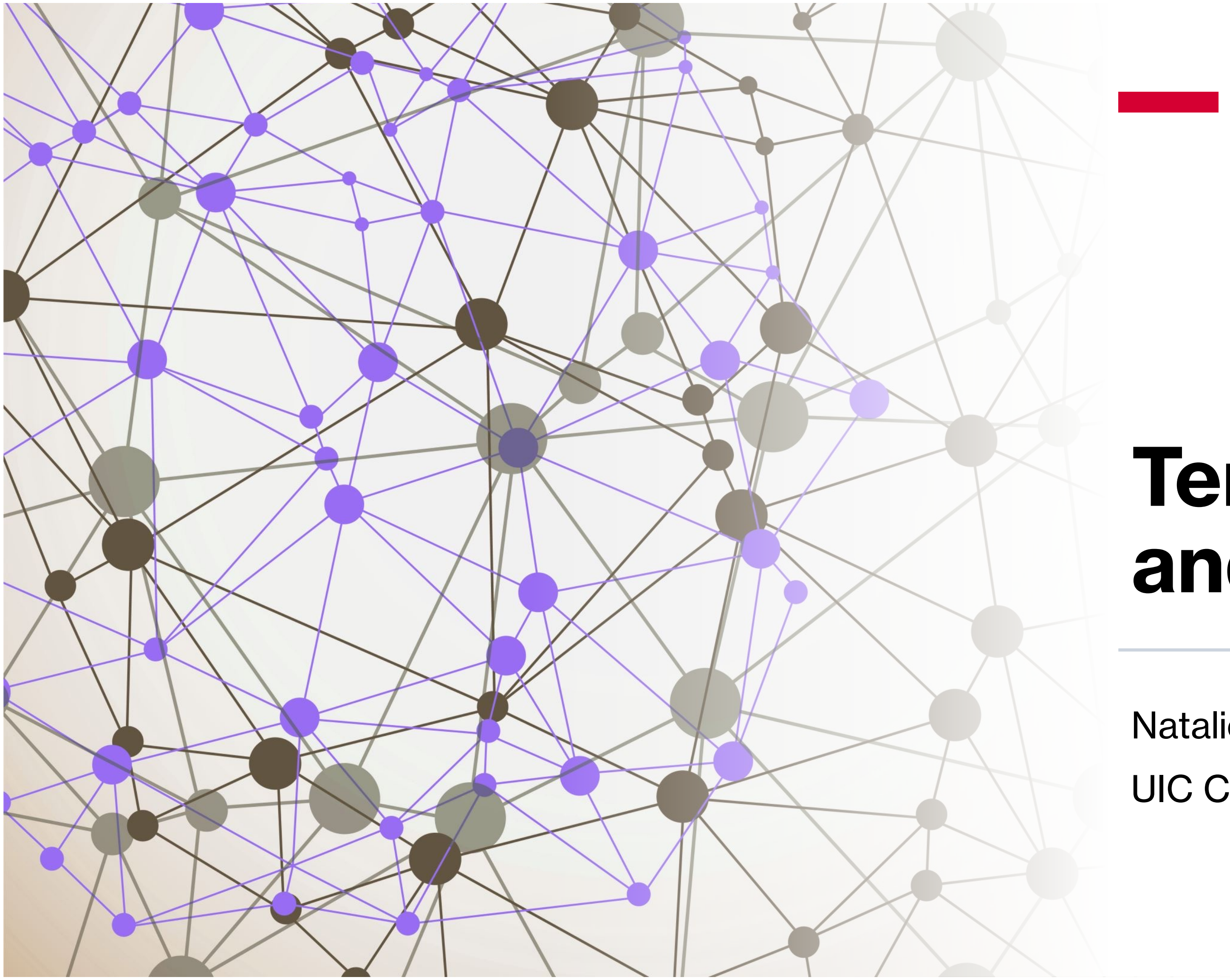


Temporality and Affect

Natalie Parde

UIC CS 421

Many natural language reasoning tasks require that language is converted to structured representations.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, “Oh dear! Oh dear! I shall be late!” (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

Interdimensional Travel

- Leader: Rabbit
- Follower: Alice
- Vehicle: Rabbit-hole

Information Extraction

- The process of converting unstructured information to **structured data**
- Many information extraction tasks involve **relation extraction**
  - The process of finding and classifying semantic relations among entities mentioned in text
  - Semantic relations may be structural, geospatial, or hierarchical
  - Example relations include:
    - Physical location (e.g., relating a person to a geopolitical entity)
    - Organizational subsidiaries (e.g., relating a specialized company to its parent company)
    - Social roles (e.g., relating a person to their sibling)

This Week's Topics

Tuesday

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

This Week's Topics

Tuesday

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

What is an entity?

- **Entities** in text are specific subjects that have been referenced
  - People
  - Locations
  - Times
  - Organizations
- Entities are typically identified and classified using a process known as **named entity recognition**

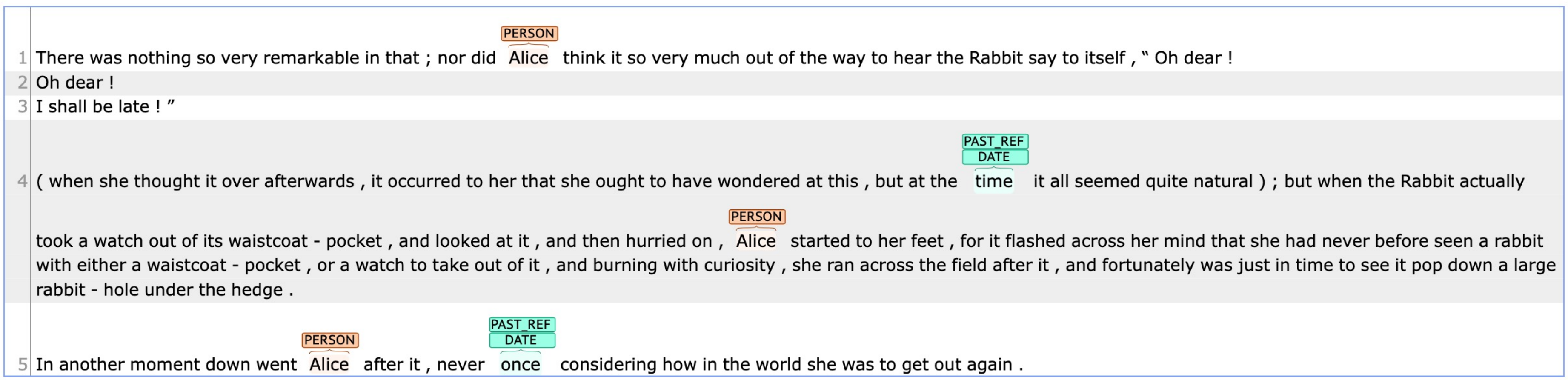
Named Entity Recognition

- Goal:
  1. Find spans of text that constitute proper names
  2. Tag the type of entity
- **Named entity tagsets** define the specific types of entities identified by a named entity recognizer
- Common entity tags:
  - **PER:** Person
    - People or characters
  - **LOC:** Location
    - Regions, mountains, or seas
  - **ORG:** Organization
    - Companies or sports teams
  - **GPE:** Geopolitical entity
    - Countries or states
- Entities can also include **temporal** and **numerical** expressions

Sample Named Entity Tagger Output

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, “Oh dear! Oh dear! I shall be late!” (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.



Named entity recognition is challenging!

- Why?
  - Text segmentation can be ambiguous
    - What is part of the entity and what isn't (where are the entity's boundaries)?
  - Type determination can be ambiguous
    - Some words refer to different types of entities depending on the context

- Did [$_{Org}$ Chicago] win the game?
- I'm visiting friends in [$_{Loc}$ Chicago].
- [$_{GPE}$ Chicago] proposed a new tax ordinance.

How does named entity recognition work?

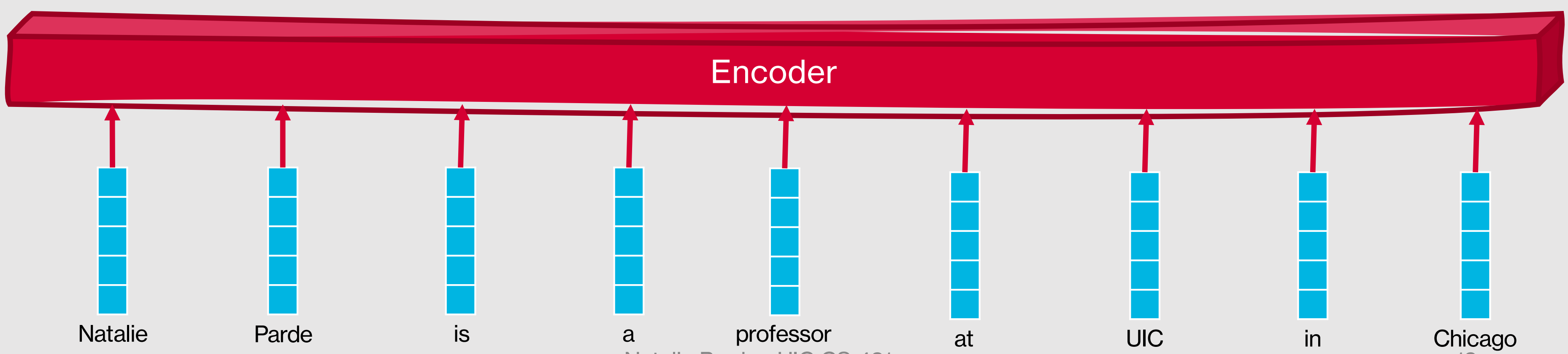
- Typically framed as a sequence labeling problem that performs span recognition
- BIO tags can be used to capture both span boundaries and named entity types

B-PER	I-PER	O	O	O	O	O	O	B-ORG	I-ORG	I-ORG	I-ORG	O	O	O	O	O	B-LOC
Natalie	Parde	is	a	faculty	member	at	the	University	of	Illinois	Chicago,	a	large	public	university	in	Chicago.

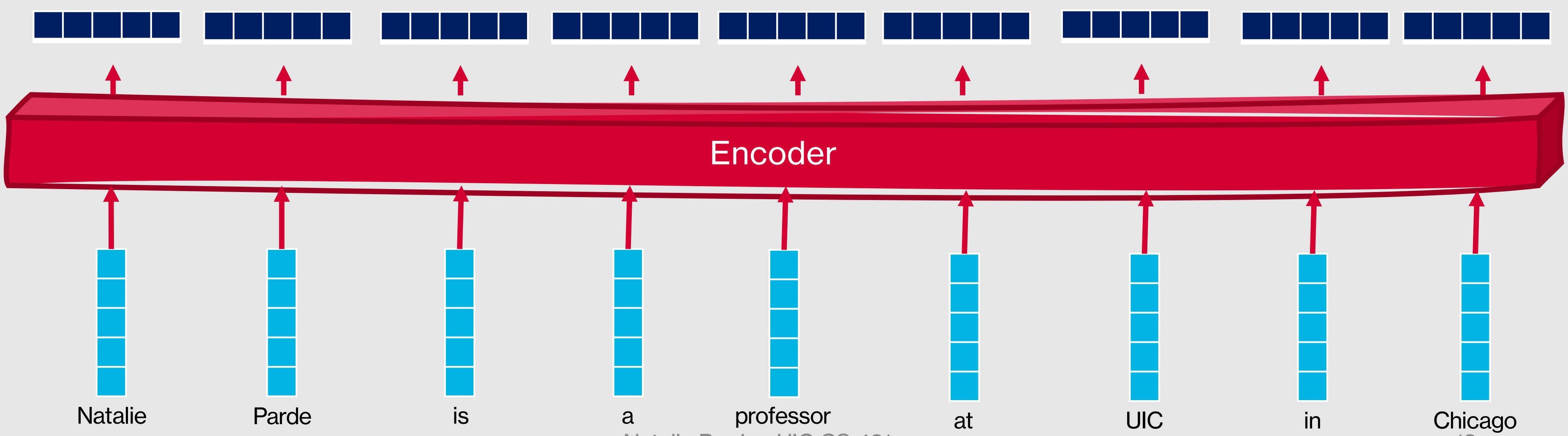
What kind of approaches do we use to implement named entity recognizers?

- Typically we will use supervised machine learning approaches to perform BIO tagging for named entity recognition
- Popular general-purpose named entity corpus:
  - **OntoNotes:** https://catalog.ldc.upenn.edu/LDC2013T19
    - Available in English, Chinese, and Arabic
- Popular biomedical named entity corpus:
  - **CRAFT:** https://github.com/UCDenver-ccp/CRAFT
- Popular literary named entity corpus:
  - **LitBank:** https://github.com/dbamman/litbank

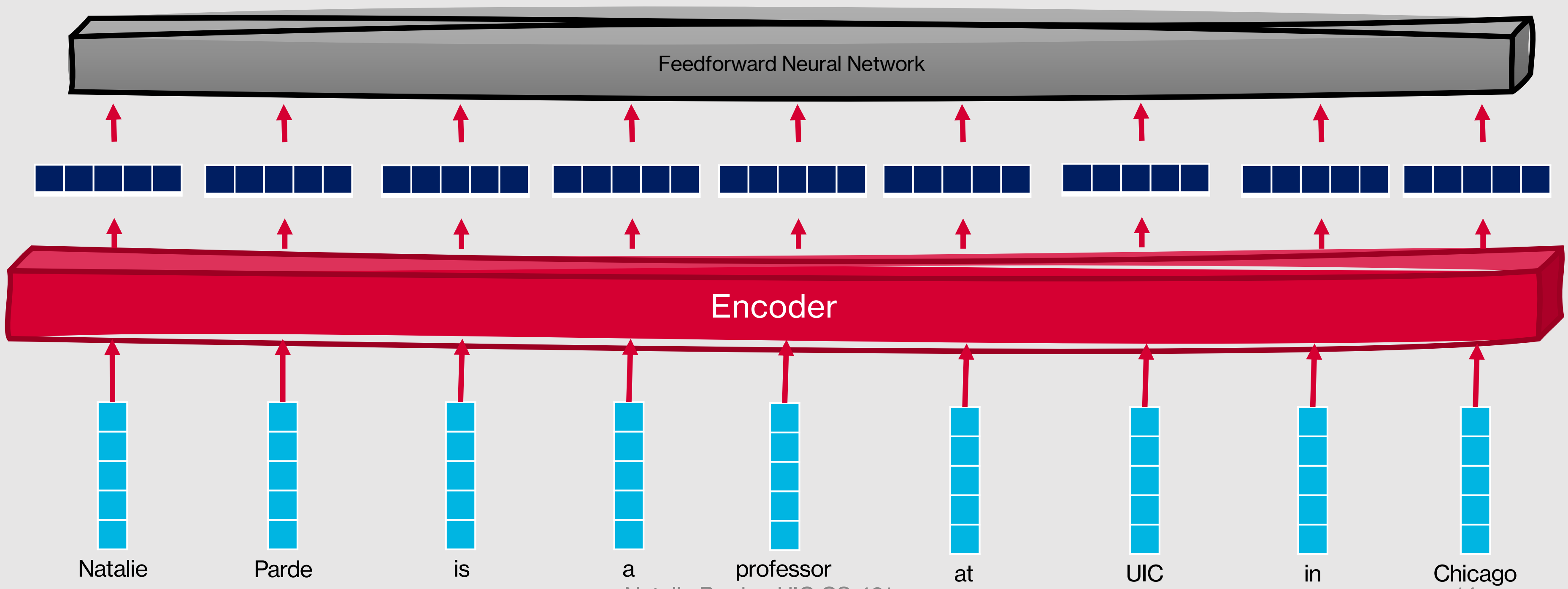
Example Neural Named Entity Recognizer



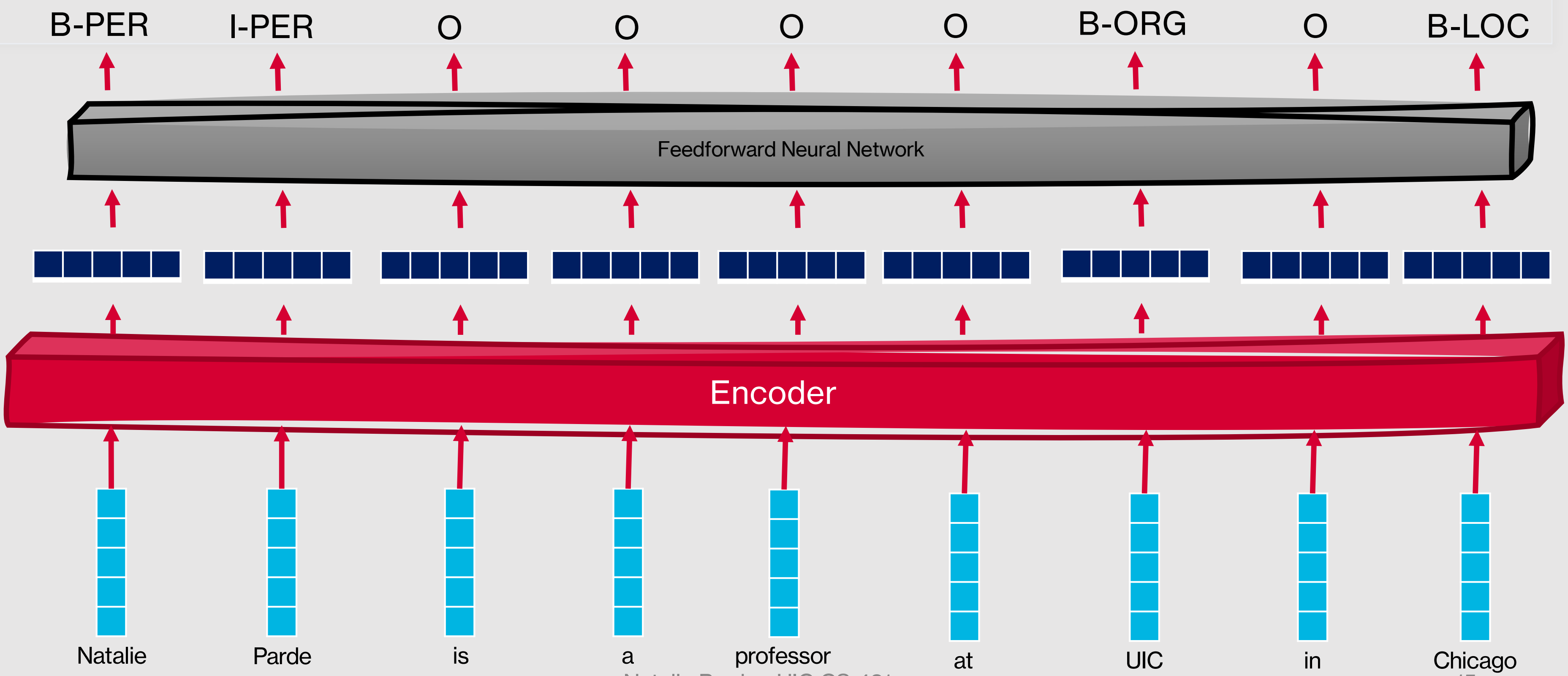
Example Neural Named Entity Recognizer



Example Neural Named Entity Recognizer



Example Neural Named Entity Recognizer



Rule-Based Named Entity Recognition

- Specialized commercial approaches often combine supervised models with rule-based methods
- Use regular expressions to tag unambiguous entity mentions, and then use supervised learning methods incorporating this information

This Week's Topics

Named Entity Recognition
Relation Extraction
Event Extraction
Temporal Logic

Tuesday

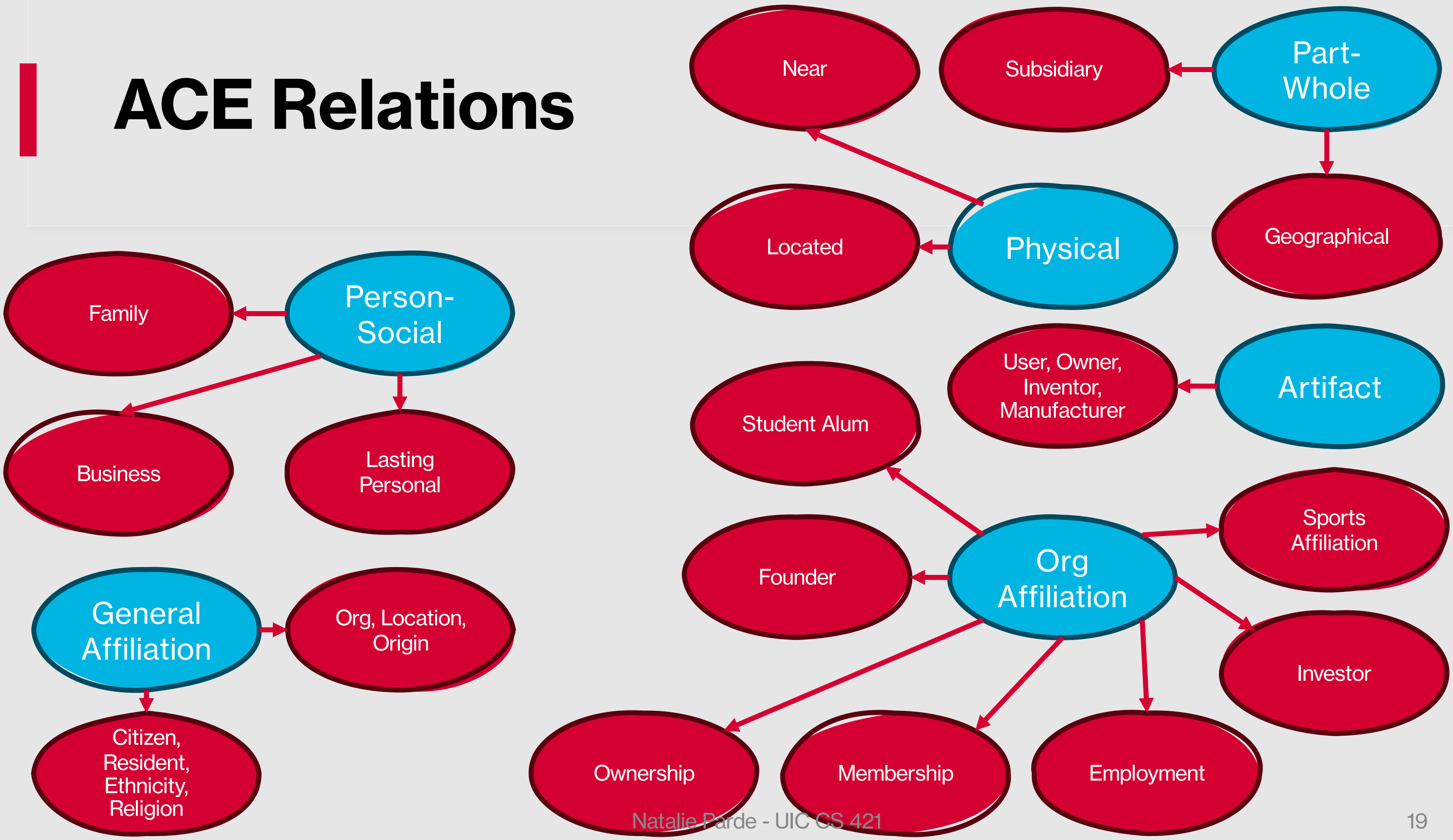
Thursday

Temporal Analysis
Affective Lexicons
Inducing Affect and Association
Affective Analysis

Once we've detected named entities, we can determine relationships between them.

- Numerous tagsets available
- Popular tagset: **Automatic Content Extraction (ACE) Relations**
  - 17 physical, membership, affiliation, citizenship, and discourse relations
  - Each relation links two entities

ACE Relations



Extracting these relations also helps us interpret input according to model-theoretic semantics!

Domain:
Natalie, Devika, Nikolaos, Mina = {a, b, c, d}
Giordano's, IDOF, Artopolis = {e, f, g}

Classes:
PER = {Natalie, Devika, Nikolaos, Mina} = {a, b, c, d}
ORG = {Giordano's, IDOF, Artopolis} = {e, f, g}

Relations:
Org Affiliation = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}

Domain-Specific Relation Sets

- For specific applications, domain-specific sets of relations may be used instead
- Popular set of entities and relations in the medical domain: **UMLS**
  - 134 broad subject categories and entity types
  - 54 relations between entities
- Browse the UMLS semantic network: https://uts.nlm.nih.gov/uts/umls/semantic-network/root

UMLS Example

Entity or event	Relation	Entity or event
Injury	Disrupts	Physiological function
Medical device	Diagnoses	Disease or syndrome
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

doppler echocardiography diagnoses *acquired stenosis*

Other Sources of Relations

- Wikipedia contains structured tables associated with certain articles that can be turned into a metalanguage known as the **Resource Description Framework (RDF)**
  - RDF tripes are relationship(entity, entity) relations created using this metalanguage, which map to subject-predicate-object expressions
    - Location(Learning Center Building C, UIC)
- **DBpedia** was created from Wikipedia: https://www.dbpedia.org/
- **TACRED** contains 106,264 examples of relation triples from news and web text, with 41 relation types: https://nlp.stanford.edu/projects/tacred/
- **SemEval 2010 Task 8** dataset contains 10,717 examples of relation triples, with 9 relation types: http://www.kozareva.com/downloads.html

How can we extract relations?

- Rule-based approaches were established first, but are still common
- Works by searching for specific lexicosyntactic patterns originally proposed by Marti Hearst, known as **Hearst patterns**
  - https://aclanthology.org/C92-2082/

Example Hearst Patterns

- Identifying hypernyms (NP_H)
  - 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

Extended Hearst Patterns with Named Entity Constraints (Modern Update to Hearst Patterns!)

- PER, POSITION of ORG → George Marshall, Secretary of State of the United States
- PER (be)? ((named)|(appointed)) Prep? ORG POSITION → George Marshall was named US Secretary of State

Advantages and Disadvantages of Hearst Patterns

Advantages:

- High precision
- Can be tailored to specific domains

Disadvantages:

- Tend to have low-recall
- Require substantial human effort for pattern creation

Supervised Relation Extraction

- Typically requires two steps:
  1. Find all possible pairs of named entities in the text (typically restricted to the same sentence)
  2. Classify the relation for each pair (either non-existent or one of the allowable relation types)
- Optional filtering step: First make a binary decision regarding whether a given pair of entities are related in any way (and then only classify relations for those that are expected to be related)

Supervised Relation Extraction Algorithm

```
function FindRelations(words) returns relations

      relations ← []
      entities ← FindEntities(words)
      forall entity pairs {e1, e2} in entities do
            if Related?(e1, e2)
                  relations ← relations + ClassifyRelation(e1, e2)
      return relations
```

Feature-Based Implementations

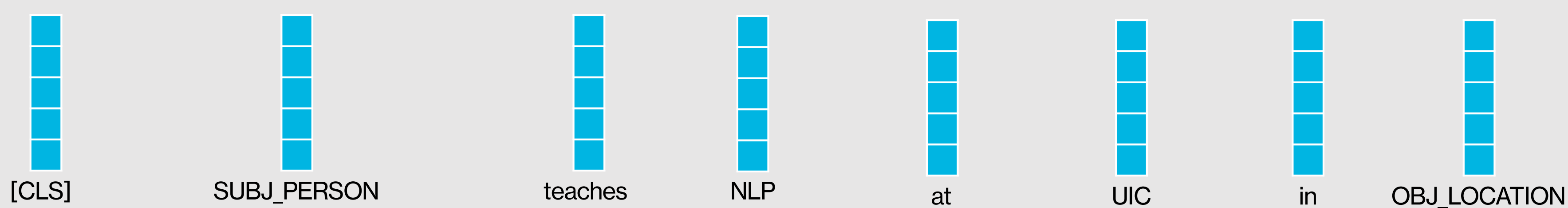
- Feature-based implementations of this algorithm (e.g., using logistic regression for relation classification) may use:
  - **Word features:** Unigrams and bigrams related to specific mentions of the entities and the words around them
  - **Named entity features:** Named entity types and high-level categorizations of those types, and number of entities between these two entities
  - **Syntactic structure features:** Constituent and dependency paths between the entities

Neural Implementations

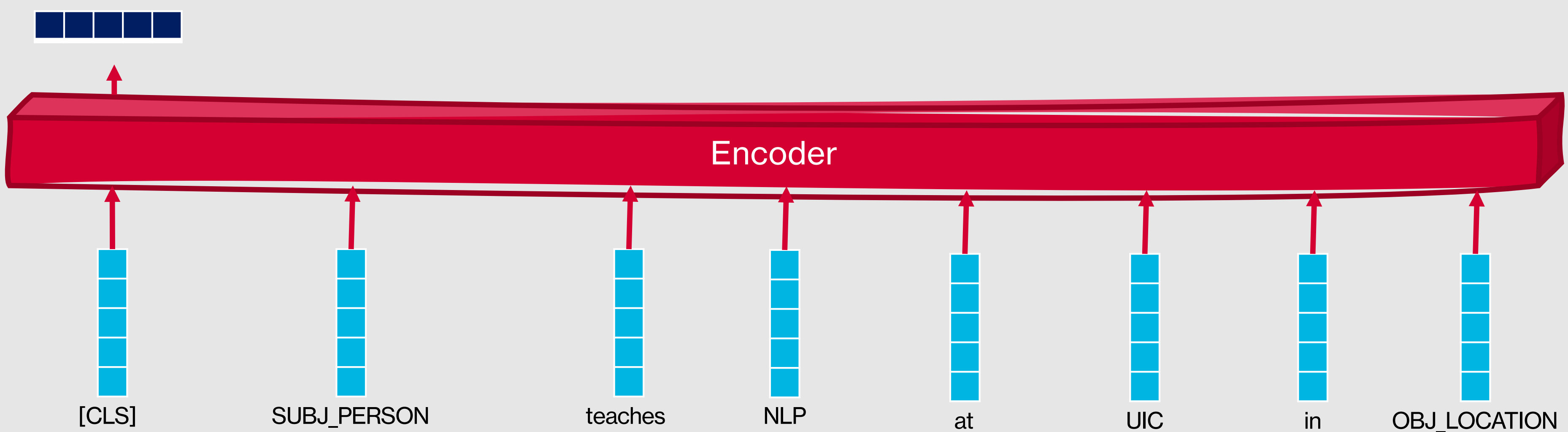
- Create a partially **delexicalized** version of the input by replacing the entities *being classified* with their named entity tags
- Finetune a pretrained model to predict the correct relation for the entities using the [CLS] token
  - Ideally, the base model should be pretrained on tasks that do not specify a sequence [SEP] token

Example Neural Relation Extraction

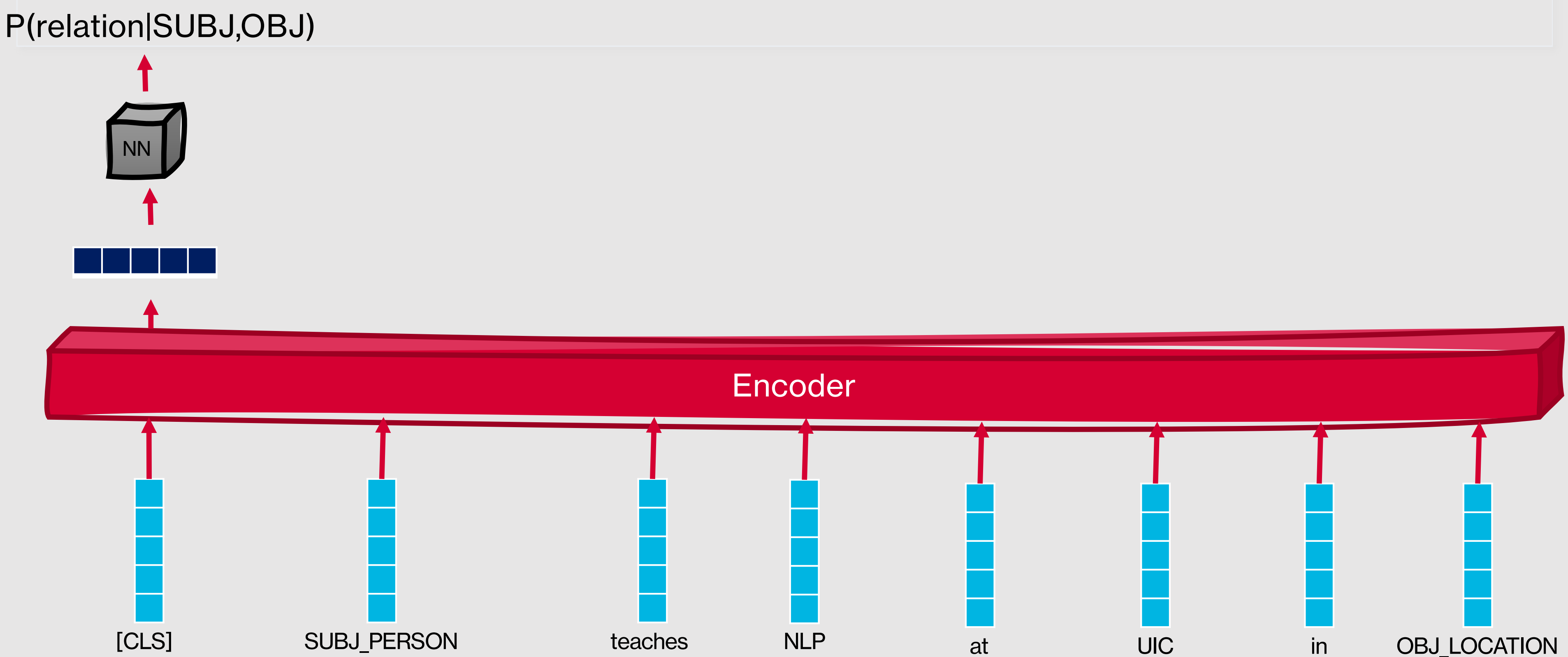
Example Neural Relation Extraction



Example Neural Relation Extraction



Example Neural Relation Extraction



Advantages and Disadvantages of Supervised Relation Extraction

Advantages:	Disadvantages:
• Given sufficient training data and test data from a similar distribution, generally results in strong performance	• Expensive • Tends not to generalize well to out-of-domain text

Semi-Supervised Relation Extraction

- Since building high-quality relation datasets is expensive and time-consuming, there has been substantial interest in **semi-supervised** and **unsupervised** approaches for this task
- Semi-supervised learning requires a small number of high-quality **seed examples**, such as:
  - Hearst patterns
  - Seed tuples

Bootstrapping Algorithm

- One way to perform semi-supervised learning is by **bootstrapping** a classifier
  - Take the seed examples and find data matching those samples in some other data source
  - Extract and generalize the context in those new samples to learn new patterns
  - Repeat
- For relation extraction, the seed examples would be seed pairs of entities

Bootstrapping Algorithm

```
function Bootstrap(Relation R) returns new relation tuples

        tuples ← Gather a set of seed tuples that have relation R
        iterate
                sentences ← Find sentences that contain entities in tuples
                patterns ← Generalize the context between and around entities in sentences
                newpairs ← Use patterns to identify more tuples
                newpairs ← newpairs with high confidence
                tuples ← tuples + newpairs
        return tuples
```

Bootstrapping in Practice

Task: Create a list of airline/hub pairs from free text.

Seed Fact:
hub(Ryanair, Charleroi)

Bootstrapping in Practice

Task: Create a list of airline/hub pairs from free text.

Seed Fact:
hub(Delta, Detroit)

Corpus search:
“Delta Detroit hub”

Bootstrapping in Practice

Task: Create a list of airline/hub pairs from free text.

Seed Fact:
hub(Delta, Detroit)

Corpus search:
“Delta Detroit hub”

Results:
- Popular airline Delta, which uses Detroit as a hub, scrapped all weekend flights out of the airport.
- All flights in an out of Delta’s hub at Detroit airport were grounded on Friday…
- A spokesperson at Detroit, a main hub for Delta, estimated that 8000 passengers had already been affected.

Bootstrapping in Practice

Task: Create a list of airline/hub pairs from free text.

Seed Fact:
hub(Delta, Detroit)

Results:
- Popular airline Delta, which uses Detroit as a hub, scrapped all weekend flights out of the airport.
- All flights in an out of Delta's hub at Detroit airport were grounded on Friday…
- A spokesperson at Detroit, a main hub for Delta, estimated that 8000 passengers had already been affected.

Pattern Extraction:
- / [ORG], which uses [LOC] as a hub /
- / [ORG]'s hub at [LOC] /
- / [LOC], a main hum for [ORG] /

Bootstrapping in Practice

Task: Create a list of airline/hub pairs from free text.

Seed Fact:
hub(Delta, Detroit)

Patterns:

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

Bootstrapping in Practice

Task: Create a list of airline/hub pairs from free text.

Seed Fact:
hub(Delta, Detroit)

Patterns:
- / [ORG], which uses [LOC] as a hub /
- / [ORG]'s hub at [LOC] /
- / [LOC], a main hum for [ORG] /

New Facts:
- hub(United, O'Hare)
- hub(American Airlines, DFW)
- hub(Southwest, Dallas)

Semantic Drift

- With bootstrapping, an erroneous pattern can lead to erroneous facts being added to the data
- Erroneous facts can then in turn lead to more erroneous patterns being created
- This problematic cycle is known as **semantic drift**

How can we address semantic drift?

- Assign confidence values to new patterns and tuples!
- Confidence values for patterns should balance:
  - The pattern's performance with respect to the current set of tuples
    - hits(*p*): The set of tuples in the larger set *T* that *p* matches while searching the document collection *D*
  - The pattern's productivity (number of matches it produces)
    - finds(*p*): The total set of tuples that *p* finds in *D*
  - $\text{Confidence}(p) = \frac{|\text{hits}(p)|}{|\text{finds}(p)|} \log(|\text{finds}(p)|)$
  - In turn, the probability of all supporting patterns for a tuple *t* being wrong is the product of their individual failure probabilities:
    - $\text{Confidence}(t) = 1 - \prod_{p \in P'}(1 - \text{Confidence}(p))$

Distant Supervision for Relation Extraction

- Another way to avoid expensive manual labeling of relation labels is to perform **distant supervision** with indirect sources of training data
- Combines advantages of bootstrapping and supervised learning by:
  - Using a large database to acquire many seed examples
  - Create many noisy pattern features from these examples
  - Combine these in a supervised classifier

Distant Supervision Algorithm

```
function DistantSupervision(Database D, Text T) returns relation classifier C
        observations = []
        foreach relation R in D
                foreach tuple (e1, e2) of entities with relation R
                        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
```

Distant Supervision in Practice

Task: Learn the place-of-birth relationship between people and their birth cities from free text.

Database Search:
place-of-birth

DBpedia

Distant Supervision in Practice

Task: Learn the place-of-birth relationship between people and their birth cities from free text.

Database Search:
place-of-birth

Retrieved Seed Tuples:
<Edwin Hubble, Marshfield>
<Albert Einstein, Ulm>
(and many more!)

DBpedia

Distant Supervision in Practice

Task: Learn the place-of-birth relationship between people and their birth cities from free text.

Database Search: place-of-birth

Retrieved Seed Tuples:
<Edwin Hubble, Marshfield>
<Albert Einstein, Ulm>
(and many more!)

Sentences with two named entities that match a seed tuple:
Hubble was born in Marshfield....
Einstein, born (1879), Ulm....
Hubble's birthplace in Marshfield....
(and many more!)

Distant Supervision in Practice

Task: Learn the place-of-birth relationship between people and their birth cities from free text.

Database Search: place-of-birth

Retrieved Seed Tuples:
<Edwin Hubble, Marshfield>
<Albert Einstein, Ulm>
(and many more!)

Sentences with two named entities that match a seed tuple:
Hubble was born in Marshfield….
Einstein, born (1879), Ulm….
Hubble's birthplace in Marshfield….
(and many more!)

With this information, extract features to train the classification model!

Advantages and Disadvantages of Distant Supervision

Advantages:

- Allows for the use of a large amount of training data
- Allows for detailed pattern learning
- Prevents semantic drift
- Doesn't require labeled training data

Disadvantages:

- Tends to produce low-precision results
- Can only work when a large enough relevant database already exists

Unsupervised Relation Extraction

- Uses in situations with no labeled training data, such as when you are working with:
  - Low-resource domains
  - New domains
- Often referred to as **open information extraction (Open IE)**
- Relations are generally strings of words, rather than more formal relations

How does open information extraction work?

1. Run a part-of-speech tagger and named entity recognizer over the sentence
2. For each verb in that sentence, find the longest sequence of words w that starts with a verb and satisfies any specified syntactic and lexical constraints
3. For each retrieved sequence of words, find the nearest noun phrase x prior to the sequence that is not a relative pronoun, wh-word, or existential “there”
4. Likewise, find the nearest noun phrase y after the sequence
5. Assign a confidence c to the relation $r = (x, w, y)$ using a confidence classifier

Advantages and Disadvantages of Unsupervised Relation Extraction

Advantages:

- Can handle many relations without having to specify them in advance
- No labeled training data or seed examples are necessary

Disadvantages:

- All recognized relation strings need to be mapped to canonical form prior to storage in databases or knowledge graphs
- Most methods focus on verb-based relations, so nominal relations may not be adequately captured

Evaluating Methods for Relation Extraction

- Supervised relation extraction systems can be evaluated using standard NLP metrics (e.g., precision, recall, and F-measure)
- Semi-supervised and unsupervised relation extraction systems are more challenging to evaluate since there is typically not an existing gold standard for comparison

Evaluating Semi-Supervised and Unsupervised Systems

- **Approximate precision** by randomly sampling output relations and asking a human to manually evaluate them
  - $\hat{P} = \frac{\text{\# correctly extracted relation tuples in the sample}}{\text{\# extracted relation tuples in the sample}}$
- **Approximate recall** by computing precision at different sample sizes
  - Precision for most-confident 1000 new relations
  - Precision for most-confident 10,000 new relations
  - And so forth!

This Week's Topics

Tuesday

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

Event Extraction

- **The process of finding events in which entities in the text participate**
- In general, event extractors are designed to classify events based on their aspectual and temporal properties
- Different event tagsets exist, but may include:
  - Actions
  - States
  - Reporting events
  - Perception events

What is an event mention?

Any expression denoting an event or state that can be assigned to a particular point or interval in time

Most event mentions in English correspond to verbs (and most English verbs introduce events) although this is not a requirement

[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 Datasets

- Typically also include annotations for temporal and aspectual information
- TempEval Shared Tasks:
  - https://aclweb.org/aclwiki/Temporal_Information_Extraction_(State_of_the_art)
  - https://alt.qcri.org/semeval2017/task12/

[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.

said(class=reporting, tense=past, aspect=perfective)

Event Extraction Approaches

- Generally framed as a supervised sequence labeling problem
- BIO tags are used to assign event classes and attributes
- Many events correspond to fairly common, stereotypical situations in the world
  - **Scripts** are prototypical sequences of sub-events, participants, and their roles
  - We can represent scripts using simple **templates**

Template Filling

- Templates have fixed sets of **slots**
- Each slot takes **slot-fillers** as values belonging to particular classes
- Slot fillers may be text segments extracted directly from the text, or concepts that have been inferred from text elements through additional preprocessing

Filled Template

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.

Fair-Raise Attempt:

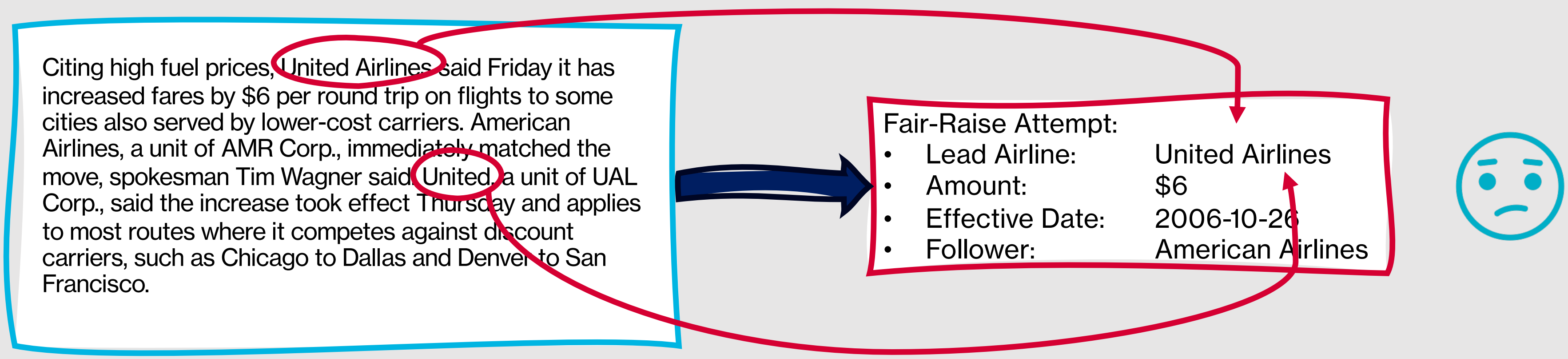
- Lead Airline: United Airlines
- Amount: $6
- Effective Date: 2006-10-26
- Follower: American Airlines

How do we perform template filling?

- Goal: Create one template for each event in the input, filling in the slots with text spans
- Requires two separate tasks:
  - **Template recognition:** Determines whether a given template is present in the sentence
  - **Role-filler extraction:** Detects each role for a template

What if we find multiple text segments that can fill the same slot?

- This is fine (if role-filler extraction is performed correctly, they likely refer to the same entity!)
- We can resolve this using **coreference resolution**



This Week's Topics

Tuesday

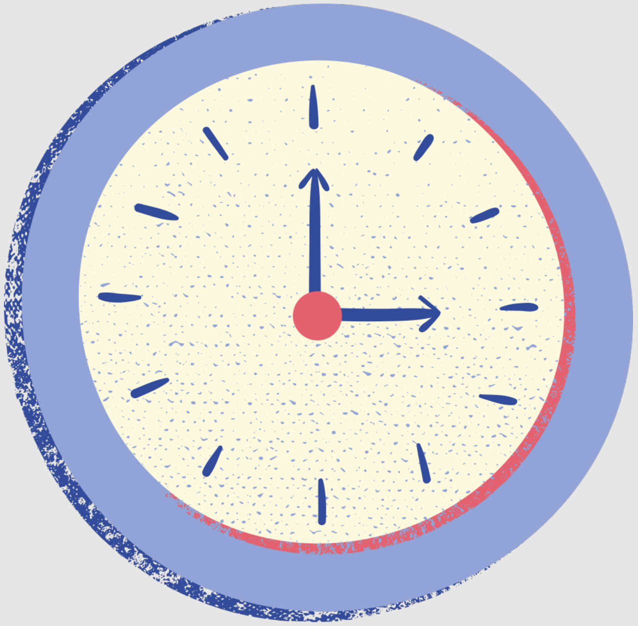
- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

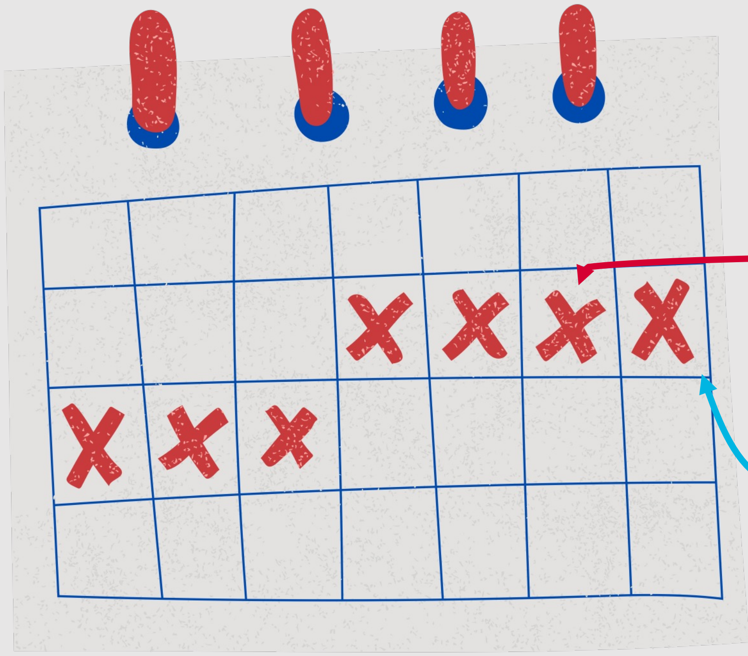
- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

Time and Temporality

- Events are situated within time, and they can also relate to one another temporally
  - Events happen at particular dates and times
  - Events can occur before, after, or simultaneously with one another



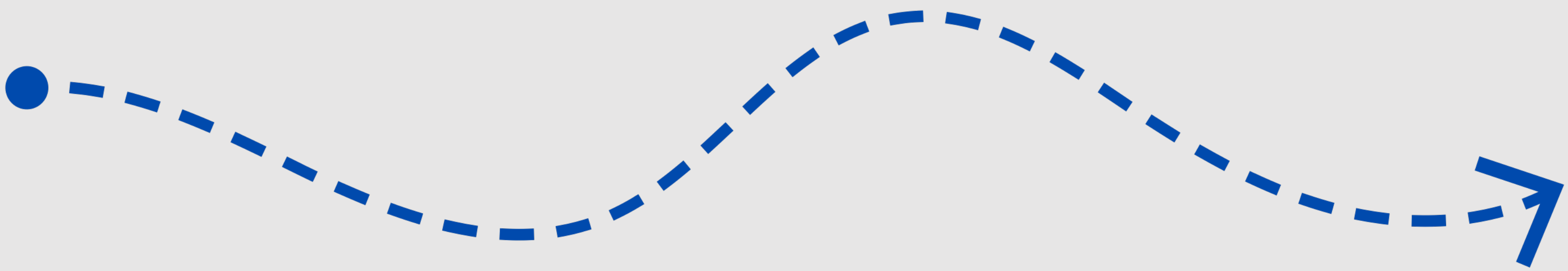
Temporal Expressions



- Temporal expressions allow us to understand how events are situated within time
- These expressions may be explicit statements of date or time:
  - Project Part 2 is due on Friday at 12 p.m.
- Or they may be in reference to other expressions or events:
  - 26 hours from now

Representing Time

- Although there are numerous philosophical theories of time, the most straightforward theory holds that:
  - **Time flows forward**
  - **Events are associated with points or intervals in time**
- Assuming this to be true, we can order events by situating them on a timeline, with one event preceding another if time flows from the first event to the second
- If we situate the current moment in time along this timeline, we build notions of past, present, and future
- **Temporal logic** is a formal way to represent temporal information, typically within the first-order logic framework



How do we adapt first-order logic to represent temporality?

- Simple first-order logic representations focus on meaning irrespective of time

$$\boxed{\exists e\ \mathrm{Arriving}(e) \wedge \mathrm{Arriver}(e, \mathrm{Speaker}) \wedge \mathrm{Destination}(e, \mathrm{Chicago})}$$

I arrived in Chicago.

I am arriving in Chicago.

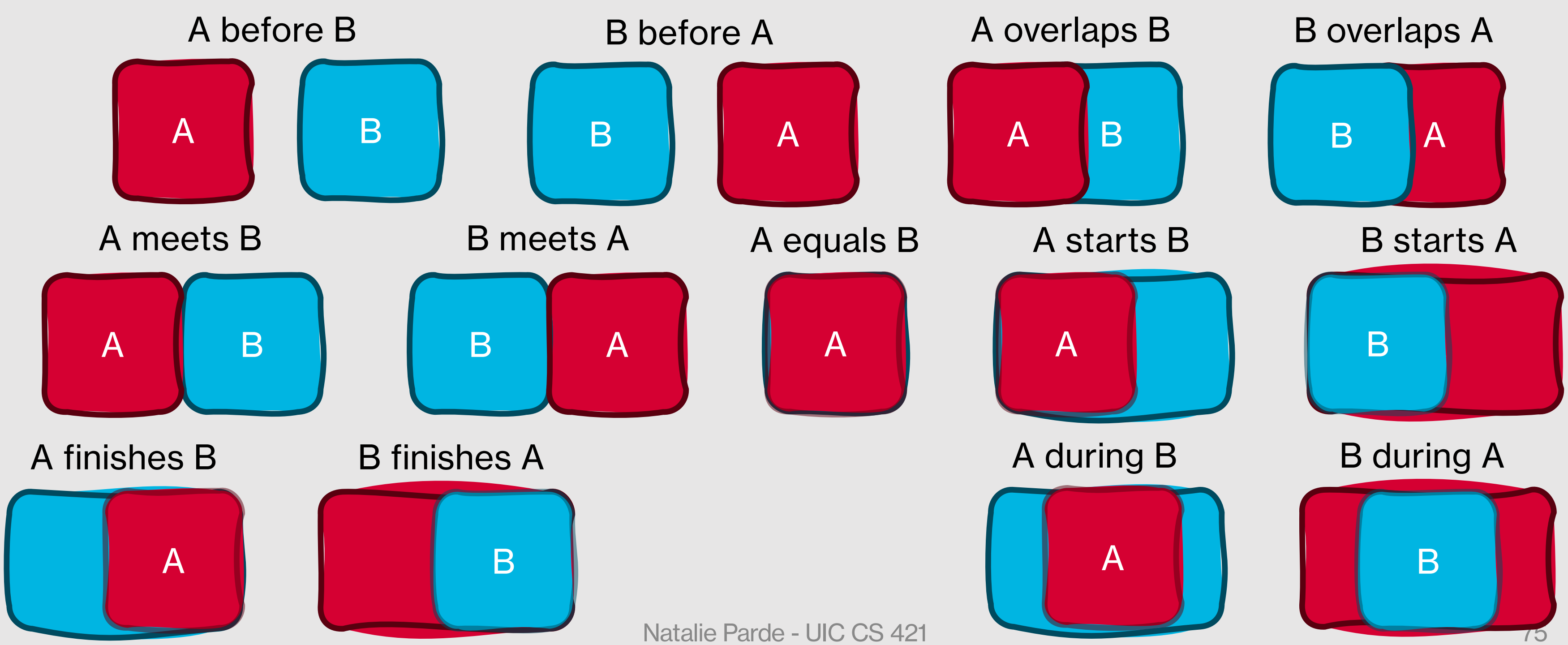
I will arrive in Chicago.

How do we adapt first-order logic to represent temporality?

- Specify temporal information based on the event's order with respect to other events using attributes of the **event variable**
  - **Interval algebra** is one framework developed by James F. Allen that is used to discuss temporal ordering relationships
    - All events and time expressions are modeled as intervals (no time points!), which can be long or very short

Allen Relations

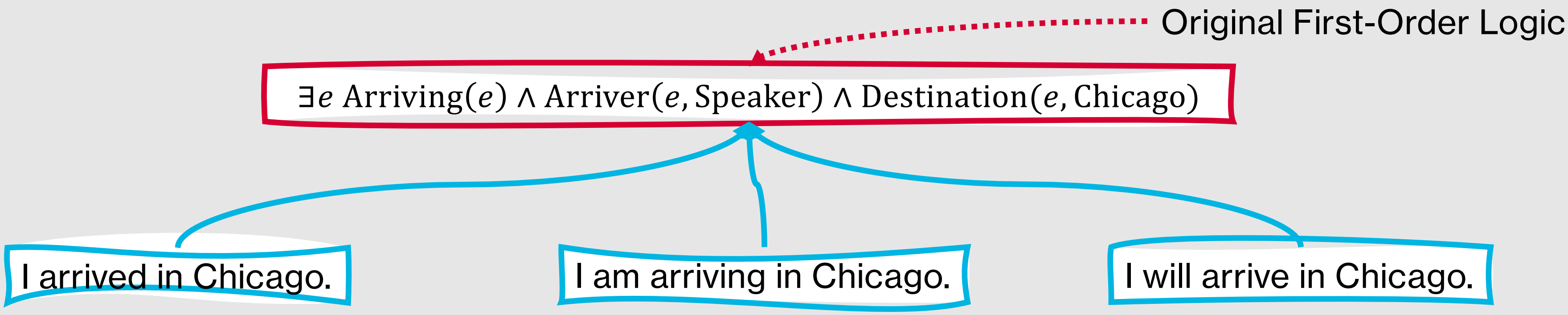
- There are 13 relations that can hold between intervals A and B in interval algebra



How do we include interval algebra in our first-order logic model?

1. Add a temporal variable representing the interval corresponding to the event
2. Add a very small interval corresponding to the current time (“Now”)
3. Add temporal predicates relating the event to the current time as indicated by event verb tense

Interval Algebra in Practice



Interval Algebra in Practice

Updated First-Order Logic

Original First-Order Logic

$$\exists e \ \text{Arriving}(e) \land \text{Arriver}(e, \text{Speaker}) \land \text{Destination}(e, \text{Chicago})$$

I arrived in Chicago.

I am arriving in Chicago.

I will arrive in Chicago.

$$\exists e, i \ \text{Arriving}(e) \land \text{Arriver}(e, \text{Speaker}) \land \text{Destination}(e, \text{Chicago}) \land \text{IntervalOf}(e, i) \land \text{Before}(i, \text{Now})$$

Interval Algebra in Practice

Original First-Order Logic

$$\exists e\ Arriving(e) \wedge Arriver(e, Speaker) \wedge Destination(e, Chicago)$$

I arrived in Chicago.

I am arriving in Chicago.

I will arrive in Chicago.

Updated First-Order Logic

$$\exists e, i\ Arriving(e) \wedge Arriver(e, Speaker) \wedge Destination(e, Chicago) \wedge IntervalOf(e, i) \wedge Before(i, Now)$$

$$\exists e, i\ Arriving(e) \wedge Arriver(e, Speaker) \wedge Destination(e, Chicago) \wedge IntervalOf(e, i) \wedge During(i, Now)$$

Interval Algebra in Practice

Updated First-Order Logic

Original First-Order Logic

$\exists e\ Arriving(e) \land Arriver(e, Speaker) \land Destination(e, Chicago)$

I arrived in Chicago.

I am arriving in Chicago.

I will arrive in Chicago.

$\exists e, i\ Arriving(e) \land Arriver(e, Speaker) \land Destination(e, Chicago) \land IntervalOf(e, i) \land Before(i, Now)$

$\exists e, i\ Arriving(e) \land Arriver(e, Speaker) \land Destination(e, Chicago) \land IntervalOf(e, i) \land During(i, Now)$

$\exists e, i\ Arriving(e) \land Arriver(e, Speaker) \land Destination(e, Chicago) \land IntervalOf(e, i) \land After(i, Now)$

Summary: Relation and Event Extraction

- Natural language can be converted to **structured representations** using information extraction techniques
- **Named entity recognition** identifies the specific entities that participate in structured relationships or events
- **Relations** can be extracted using a variety of approaches
  - **Hearst patterns** are specialized rules for extracting relations
  - Supervised learning can be used to train feature-based or neural approaches to relation extraction
  - **Semi-supervised** learning or **distant supervision** can be used to extract relations without using a large labeled training set
  - **Open information extraction** uses unsupervised methods to extract relations
- **Events** are actions or states that can be assigned to a particular point or interval in time
- **Template filling** approaches recognize specific **scripts** or events as **templates** and assign segments from the text to roles represented by a fixed set of slots participating in those templates
- **Temporal expressions** allow us to understand how events are situated within time
- We can represent **temporal logic** using the first-order logic framework and incorporating **interval algebra**

This Week's Topics

Tuesday

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

Temporal Ambiguity

- Although relating verb tenses with points in time may initially seem simple, there are many opportunities for ambiguity to arise
  - *Okay, we fly from Chicago to Burlington at noon.*
    - Present tense verb indicating future event
  - *Flight 1902 arrived late.*
    - Past tense verb.
  - *Flight 1902 had arrived late.*
    - Past tense verb with respect to some unnamed event.
- To address this, Hans Reichenbach introduced the notion of reference points that are separate from the utterance time and event time
  - Reichenbach, Hans (1947). *Elements of Symbolic Logic*. New York: Macmillan & Co.

Reichenbach's Reference Point

When I got off the blue line, I taught CS 421.

Happened before the utterance

When I got off the blue line, I had taught CS 421.

Reichenbach's Reference Point

Reference point

When I got off the blue line, I taught CS 421.

Happened before the utterance

Reference point

When I got off the blue line, I had taught CS 421.

Reichenbach's Reference Point

Reference point

When I got off the blue line, I taught CS 421.

Happened before the utterance

Happened *after* the speaker got off the blue line

Reference point

When I got off the blue line, I had taught CS 421.

Happened *prior to* when the speaker got off the blue line

Reichenbach's Reference Point

Reference point

When I got off the blue line, I taught CS 421.

Happened before the utterance

Reference point

When I got off the blue line, I had taught CS 421.

Happened *after* the speaker got off the blue line

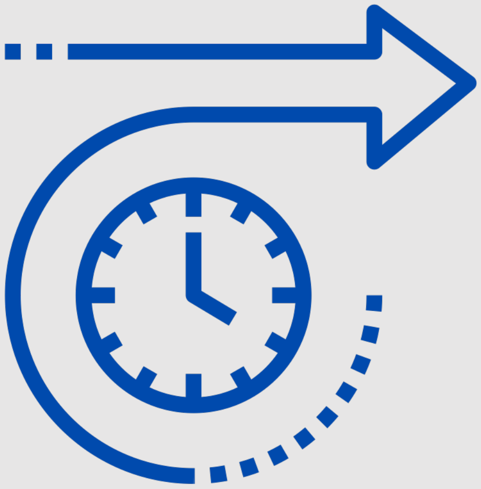
- Reference point precedes the teaching event

Happened *prior to* when the speaker got off the blue line

- Teaching event precedes the reference point

Temporal Information Beyond Tense

- Although Reichenbach's approach is clearly illustrated using English verb tenses, there are also many other ways that languages can convey temporal information
  - CS 421 is held *in the morning*
  - Assignments are due at *noon* and the weekend begins *afterwards*!



Time and Metaphor

- Many languages (including English) frequently rely on **metaphors** to express temporality
- Most frequent in English: <TIME> is <SPACE>
  - *In* the morning
  - *Around* noon
  - Midnight is *near*
- This facilitates understanding of a complex topic for humans, but may create additional complexities when processing temporal expressions computationally

Aspect

- **Aspect** defines categories of events or states based on their temporal structure
- Events may be:
  - Ongoing or complete
  - At a specific point or over an interval of time

Aspectual Distinctions

- **Events** involve change, whereas **states** do not
- **Stative expressions** indicate the state or property of an event participant at a given point in time

I **like** CS 421.

I **need** to finish my research project.

I **want** to take CS 521.

- Event participant is experiencing something at a specific point in time
- Event doesn't involve internal change over time

Aspectual Distinctions

Activity, **achievement**, and **accomplishment** expressions all involve some form of change

Activity expressions describe events undertaken by a participant that occur over a span of time and have no particular end point

Accomplishment expressions describe events that take place over time but have a natural end point and result in a particular state

Achievement expressions describe events that happen in an instant and result in a particular state, without conceptualizing the process or activity leading up to that state

Activity, Accomplishment, and Achievement Expressions

Activity

Souvik lives in Chicago.

Natalie works at UIC.

- Event participant is or has engaged in the activity for some period of time
- No specification that the activity might have stopped

Accomplishment

Abari ordered some ice cream.

Meghan traveled to New Orleans.

- Event occurs over some period of time
- Event ends when the intended state is accomplished

Achievement

Eli found the missing assignment.

Gyeongeun finished the paper.

- Event occurs in an instant
- Event results in a new state

Event expressions can easily be shifted to other aspectual classes!

- Surrounding context guides the interpretation of the event

Mina sailed. → Mina sailed to Paris.

Activity → Accomplishment

There are numerous temporal analysis tasks that we may want to perform.

- Popular dataset for this domain: **TimeBank**
  - American English text annotated using **TimeML**, a markup language based on interval algebra
  - TimeML includes three types of objects:
    - **Events** (representing events and states)
    - **Times** (representing temporal expressions like dates)
    - **Links** (representing relationships between events and times)
      - **TLinks** describe Allen relations
      - **ALinks** describe aspectual relationships
      - **SLinks** describe subordination relationships involving modality, evidentiality, and factuality
  - More details: https://timeml.github.io/site/timebank/documentation-1.2.html

TimeBank

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

```
<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 <TIMEX 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.
```

TimeBank

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

- Three events

```
<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 <TIMEX 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.
```

TimeBank

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

```
<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 <TIMEX 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.
```

- Three events
- Two temporal expressions

TimeBank

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

- Three events
- Two temporal expressions
- Four temporal links capturing Allen relations

```
<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.

<TLINK lid="l1" relType="IS_INCLUDED" eventInstanceID="e1" relatedToTime="t58" />
<TLINK lid="l2" relType="BEFORE" eventInstanceID="e1" relatedToTime="t57" />
<TLINK lid="l3" relType="SIMULTANEOUS" eventInstanceID="e1" relatedToEventInstance="e3" />
<TLINK lid="l4" relType="IS_INCLUDED" eventInstanceID="e1" relatedToEventInstance="e4" />
```

How can we automatically recognize and interpret the types of information present in TimeBank?

- Automated **temporal analysis** involves three common steps:
  - **Extracting temporal expressions**
  - **Normalizing these expressions** by converting them to a standard format
  - **Linking events to times** and **extracting time graphs and timelines** from the text

Extracting Temporal Expressions

- Temporal expressions may refer to absolute points in time, relative times, durations, or combinations thereof
- **Absolute temporal expressions** can be mapped directly to calendar dates, times of day, or both
- **Relative temporal expressions** can be mapped to particular times through some other reference point
  - *A week from last Tuesday*
- **Durations** can be mapped to spans of time at varying levels of granularity
  - Seconds, minutes, days, weeks, years, etc.

Example Temporal Expressions

Absolute	Relative	Duration
October 31, 2024	Yesterday	Two hours
The summer of 2024	Next semester	Three days
9:30 a.m.	Two weeks from yesterday	Four years
The fall semester in 2024	Last year	Two semesters

Challenges in Extracting Temporal Expressions

- Most people use heavily lexicalized rule-based or supervised span labeling approaches to extract temporal expressions
- This reliance on lexical triggers may also lead to false positives being tagged as temporal expressions
  - I'm reading *1984* by George Orwell
  - "*Wednesday Morning, 3 A.M.*" was Simon & Garfunkel's first album
- To avoid this issue, it is important to consider broader context as well

Temporal Normalization

- Once we recognize temporal expressions, we can **normalize** them by mapping them to specific durations or points in time
- Normalized times are represented using the **ISO 8601** standard for encoding temporal values
  - **Date:** YYYY-MM-DD
  - **Weeks:** YYYY-W*nn*, with weeks numbered from 01-53 in the year (W001 has the first Thursday of the year)
    - Note: ISO weeks begin on Monday
  - **Durations:** P*nx*, where *n* denotes the length as an integer and *x* represents the temporal unit of measurement (e.g., P3Y="three years" or P2D="two days")

Sample ISO Patterns for Times and Durations

Unit	Pattern	Sample Value
Fully specified dates	YYYY-MM-DD	2024-10-31
Weeks	YYYY-Wnn	2024-W10
Weekends	PnWE	P1WE
24-hour clock times	HH:MM:SS	09:30:00
Dates and times	YYYY-MM-DDTHH:MM:SS	2024-10-31T09:30:00

Additional Examples: https://en.wikipedia.org/wiki/ISO_8601

For our previous example....

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

```
<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.

<TLINK lid=“l1” relType=“IS_INCLUDED” eventInstanceID=“e1” relatedToTime=“t58” />
<TLINK lid=“l2” relType=“BEFORE” eventInstanceID=“e1” relatedToTime=“t57” />
<TLINK lid=“l3” relType=“SIMULTANEOUS” eventInstanceID=“e1” relatedToEventInstance=“e3” />
<TLINK lid=“l4” relType=“IS_INCLUDED” eventInstanceID=“e1” relatedToEventInstance=“e4” />
```

How can we perform temporal normalization?

- Temporal normalization is often handled using rule-based approaches that match patterns associated with different types of temporal expressions

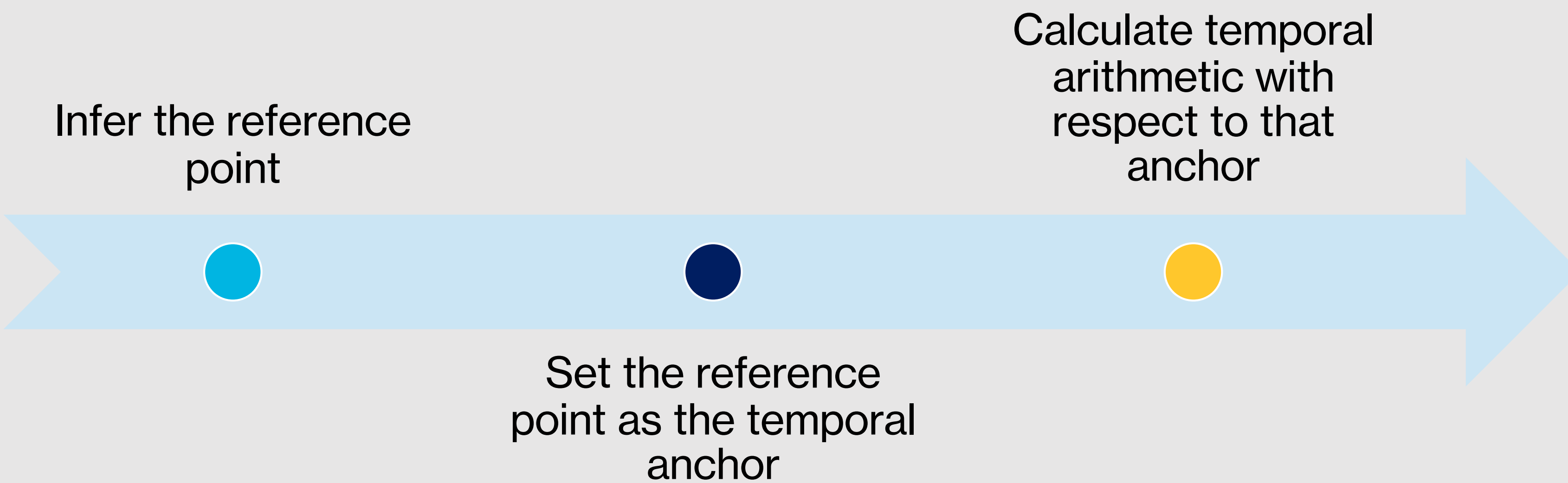
```
Pattern: /(\d+)[-\s]($TEUnits)(s)?([-\s]old)?/
Result:  Duration($1, $2)
```

- This is challenging because:
  - Fully qualified temporal expressions tend to be rare
  - Most expressions in text instead do not explicitly state a **temporal anchor**

Temporal Anchors

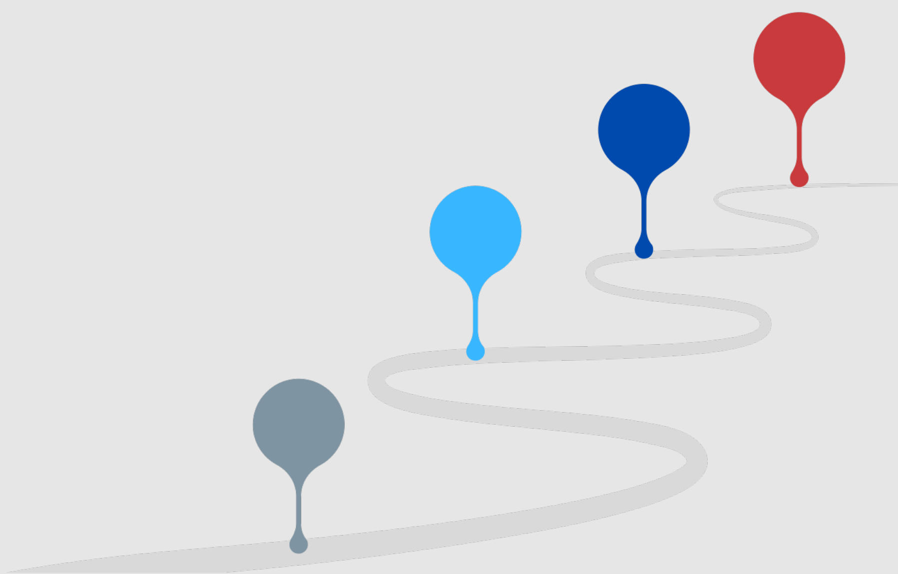
- Temporal anchors serve as the base point across which temporal expressions are normalized
  - “Today” = temporal anchor
  - “Yesterday” = temporal anchor – 1
  - “Tomorrow” = temporal anchor + 1
- **Without an explicit temporal anchor, we must infer the temporal anchor implicitly** (generally using domain-specific heuristics)
  - News articles: The temporal anchor can generally assumed to be the dateline for the news article

Relative Temporal Expressions

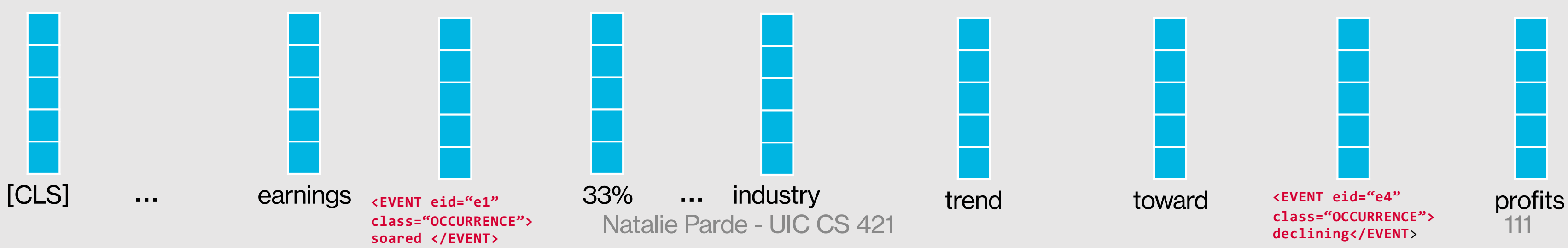


Temporal Ordering of Events

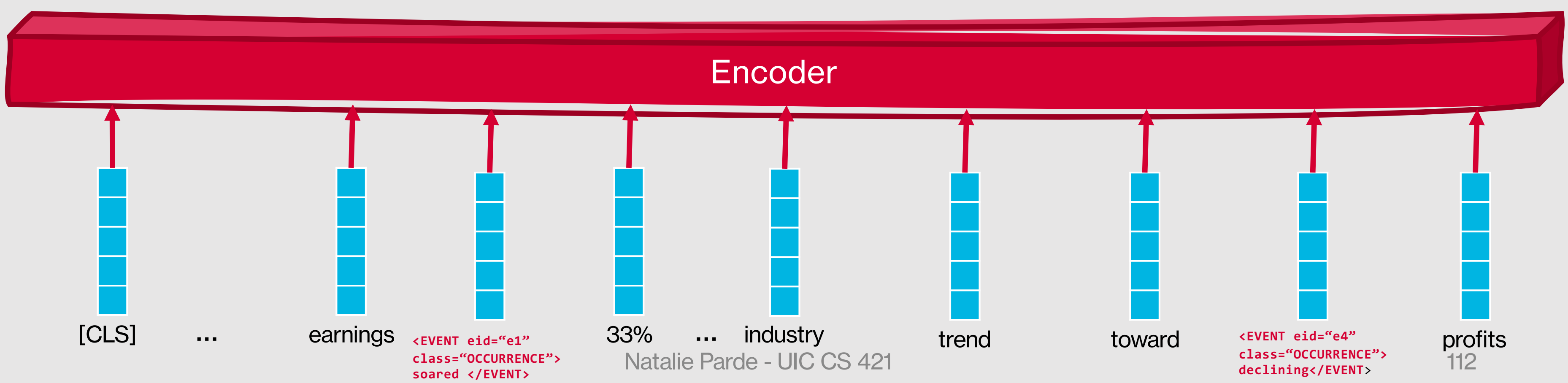
- The broader goal of extracting and normalizing temporal expressions is to be able to situate them along a timeline
- One step toward achieving this goal is to perform **temporal ordering**, such that extracted events are ordered based on the resolved temporal expressions within the text
  - Detect all events and temporal expressions from the text
  - For all possible event-event, event-time, and time-time pairs, assert links
    - Train temporal relation classifiers to predict TLinks



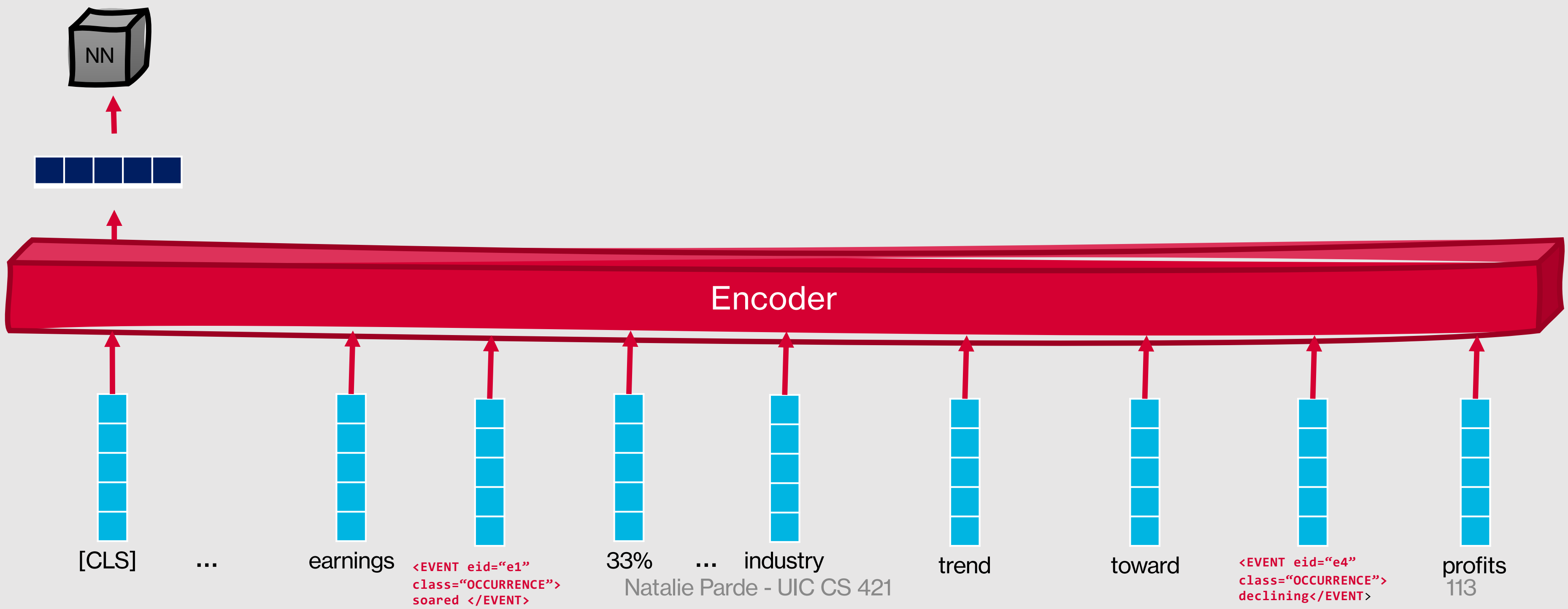
Example Neural Temporal Ordering



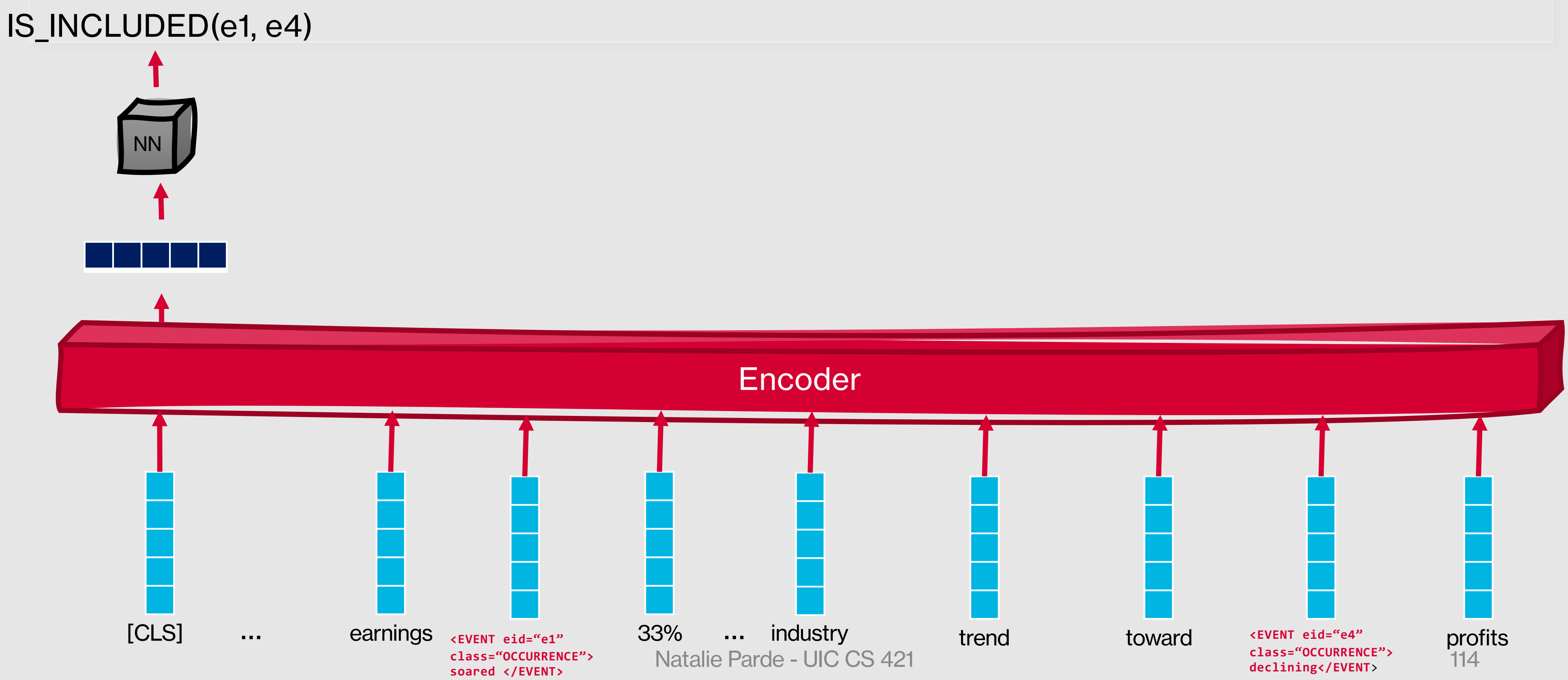
Example Neural Temporal Ordering



Example Neural Temporal Ordering



Example Neural Temporal Ordering



This Week's Topics

Named Entity Recognition
Relation Extraction
Event Extraction
Temporal Logic

Tuesday

Thursday

Temporal Analysis
Affective Lexicons
Inducing Affect and Association
Affective Analysis

Language is tricky for many reasons!

- Information extraction helps us more fully understand the meaning of unstructured text
- Another helpful tool for this: Affective analysis

Affective Analysis

- The automated analysis of the **emotions**, **moods**, **attitudes**, **stance**, or **personality** that is conveyed or evoked by a language sample
- Popular tool to facilitate social science research
  - Determining views towards a specific topic
  - Assessing public opinion
  - Interpreting intent

When can these different forms of analysis be useful?

- **Attitudes** help us figure out what people like or dislike
  - Useful for processing and interpreting reviews
  - Useful for measuring public sentiment
- **Emotions** and **moods** help us measure engagement or frustration, among other factors
  - Useful for studying how people interact with automated systems
  - Useful for psycholinguistic tasks

When can these different forms of analysis be useful?

- **Interpersonal stance** can help us understand perspectives and characteristics of multi-party interaction
  - Useful for determining views with respect to specific topics
  - Useful for summarizing conversations (is the interaction friendly or awkward?)
- **Personality** can help us customize interactive agents
  - Useful for matching user expectations
  - Useful for optimizing user experience

What is an affective lexicon?

- Known list of words corresponding to different affective dimensions
  - Optionally including scores indicating the closeness of their association with a given dimension

What are some common forms of affective analysis?

SENTIMENT ANALYSIS

EMOTION RECOGNITION

CONNOTATION FRAMING

Emotion Recognition

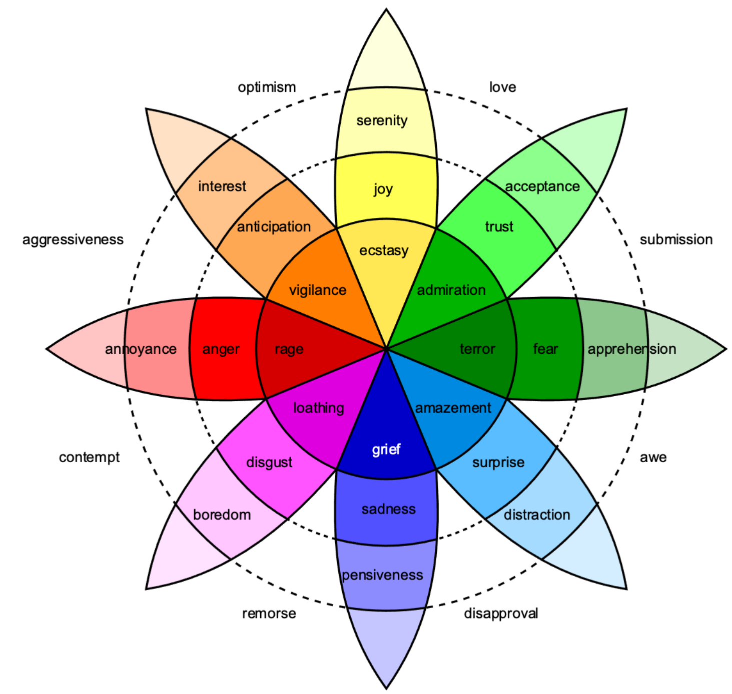
- Emotion can be defined in numerous ways
- In some frameworks, emotion is an **atomic unit**
  - IsHappy = TRUE
- In other frameworks, emotion is a **point along a multi-dimensional continuum**
  - Happiness = 0.78
- Common emotion frameworks:
  - Ekman’s six basic emotions
  - Plutchik’s wheel of emotion

Ekman's Basic Emotions

- Paper:
  - Ekman, P. (1999). Basic emotions. In T. Dalgleish & M. J. Power (Eds.), Handbook of cognition and emotion (pp. 45–60). John Wiley & Sons Ltd. https://doi.org/10.1002/0470013494.ch3
- Six basic emotions:
  - Happiness
  - Sadness
  - Anger
  - Fear
  - Disgust
  - Surprise
- Emotions are distinct from one another
- Generally known to be present across cultures

Plutchik Wheel of Emotion

- Paper:
  - Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of emotion* (pp. 3-33). Academic press. https://doi.org/10.1016/B978-0-12-558701-3.50007-7
- Situates eight basic emotions in a wheel
- Emotions located opposite one another also oppose one another semantically
- Stronger emotions derived from the basic emotions are located at more internal locations
- Weaker emotions derived from the basic emotions are located at more external locations



Atomic vs. Continuous Emotions

- Ekman and Plutchik both define emotion as an atomic unit
- Emotion along a continuum is often represented using a set of common dimensions
  - **Valence:** Pleasantness (e.g., positive or negative)
  - **Arousal:** Intensity of emotion provoked (e.g., strong or weak)
  - (Sometimes) **Dominance:** Degree of control exerted (e.g., active or passive)
- Sentiment is sometimes viewed as a measure of valence

Sentiment and Affect Lexicons

- A wide range of resources for sentiment and affect recognition are available for public use!
- Can be highly useful for performing automated sentiment or affect analysis

General Inquirer

- Classic resource created during the 1960s
  - Stone, P. J., In Kirsch, J., & Cambridge Computer Associates. (1966). *The general inquirer: A computer approach to content analysis*. Cambridge, Mass: M.I.T. Press.
- 1915 positive words
- 2291 negative words
- Additional words associated with other categories
- Information:
  - https://inquirer.sites.fas.harvard.edu/
  - https://web.archive.org/web/20110805192759/http://www.webuse.umd.edu:9090/

MPQA Subjectivity Lexicon

- Collection of positive and negative words from existing lexicons
  - 2718 positive words
  - 4912 negative words
- Additional subjective words learned via bootstrapping, with manually-provided sentiment and subjectivity levels
  - Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 347-354).
- Link:
  - https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

Opinion Lexicon

- Positive and negative words collected from product reviews via bootstrapping
  - Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 168-177).
- 2006 positive words
- 4783 negative words
- Link:
  - https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

VAD Lexicon Scores

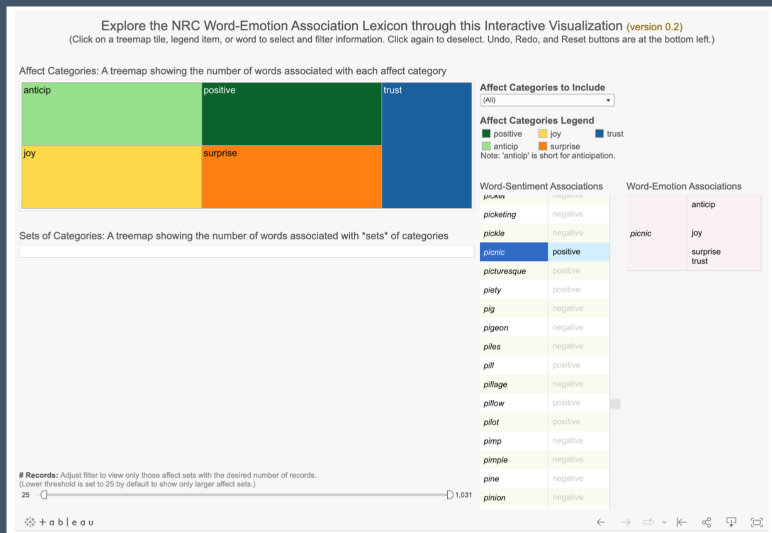
- 20,000 words labeled with valence, arousal, and dominance scores
  - Mohammad, S. (2018a). Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 174-184).
- Link: https://saifmohammad.com/WebPages/nrc-vad.html
- Translations are available in 100+ languages

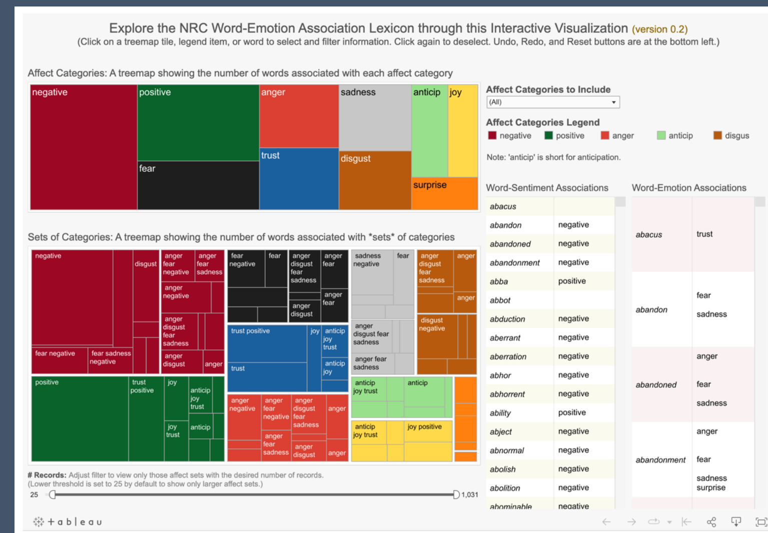
The NRC VAD Lexicon

term	valence	arousal	dominance
aaaaaaah	0.479	0.606	0.291
aaaah	0.520	0.636	0.282
aardvark	0.427	0.490	0.437
aback	0.385	0.407	0.288
abacus	0.510	0.276	0.485
abalone	0.500	0.480	0.412
abandon	0.052	0.519	0.245
abandoned	0.046	0.481	0.130
abandonment	0.128	0.430	0.202
abashed	0.177	0.644	0.307
abate	0.255	0.696	0.604
abatement	0.388	0.338	0.336

Most extreme scores for each dimension:

Dimension	Word	Score ⬆	Word	Score ⬇
valence	love	1.000	toxic	0.008
arousal	abduction	0.990	mellow	0.069
dominance	powerful	0.991	empty	0.081





NRC Word-Emotion Association Lexicon

- Approximately 14,000 words labeled for the eight basic emotions from Plutchik's wheel of emotions
  - Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3), 436-465.
- Link:
  - https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

NRC Emotion Intensity Lexicon

- Approximately 10,000 words labeled with continuous scores for the eight basic emotions from Plutchik's wheel of emotions
  - Mohammad, S. (2018b). Word Affect Intensities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Link:
  - https://www.saifmohammad.com/WebPages/AffectIntensity.htm

Word	Anger	Word	Fear	Word	Joy	Word	Sadness
outraged	0.964	*horror*	0.923	*sohappy*	0.868	*sad*	0.844
brutality	0.959	*horrified*	0.922	*superb*	0.864	*suffering*	0.844
satanic	0.828	*hellish*	0.828	*cheered*	0.773	*guilt*	0.750
hate	0.828	*grenade*	0.828	*positivity*	0.773	*incest*	0.750
violence	0.742	*strangle*	0.750	*merrychristmas*	0.712	*accursed*	0.697
molestation	0.742	*tragedies*	0.750	*bestfeeling*	0.712	*widow*	0.697
volatility	0.687	*anguish*	0.703	*complement*	0.647	*infertility*	0.641
eradication	0.685	*grisly*	0.703	*affection*	0.647	*drown*	0.641
cheat	0.630	*cutthroat*	0.664	*exalted*	0.591	*crumbling*	0.594
agitated	0.630	*pandemic*	0.664	*woot*	0.588	*deportation*	0.594
defiant	0.578	*smuggler*	0.625	*money*	0.531	*isolated*	0.547
coup	0.578	*pestilence*	0.625	*rainbow*	0.531	*unkind*	0.547
overbearing	0.547	*convict*	0.594	*health*	0.493	*chronic*	0.500
deceive	0.547	*rot*	0.594	*liberty*	0.486	*injurious*	0.500
unleash	0.515	*turbulence*	0.562	*present*	0.441	*memorials*	0.453
bile	0.515	*grave*	0.562	*tender*	0.441	*surrender*	0.453
suspicious	0.484	*failing*	0.531	*warms*	0.391	*beggar*	0.422
oust	0.484	*stressed*	0.531	*gesture*	0.387	*difficulties*	0.421
ultimatum	0.439	*disgusting*	0.484	*healing*	0.328	*perpetrator*	0.359
deleterious	0.438	*hallucination*	0.484	*tribulation*	0.328	*hindering*	0.359

Table 1: Example entries for four (of the eight) emotions in the NRC Affect Intensity Lexicon. For each emotion, the table shows every 100th and 101th entry, when ordered by decreasing emotion intensity.

Linguistic Inquiry and Word Count

- Approximately 2300 words across 73 lexical resources associated with different psychological tasks
  - Pennebaker, J. W., Booth, R. J., and Francis, M. E. (2007). *Linguistic Inquiry and Word Count: LIWC 2007*. Austin,TX.
- Link:
  - https://www.liwc.app/
- Actively maintained and updated (most recent version is from 2022)
- Not free!

RESULTS

Traditional LIWC Dimension	Your Text	Average for E-mail Language
I-words (I, me, my)	2.10	1.85
Positive Tone	2.40	2.20
Negative Tone	0.00	0.50
Social Words	8.71	6.07
Cognitive Processes	10.81	11.40
Allure	4.50	4.83
Moralization	0.00	0.07
Summary Variables		
Analytic	71.65	75.99
Authentic	80.33	36.72

Brysbaert Concreteness Lexicon

- Approximately 40,000 words labeled with continuous concreteness labels ranging from 1-5
  - Brysbaert, M., Warriner, A.B., & Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46, 904-911.
- Link:
  - http://crr.ugent.be/archives/1330

Word	Bigram	Conc.M	Conc.SD
roadsweeper	0	4.85	0.37
traindriver	0	4.54	0.71
tush	0	4.45	1.01
hairdress	0	3.93	1.28
pharmaceutics	0	3.77	1.41
hoover	0	3.76	1.23
shopkeeping	0	3.18	1.19
pushiness	0	2.48	1.24
underdevelop	0	2.37	1.4
tirelessness	0	2.28	1.28
oldfashioned	0	2.26	1.02
wellmannered	0	2.25	1.14
dismissiveness	0	1.83	1
spitefulness	0	1.8	0.76
untruthfulness	0	1.73	0.92
dispiritedness	0	1.56	0.71
sled	0	5	0
plunger	0	4.96	0.2
human	0	4.93	0.26
waterbed	0	4.93	0.27
cymbal	0	4.92	0.28
ginger	0	4.92	0.27
bobsled	0	4.9	0.41
cardboard	0	4.9	0.41
olive	0	4.9	0.31
dogsled	0	4.89	0.32

Personality and Stance

- Two other popular forms of affective analysis:
  - Personality detection
  - Stance detection
- Personality detection focuses on recognizing and classifying predefined aspects of a user's personal character
  - Most work in NLP makes use of the "Big Five" **personality dimensions**
- Stance detection focuses on recognizing a user's opinion towards a specific topic
  - Friendly
  - Distant
  - Supportive
  - Unsupportive

This Week's Topics

Tuesday

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

Semi-Supervised Induction of Affect Lexicons

- **Semi-supervised label induction:** The process of labeling new, unlabeled instances based on their similarity to instances in a small, labeled seed set
- Two main families:
  - **Axis-based** induction
  - **Graph-based** induction

Axis-Based Lexicon Induction

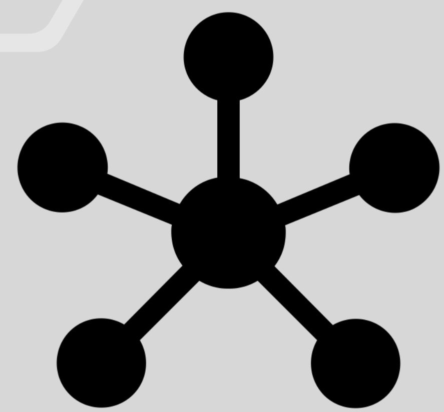
- Given a seed set, how similar is the instance to positive instances and how different is it from negative instances?
- The seed set may be:
  - Fine-tuned to the domain using induction techniques
  - Chosen manually to represent your domain

Axis-Based Lexicon Induction

- Once we've determined our seed words:
  - Compute an embedding for each seed word
  - Find the centroid of the embeddings for positive words, and the centroid of the embeddings for negative words
    - $\mathbf{V}^+ = \frac{1}{n}\sum_n^n E(w_i^+)$
  - Compute the axis by subtracting one centroid from another
    - $\mathbf{V}_{axis} = \mathbf{V}^+ - \mathbf{V}^-$
    - This produces a semantic axis vector that encodes the antonymy between the sets of words (in this case, creating a vector in the direction of the positive sentiment)
  - Compute the similarity between a given word embedding and the axis
    - $\text{score}(w) = \cos(E(w), \mathbf{V}_{axis}) = \frac{E(w)\cdot\mathbf{V}_{axis}}{\|E(w)\|\|\mathbf{V}_{axis}\|}$
  - Higher similarities indicate closer alignment with the positive class
- Paper: https://aclanthology.org/P18-1228.pdf

As an alternative....

- Graph-based induction techniques allow us to define lexicons by propagating sentiment labels on graphs

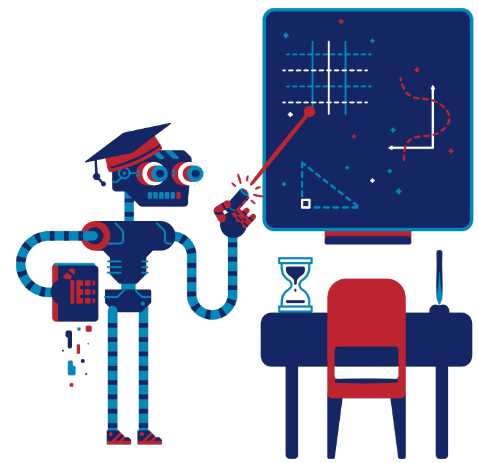


Graph-Based Lexicon Induction

- Given a graph that connects words with their nearest neighbors, how likely is it that a random walk from a positive word ends on the given word?
  - Define a graph that connects each word to its *k* nearest neighbors, with edges weighted by word similarity
  - Identify words in the graph belonging to a labeled seed set
  - Starting at a word from the seed set, perform an edge-weighted random walk
  - Assign an unlabeled word's score based on the probability of landing on it during a random walk from a positive seed and a random walk from a negative seed
    - $\text{score}^+(w_i) = \frac{\text{score}^+(w_i)}{\text{score}^+(w_i)+\text{score}^-(w_i)}$
  - Repeat multiple times using bootstrapping, and assign confidence to word scores based on their standard deviation across multiple runs

How can we use supervised machine learning to predict a word's sentiment?

- Choose an input signal:
  - One example: Review scores
    - 1-5 stars
    - Rating from 1-10
    - Often associated with free-form review text
- We can use these scores and associated text to learn polarity distributions for words



Normalized Word Likelihood

- **Document-level sentiment classifier** → any statistical or neural methods we've learned about so far!
- **Word-level sentiment classifier** → also consider simple probabilistic measures
- **Normalized word likelihood**
  - $P(w|c) = \frac{\text{count}(w,c)}{\sum_{w \in C} \text{count}(w,c)}$

Potts Diagrams

- Mechanism for visualizing word sentiment
  - Sentiment class vs. normalized word likelihood
- Characteristic patterns:
  - **"J" shape:** Strongly positive word
  - **Reverse "J" shape:** Strongly negative word
  - **"Hump" shape:** Weakly positive or negative word
- Patterns may also correspond to different types of word classes
  - Emphatic and attenuating adverbs

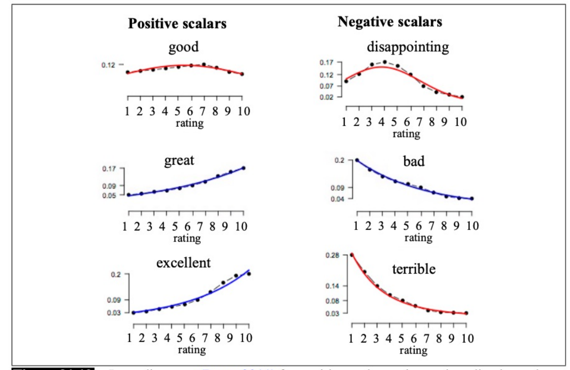


Figure 21.10 Potts diagrams (Potts, 2011) for positive and negative scalar adjectives, showing the J-shape and reverse J-shape for strongly positive and negative adjectives, and the hump-shape for more weakly polarized adjectives.

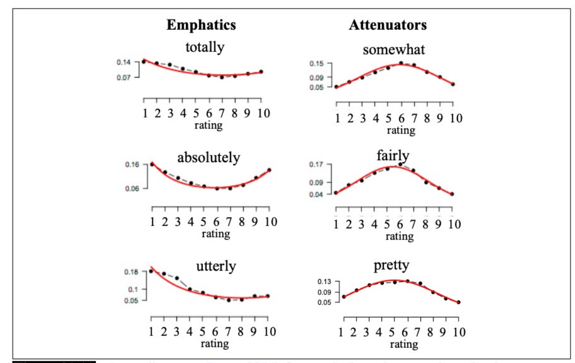


Figure 21.11 Potts diagrams (Potts, 2011) for emphatic and attenuating adverbs.

Log Odds Ratio with an Informative Dirichlet Prior

- Allows us to determine which words are closely associated with different classes
- Originally proposed for measuring partisan speech used by US politicians
- Generalizes to any other problem domain for which lexical trends are anticipated to be different between groups
- Key goal: Find words that are statistically overrepresented in one category of text compared to another

Start with a simple log odds ratio....

- **Probability of word *w* existing in a subset of words *i*:**
  - $P^i(w) = \frac{f_w^i}{n^i}$

Total number of words in *i*

- **Log odds ratio for word *w* in the subset of words *i* versus the subset of words *j*:**
  - $\mathrm{lor}(w) = \log\frac{P^i(w)}{1-P^i(w)} - \log\frac{P^j(w)}{1-P^j(w)} = \log\frac{f_w^i}{n^i-f_w^i} - \log\frac{f_w^j}{n^j-f_w^j}$

- However, this doesn't tell us anything about what we'd expect to see!
- Dirichlet intuition: Use a large background corpus to get a prior estimate of our expected frequency for each word *w*
  - To do so, add the counts from that corpus to our numerator and denominator

Prior-Modified Log Odds Ratio

- Modifying the previous equation with an informative Dirichlet prior:
  - $\delta_w^{(i-j)} = \log \frac{f_w^i + \alpha_w}{n^i + \alpha_0 - (f_w^i + \alpha_w)} - \log \frac{f_w^j + \alpha_w}{n^j + \alpha_0 - (f_w^j + \alpha_w)}$

Number of words in *i*

Count of *w* in corpus

Size of background corpus

Count of w in background corpus

Log Odds Ratio with Informative Dirichlet Prior

- Estimate of variance for the modified log odds ratio:
  - $\sigma^2\left(\hat{\delta}_w^{(i-j)}\right) \approx \frac{1}{f_w^i+\alpha_w} + \frac{1}{f_w^j+\alpha_w}$
  - Controls for the amount of variance in a word's frequency
- Final statistic for a word is then the z-score of its modified log odds ratio:
  - $\frac{\hat{\delta}_w^{(i-j)}}{\sqrt{\sigma^2\left(\hat{\delta}_w^{(i-j)}\right)}}$

This ultimately gives us a useful tool for analysis!

Words associated with one-star restaurant reviews

worst, rude, terrible, horrible, bad, awful, disgusting, bland, tasteless, gross, mediocre, overpriced, worse, poor

Words associated with five-star restaurant reviews

Great, best, love, delicious, amazing, favorite, perfect, excellent , awesome, friendly, fantastic, fresh, wonderful, incredible, sweet, yum

This Week's Topics

Tuesday

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal Logic

Thursday

- Temporal Analysis
- Affective Lexicons
- Inducing Affect and Association
- Affective Analysis

How can we perform affect recognition using supervised machine learning?

- Smaller datasets:
  - N-gram features
- Larger datasets:
  - N-gram features, pruned based on frequency or **pointwise mutual information (PMI)**
    - $\mathrm{PMI}(x; y) = \log \frac{p(x,y)}{p(x)p(y)}$
- Even larger datasets:
  - Word embeddings
- In all cases, we can support performance using features derived from external lexicons

What if we don't have labeled training data to build a supervised model for affect recognition?

Use sentiment resources (such as those already described!) to perform sentiment analysis directly

Using Lexicons for Sentiment Recognition

1. Assign a positive label to instances that contain more positive than negative words from the lexicon
2. Assign a negative label to instances that contain more negative than positive words
3. Assign a neutral label in the event of a tie

More formally....

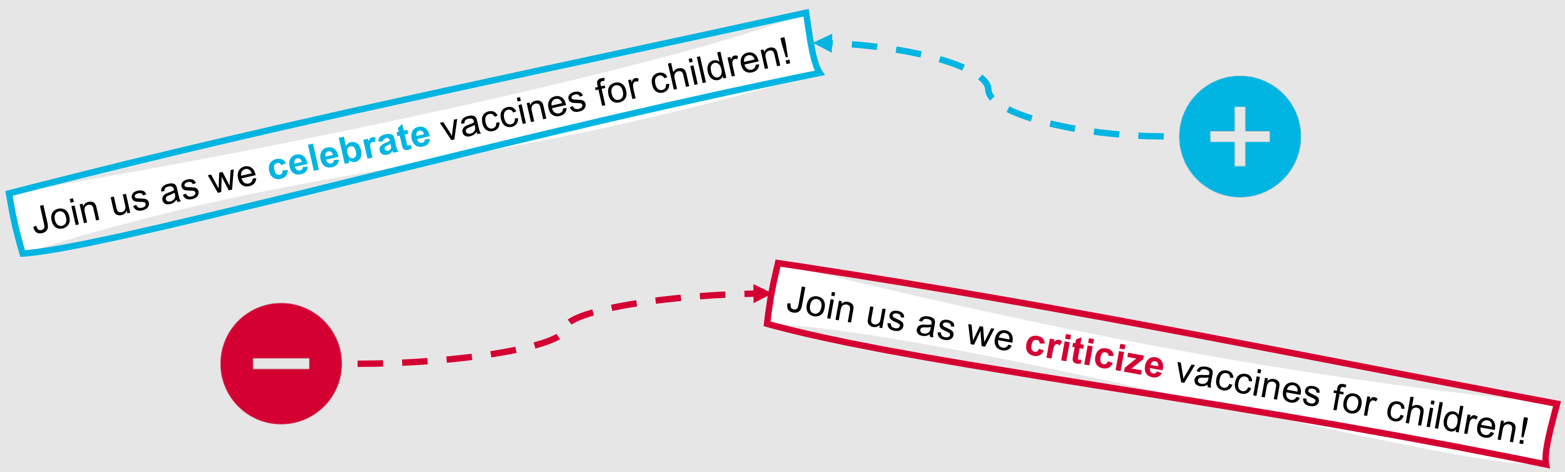
- Define a threshold λ indicating the minimum percentage of positive or negative words needed for a positive or negative classification
- Select a sentiment class as follows:
  - $f^+ = \sum_{w \in x} \theta_w^+ \text{count}_+(w)$
  - $f^- = \sum_{w \in x} \theta_w^- \text{count}_-(w)$
  - $\text{sentiment} = \begin{cases} + \text{ if } \frac{f^+}{f^-} > \lambda \\ - \text{ if } \frac{f^-}{f^+} > \lambda \\ 0 \text{ otherwise} \end{cases}$

Entity-Centric Affect

- Sometimes we don't need (or want) to recognize the affect of an entire input
  - Scope may be too broad!
- We can also learn to predict the affect of a single entity within the input
- One way to do this: Leverage both affect lexica and contextual word embeddings
  - Average the contextual embeddings for each instance of a word
  - Repeat this process for all words
  - Learn to map from averaged word embeddings to affective scores
  - When a new entity is encountered without a known affective score:
    - Create a new average embedding based on all instances of that entity in context
    - Predict a score for that embedding

Connotation Frames

- We can also represent affective meaning using **connotation frames**
  - Indicate affective properties commonly associated with words, similarly to how verb frames indicate selectional preferences

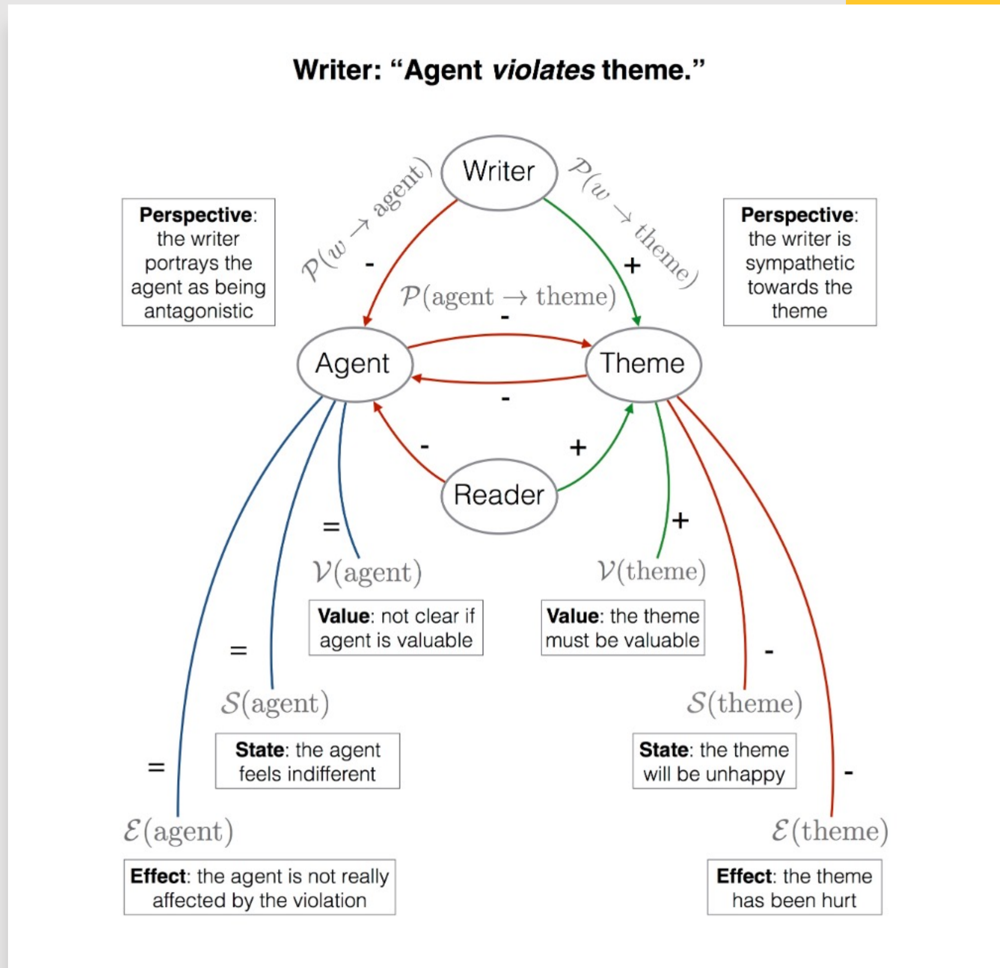


Connotation Frames

- Formalism for analyzing subjective roles and relationships implied by a predicate
- Contain labels for relationships that are inferable from the predicate:
  - Writer’s perspective
  - Reader’s perspective
  - Entity’s perspective
  - Entity’s value
  - Entity’s mental state
  - Effect on entity
- Can be:
  - Manually constructed
  - Learned automatically
- Downloadable collection of connotation frames:
  - https://hrashkin.github.io/connframe.html

Example Connotation Frame

- Relevant papers:
  - Hannah Rashkin, Sameer Singh, Yejin Choi. 2016. Connotation Frames: A Data-Driven Investigation. In Proceedings of ACL 2016.
    - https://aclanthology.org/P16-1030/
  - Maarten Sap, Marcella Cindy Prasettio, Ari Holtzman, Hannah Rashkin, & Yejin Choi. 2017. Connotation Frames of Power and Agency in Modern Films. In Proceedings of EMNLP 2017 Short Papers.
    - https://aclanthology.org/D17-1247/
- Relevant resource for measuring social dynamics between personas: https://github.com/maartensap/riveter-nlp



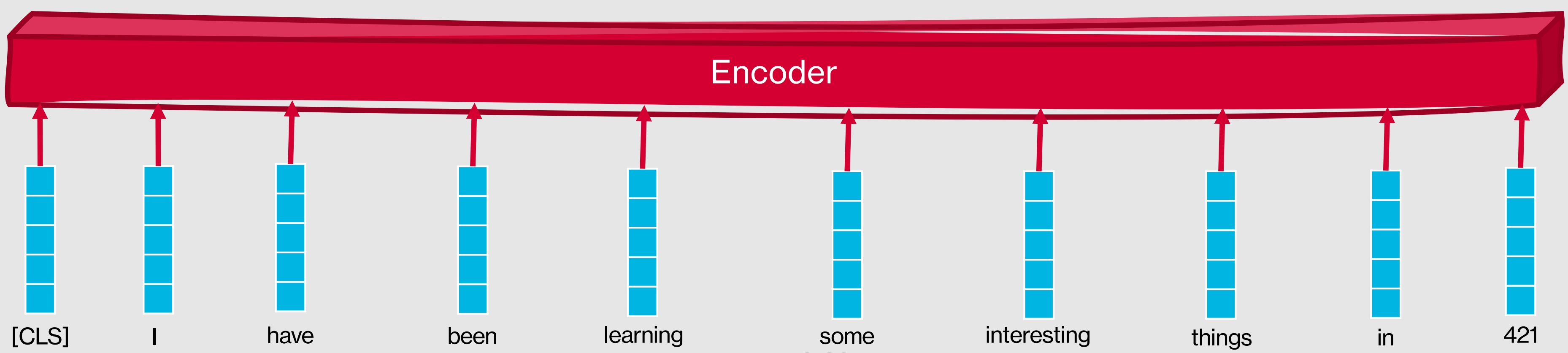
Stance Detection

- The task of determining sentiment with respect to a specific target (often a polarizing topic)
- Stance labels are generally some variation of **favor** and **against**
- Numerous datasets exist for this task:
  - **SemEval-2016 Task 6A Stance Dataset**
    - 4870 tweets manually annotated for stance with respect to: "Atheism," "Climate Change is Real Concern," "Feminist Movement," "Hilary Clinton," and "Legalization of Abortion"
  - **Multi-Perspective Consumer Health Query Data**
    - Relevant sentences from the top 50 links corresponding to common, polarizing or widely debated public health queries (e.g., "Does the MMR vaccine lead to autism in children?")
  - **Ideological Online Debates**
    - Online debates on "Existence of God," "Healthcare," "Gun Right," "Gay Rights," and "Abortion and Creationism"

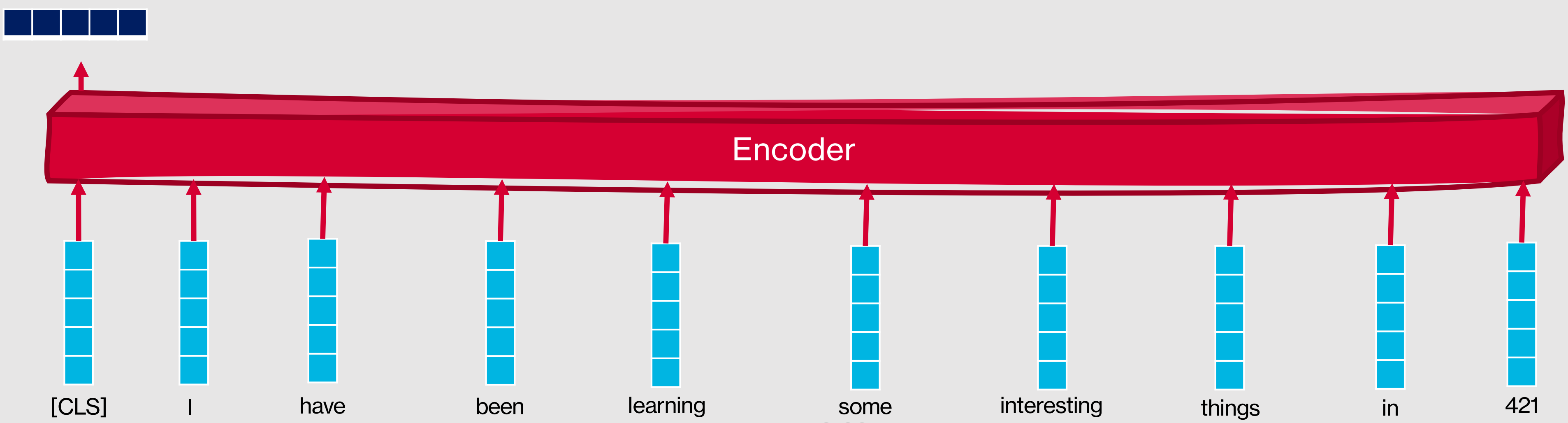
How can we build stance detection models?

- Predict favor and against (and optionally neutral) labels for each target *or* determine the relevant target as a preliminary step
- Feature-based approaches:
  - Text content
  - (If known) user-specific attributes
  - (If known) network-specific attributes
- Neural approaches:
  - Typically framed as a supervised instance-level classification task

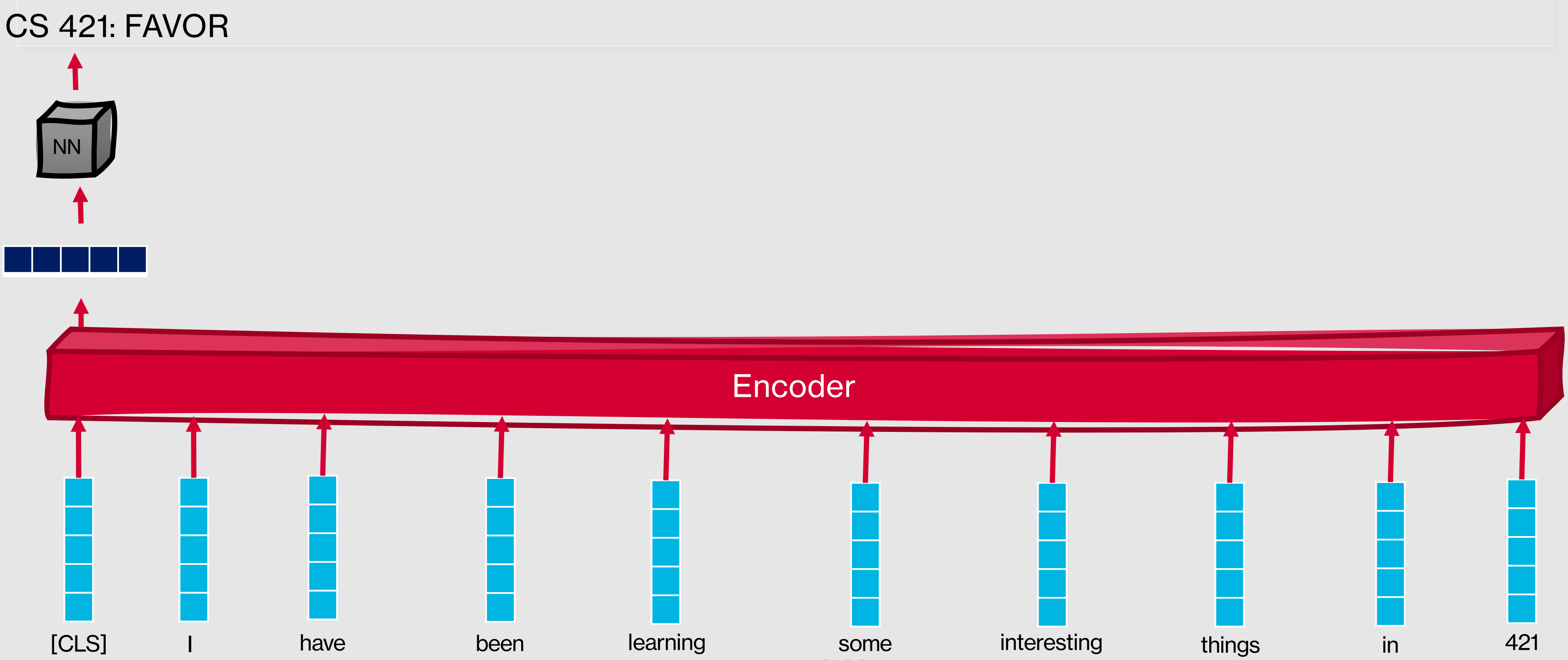
Example: Stance Detection



Example: Stance Detection



Example: Stance Detection



Summary: Temporal and Affective Analysis

- **Aspect** defines categories of events or states based on their temporal structure
- Approaches for **temporal analysis** focus on extracting temporal expressions, normalizing those expressions, and using the extracted and normalized temporal information to place events in a structured order
- Lexicons can help us distinguish many kinds of **affective states**
- Emotion can be represented using fixed **atomic units** or **dimensions** in a continuous space
- **Affective lexicons** can be built by hand, in a semi-supervised manner, or using fully supervised methods
- Words can be assigned weights in a lexicon based on frequency measures and ratio metrics like **log odds ratio with an informative Dirichlet prior**
- **Connotation frames** express richer affective relationships, similar to those seen with semantic frames
- **Stance detection** allows us to predict sentiment with respect to a specific target