  - Or to a duration (length of time but also possibly start and end time)
- Temporal annotations use the VALUE attribute from the ISO 8601 standard (attributes are associated with the TIMEX3 tag)

```
<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>.
```

Figure 17.15 TimeML markup including normalized values for temporal expressions.

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

Figure 17.16 Sample ISO patterns for representing various times and durations.

- ISO attribute notation details in example above
  - W26 above refers to week # 26 (weeks are numbered from 1 to 53 in the ISO standard)
  - Durations use a Pnx format, where n represents length and x represents the unit of length, e.g., $P3Y$ denotes 3 years, and $P1WE$ denotes 1 weekend
  - Each temporal expression is associated with an id, e.g., $t1$, $t2$,...
  - The time of creation of the document (e.g., listed on top of an article) is captured and used to anchor relative temporal expressions (it is referred to as the temporal anchor of the document)
  - Temporal expressions are anchored to a previous temporal expression using the $anchorTimeID$ attribute (to compute specific time point from relative temporal expressions)

TEMPORAL NORMALIZATION CHALLENGES AND APPROACHES

- Generally, rule-based methods are used to capture temporal expressions
- This can be quite challenging
  - In the example below, the way to figure out that the we are referring to the following weekend vs. the past weekend, is due to the tense of the verb *continue* (vs. the example in 17.15 which refers to the previous weekend)

Random security checks that began yesterday at Sky Harbor will continue at least through the weekend.

```
<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>.
```

Figure 17.15 TimeML markup including normalized values for temporal expressions.

- Domain-specific heuristics are used to handle ambiguity e.g.,
  - An expression such as "next Friday" may refer to the Friday in the current week or the following week
  - Some heuristics disambiguate the above based on how close the current date is to Friday
    - E.g., the closer we are to Friday, the more likely it is that the document is referring to Friday during the following week (rather than this week)

EXTRACTION OF EVENTS AND THEIR TIMES

EXTRACTING EVENTS AND THEIR TIMES

- Event extraction involves identifying mentions of events in text, including assigning the event to a point in time, or time interval
- Most event mentions are introduced by verbs or correspond to verbs (e.g., *raised prices*), but there are many exceptions
  - E.g, events may be introduced by NPs such as *the move* and *the increase*
  - Some verbs do not introduce events, e.g., *took effect* refers to the event rather than being event itself
- Event extraction approaches depend on the specific application
- Generally, supervised learning using sequence models with IOB tagging is used, and includes hand-crafted features that have been shown to perform well

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

Figure 17.17 Features commonly used in both rule-based and machine learning approaches to event detection.

TEMPORAL ORDERING OF EVENTS

- Once the events have been extracted, we typically want to order them in time
- Currently, we are only able to come up with a partial ordering, which is a type of binary relation extraction
  - Temporal relations between events are classified into one of the set of Allen relations
- TimeBank is a popular dataset for event extraction annotated using TimeML

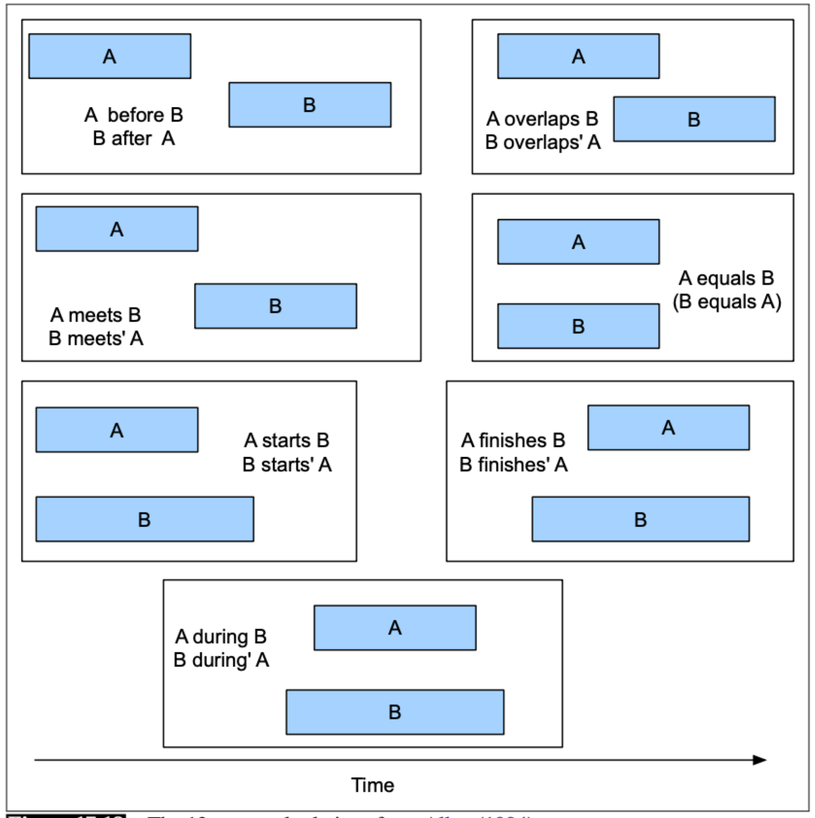


Figure 17.18 The 13 temporal relations from Allen (1984).

```
<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.
```

Figure 17.19 Example from the TimeBank corpus.

- 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}