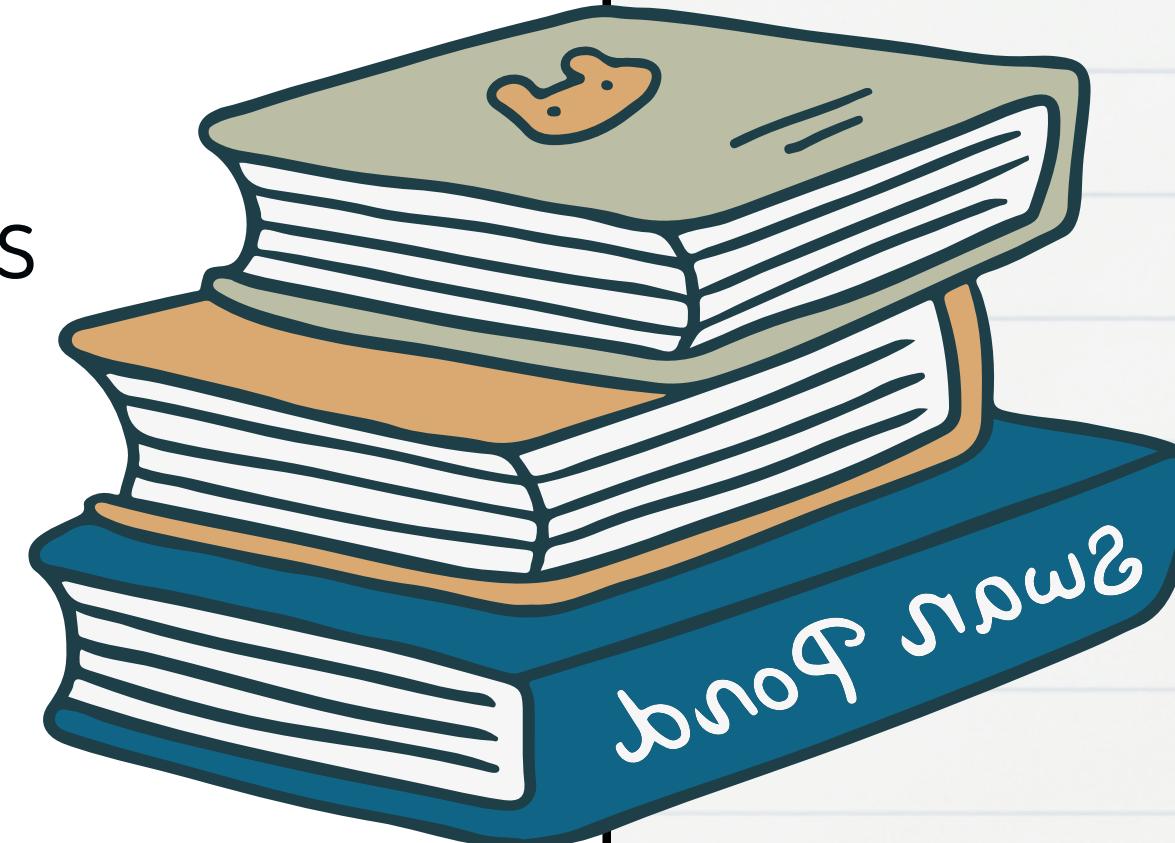


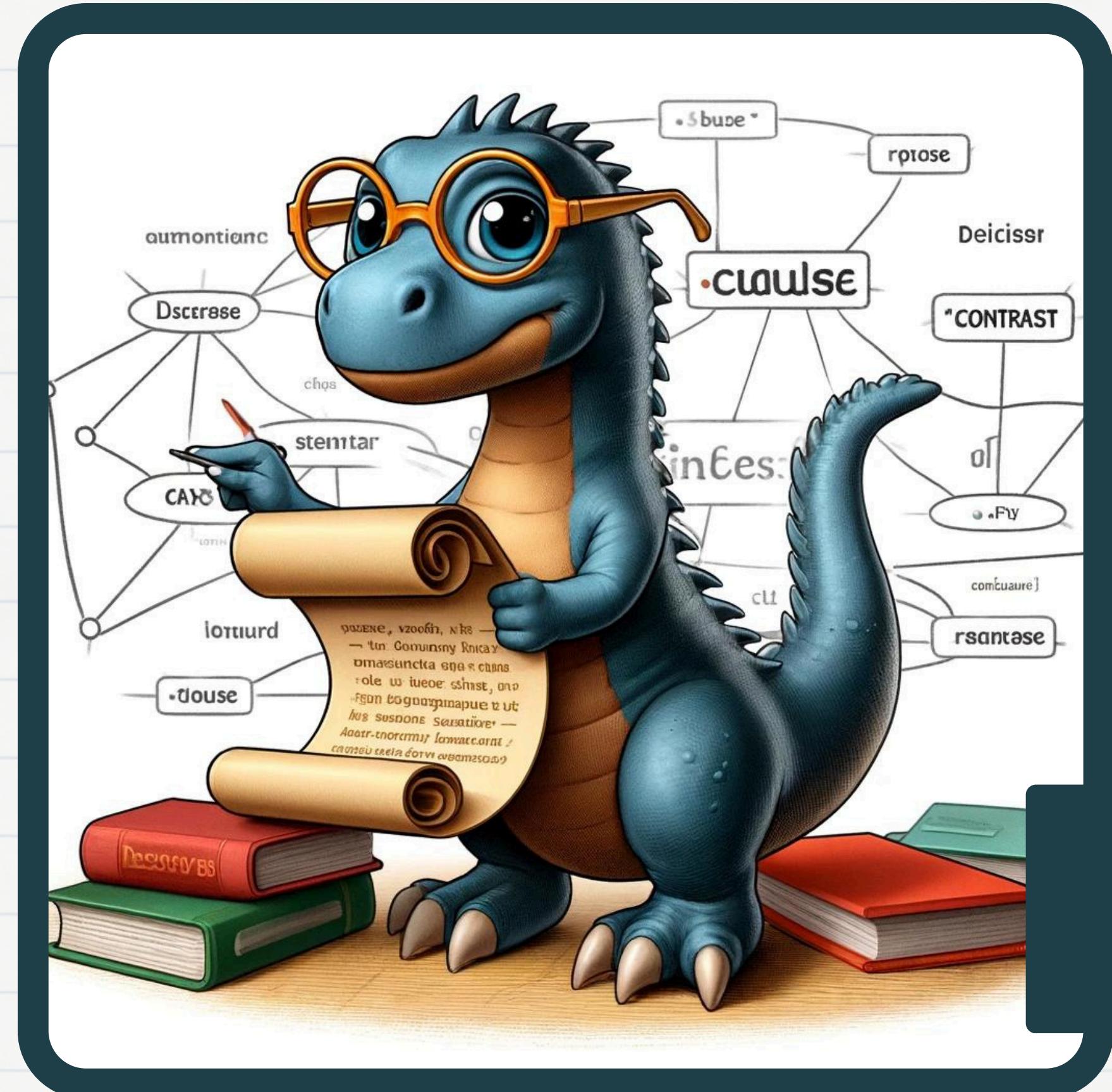
Discourse & Document Level Annotation

Presented by: Benét Post

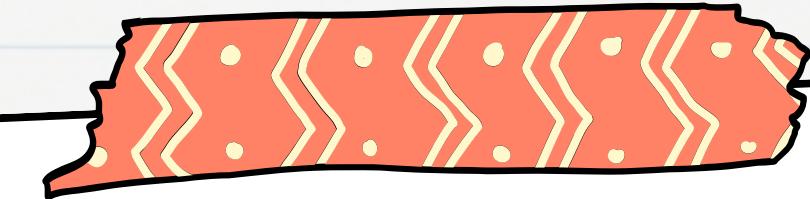
Overview

1. Discourse Relations
2. Coreference
3. Temporal Dependencies
4. Modal Dependencies





Discover
the
Discourse
Dinosaur



Discourse Relations

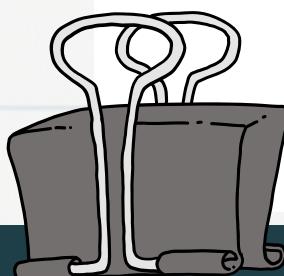
Overview

- Discourse relations describe **semantic connections** between **events** in complex sentences.
- UMR provides a **lattice** of category values to annotate these relations, drawing from typological research and partly on adverbial subordination/coordination patterns in English.
- The lattice focuses on relations expressed through **conjunctions** or **dedicated verb forms**, excluding those typically expressed by optional discourse adverbs.



The Lattice!

Discourse
Relations



and +
but

and +
unexpected

and + contrast

but
(but-91)

and

Disjunctive
(or)

Exclusive Disjunctive
(exclusive-disj)
Inclusive Disjunctive
(inclusive-disj)

Consecutive
(consecutive)

Additive
(additive)

Unexpected Co-occurrence
(unexpected-co-occurrence-91)

Pure Contrast
(contrast-91)

Apprehensive
(:apprehensive)

Purpose
(:purpose)

Means
(:manner)

Cause
(:cause)

Conditional
(:condition)

Posterior/Anterior/
Simultaneous
(:temporal)

Pure Addition
(:pure-addition)

Substitution
(:substitute)

Concessive
(:concession)

Concessive conditional
(:concessive-condition)

Subtraction
(:subtraction)

Lattice: Fine vs Course Grained

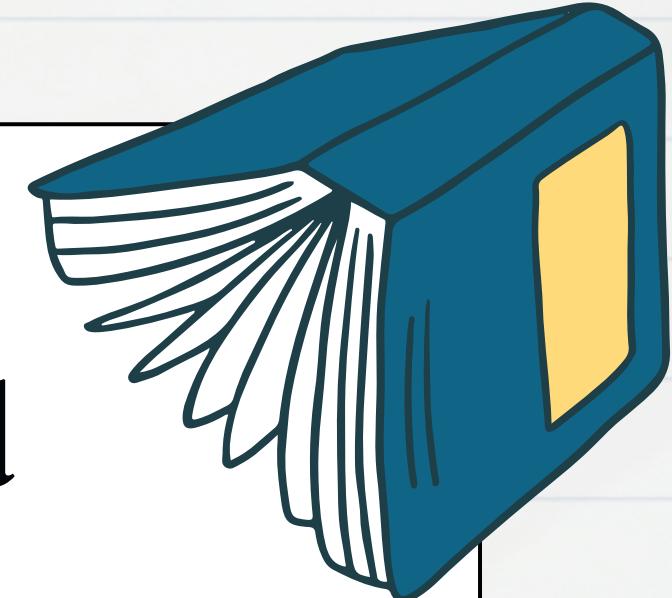
Fine Grained

- Annotators can choose more fine grained values when they feel **confident**
- (Adverbial) **subordination constructions** express overtly fine-grained structures

Course Grained

- Annotators can choose course-grained values when in **doubt**
- **Coordination constructions** are more polysemous and often require more course-grained categories

What are
adverbial
subordination
and
coordination
constructions?



Constructions

I will call you when I arrive

She went home early because she felt sick

Adverbial Subordinating

- Involve linking a **main clause** with 1+ **subordinate clauses**
- The subordinate clause **depends** on the main clause and **cannot stand alone** as a complete sentence
- In English: because, although, when, while, if

Fine Grained

I went to the store and I bought food

He's tired, but he cannot sleep

Coordination

- Involve **linking 2+ clauses** of equal semantic importance
- Each clause **can stand alone** as a complete sentence
- In English: and, but, or, so, not, for, yet

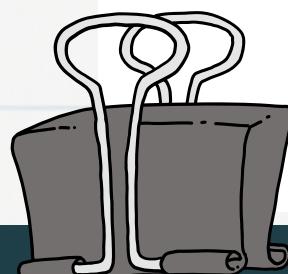
Course Grained

Constructions

Discourse Relations

Subordinating

Coordinating



Disjunctive
(or)

Exclusive Disjunctive
(exclusive-disj)

Inclusive Disjunctive
(inclusive-disj)

Apprehensive
(:apprehensive)

and

and +
unexpected

and + contrast

but
(but-91)

Consecutive
(consecutive)

Additive
(additive)

Unexpected Co-occurrence
(unexpected-co-occurrence-91)

Pure Contrast
(contrast-91)

Purpose
(:purpose)

Means
(:manner)

Cause
(:cause)

Conditional
(:condition)

Posterior/Anterior/
Simultaneous
(:temporal)

Pure Addition
(:pure-addition)

Substitution
(:substitute)

Concessive
(:concession)

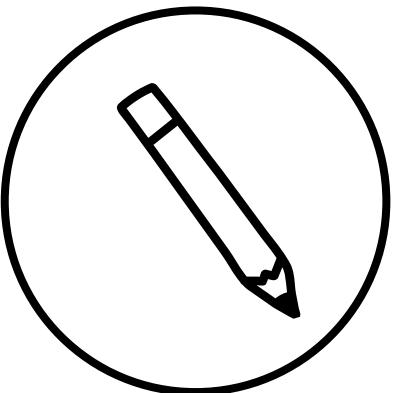
Concessive conditional
(:concessive-condition)

Subtraction
(:subtraction)

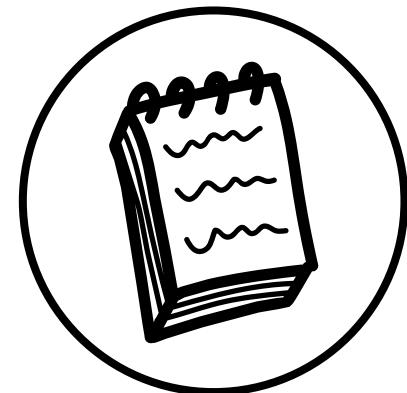
Activity 1



In the next few slides I am going to tell you about each of the discourse relation types.



If you think of a discourse relation that fits to a word or concept in a language that you speak write it down in Box 1. Also add details to the lattice in Part 2.



We will share at the end of this section!



Disjunctive



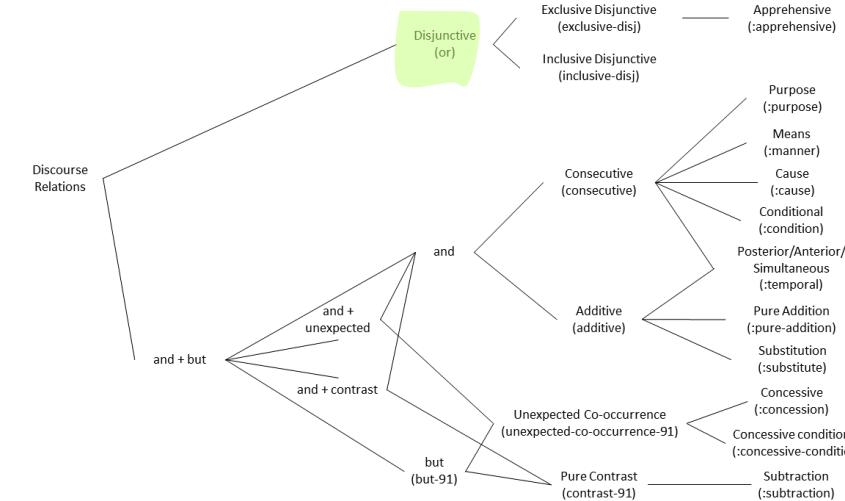
Definition: Events are alternatives

UMR Concept: 'or' with ':opX'
arguments for each 'or' item

English Equivalent: either, or

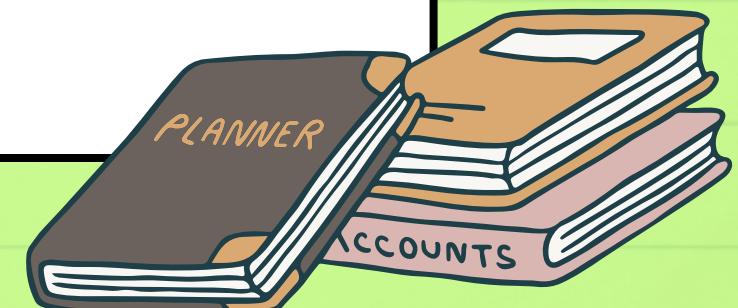
Subtypes:

- Inclusive Disjunctive
- Exclusive Disjunctive



I will go for a walk or play some soccer.

(o/ or
:op1 (w/ walk-01
:ARG0 (p/ person
:ref-person 1st
:ref-number Singular)
:aspect Process
:modstr FullAff))
:op2 (p2/ play-01
:ARG0 p
:ARG1 (s/ soccer)
:aspect Process
:modstr FullAff))



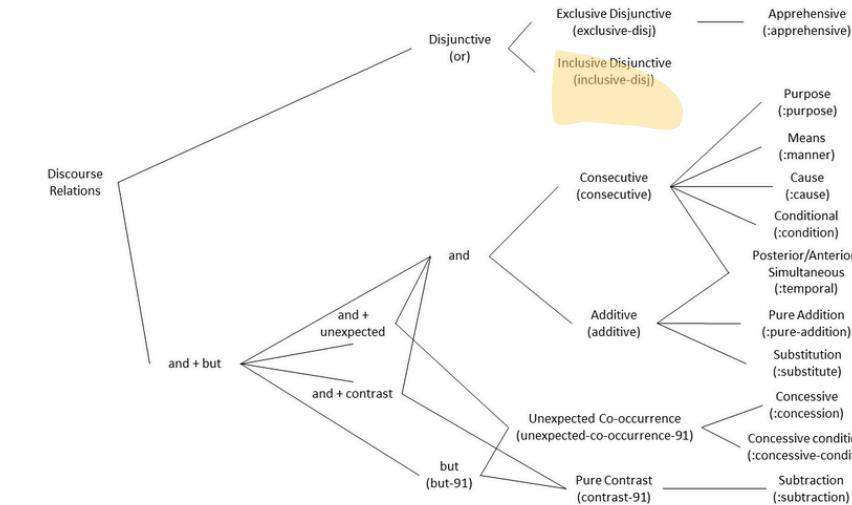
Inclusive Disjunctive



Definition: Any or all alternatives are possible (not mutually exclusive)

UMR Concept: 'inclusive-disj'

English Equivalent: not very close equivalent. In this sentence it is '-ve'



```
mnu'bo      hatai-supi'ba-ve      kire'bo kro-de-supi'ba-ve      egemo  bre-supi'ba-ve      degi   ki
pandanus    bash-PURP.1PL-NONEX  corn    husk-eat-PURP.1PL-NONEX banana plant-PURP.1PL-NONEX crazy   pl
'If we go down to a 'crazy place' to bash pandanus, husk and eat corn, or plant bananas, we don't sleep with women
(h/ have-condition-91
:ARG1 (a/ a'vogune-00 'sleep'
       :actor (p/ person
              :ref-person 1st
              :ref-number Plural)
       :companion (a2/ aki' 'woman')
       :aspect Habitual
       :modstr FullNeg)
:ARG2 (r/ rmisupamo-00 'go down'
       :actor p
       :goal (k/ kikopi' 'place'
              :mod (d/ degi 'crazy'))
       :purpose (i/ inclusive-disj
                  :op1 (h2/ hataisupi'bave-00 'bash'
                        :actor p
                        :undergoer (m/ mnu'bo 'pandanus')
                        :aspect Performance
                        :modstr FullAff)
                  :op2 (k2/ krodesupi'bave-00 'husk and eat'
                        :actor p
                        :undergoer (k3/ kire'bo 'corn')
                        :aspect Performance
                        :modstr FullAff)
                  :op3 (b/ bresupi'bave-00 'plant'
                        :actor p
                        :undergoer (e/ egemo 'banana')
                        :aspect Performance
                        :modstr FullAff)
       :aspect Performance
       :modstr FullAff
       :aspect State)
```



Exclusive Disjunctive



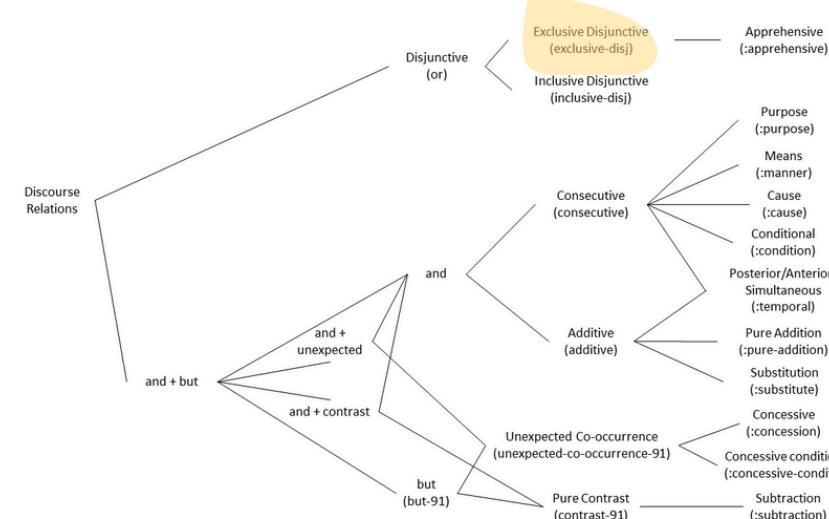
Definition: Mutually exclusive options

UMR Concept: 'exclusive-disj'

English Equivalent: not very close equivalent, in this sentence, it is 'ito'

Subtypes:

- Apprehensive



bai-ve ito 'a'-bai-e
be.3SG-INTERR EX NEG-be.3SG-IND
'Is he here or isn't he?'
(e/ exclusive-disj)
:op1 (h/ have-place-91
:ARG1 (p/ person
 :ref-person 3rd
 :ref-number Singular)
:ARG2 (p2/ place)
:aspect State
:modstr NeutAff
:mode Interrogative)
:op2 (h2/ have-place-91
:ARG1 p
:ARG2 p2
:aspect State
:modstr NeutNeg
:polarity -
:mode Interrogative))



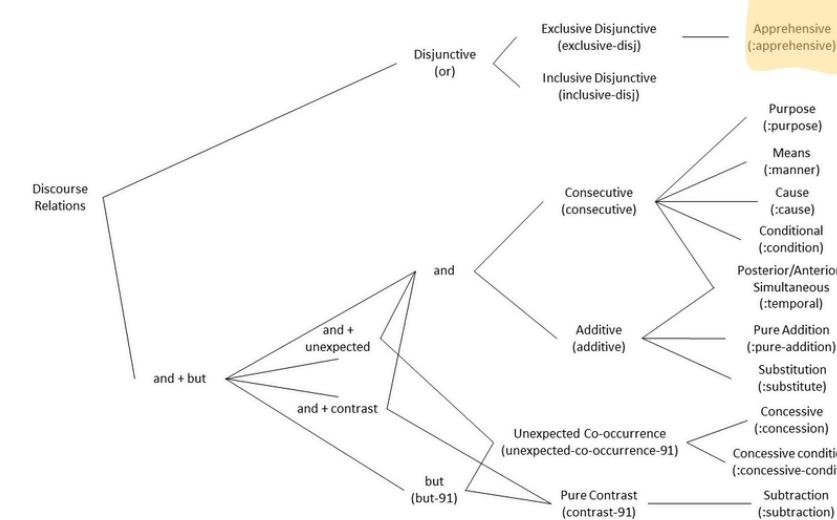
Apprehensive



Definition: One event is carried out with intention of preventing other

UMR Concept: ‘apprehensive’

English Equivalent: lest



Grab a stick **lest** he attack you!

(g/ grab-01

:ARG0 (p/ person

:ref-person 2nd

:ref-number Singular)

:ARG1 (s/ stick

:ref-number Singular)

apprehensive (a/ attack-01

:ARG0 (p2/ person

:ref-person 3rd

:ref-number Singular)

:ARG1 p

:aspect Performance

:modstr FullAff)

:mode Imperative

:aspect Performance

:modstr PrtAff)



Additive



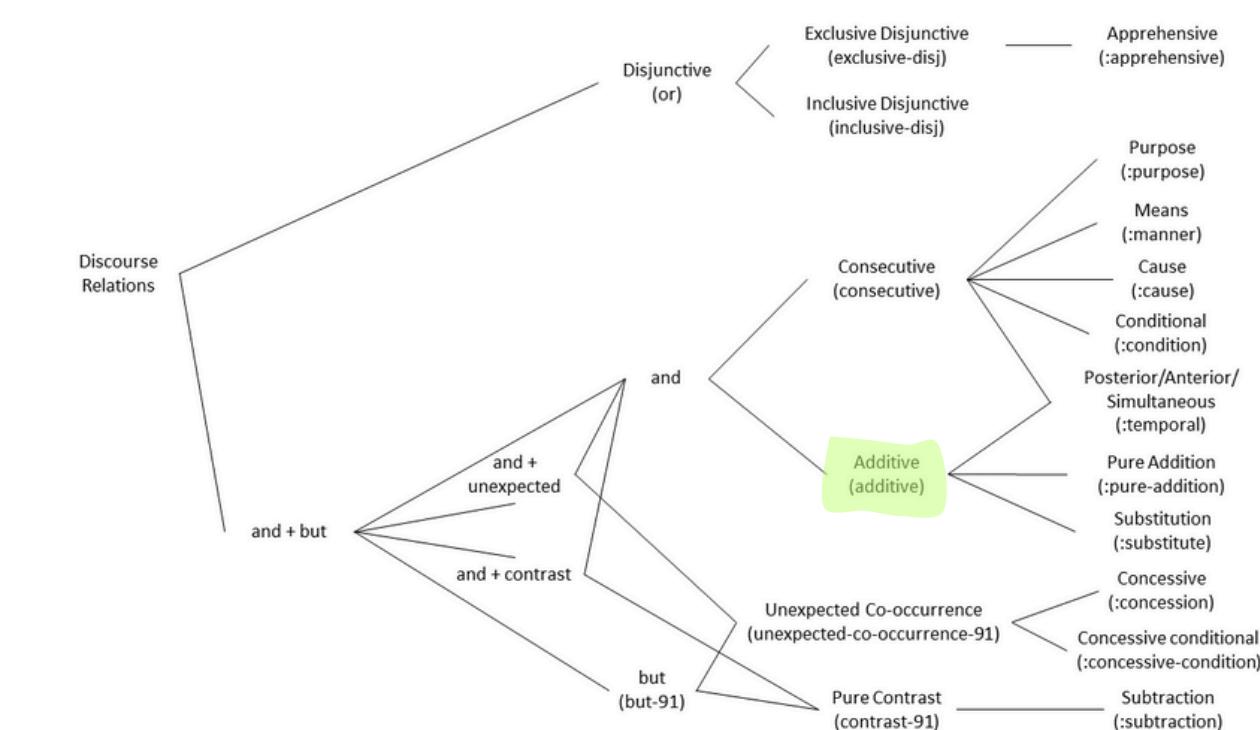
Definition: addition of events

UMR Concept: 'additive' with ':opX' roles

English Equivalent: and

Subtypes:

- Simultaneous
- Pure Addition
- Substitution



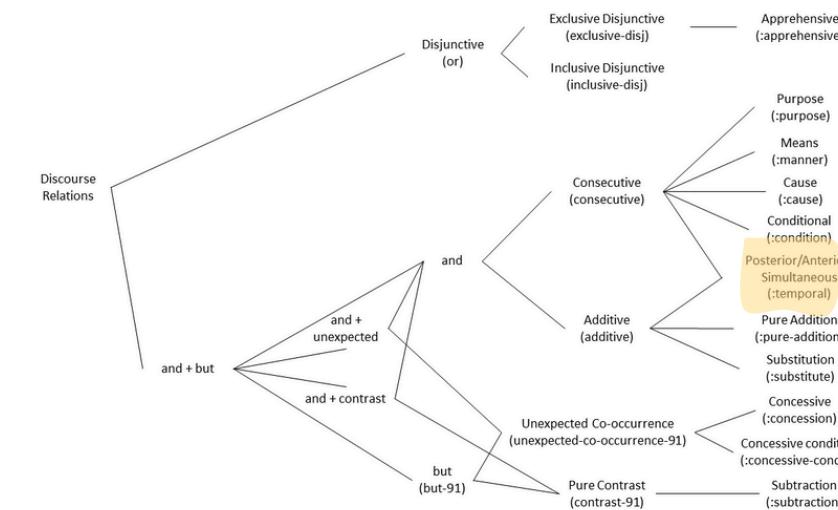
Simultaneous



Definition: temporal overlap

UMR Concept: ':temporal'

English Equivalent: while, and



I read a book **while** I listened to music.

(r/ read-01

:ARG0 (p/ person

:ref-person 1st

:ref-number Singular)

:ARG1 (b/ book

:ref-number Singular)

:temporal (listen-01

:ARG0 p

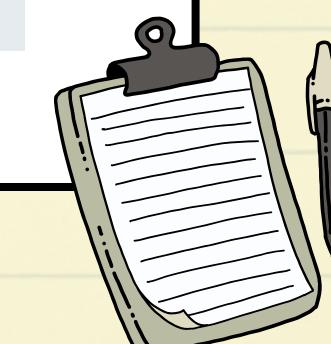
:ARG1 (m/ music)

:aspect State

:modstr FullAff)

:aspect Performance

:modstr FullAff)



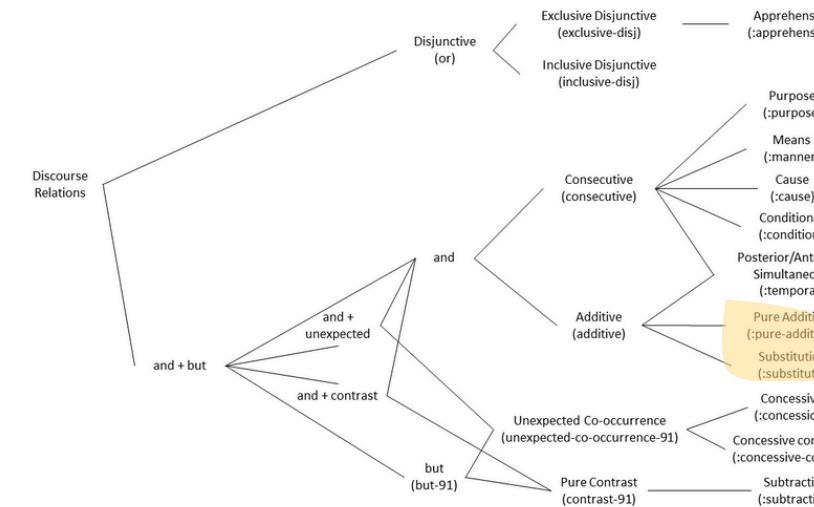
(pure) Addition



Definition: Events cannot occur separately in context of utterance

UMR Concept: ':pure-addition'

English Equivalent: and



In addition to having your hand stamped,
you have to show your ticket to get into the concert.

(h/ have-04
:ARG0 (p/ person
:ref-person 2nd
:ref-number Singular)
:ARG1 (s/ stamp-01
:ARG0 (p2/ person)
:ARG1 (h/ hand
:part p)
:aspect Performance
:modstr PrtAff)
:aspect Performance
:modstr PrtAff
:pure-addition (s2/ show
:ARG0 p
:ARG1 (t/ ticket
:poss p
:ref-number Singular)
:ARG2 (p3/ person)
:aspect Performance
:modstr PrtAff)
:purpose (g/ get-05
:ARG0 p
:ARG2 (c/ concert
:ref-number Singular)
:aspect Performance
:modstr FullAff))



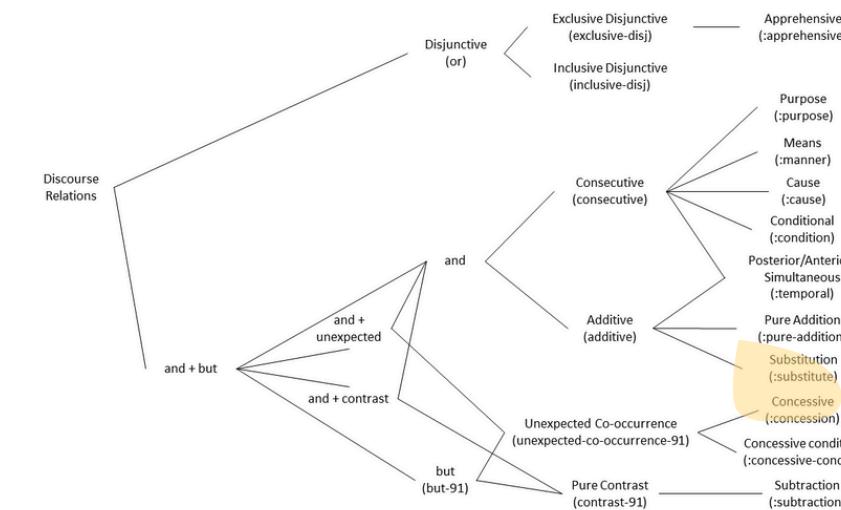
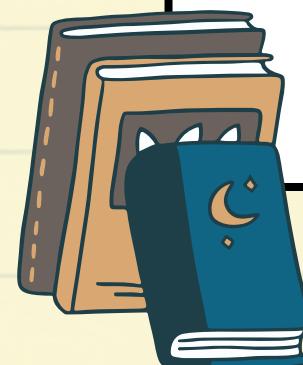
Substitution



Definition: One event replaces another

UMR Concept: ':substitute'

English Equivalent: instead



Instead of going out to eat, we barbecued chicken at home.
(b/ barbecue-01) Or (a.
:ARG0 (p/ person
:ref-person 1st
:ref-number Plural)
:ARG1 (c/ chicken)
:location (h/ home)
:substitute (g/ go_out-17
:ARG0 p
:purpose (e/ eat-01
:ARG0 p
:aspect Endeavor
:modstr FullAff)
:aspect Performance
:modstr FullNeg
:aspect Performance
:modstr FullAff



Consecutive



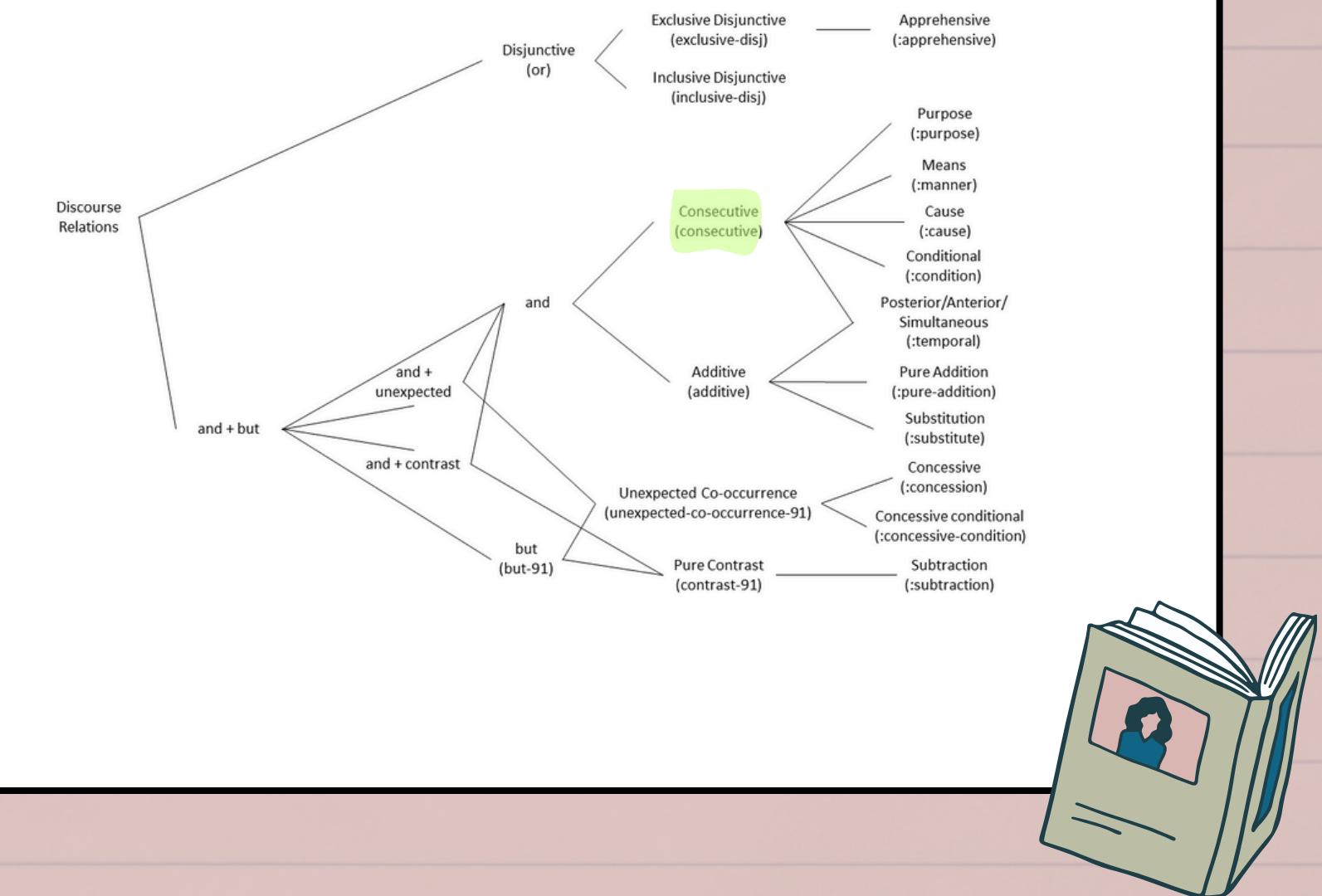
Definition: Events with temporal/logical sequencing

UMR Concept: 'consecutive' with ':opX' roles

English Equivalent: and, in order to

Subtypes:

- Purpose • Conditional
- Means • Anterior
- Causal • Posterior



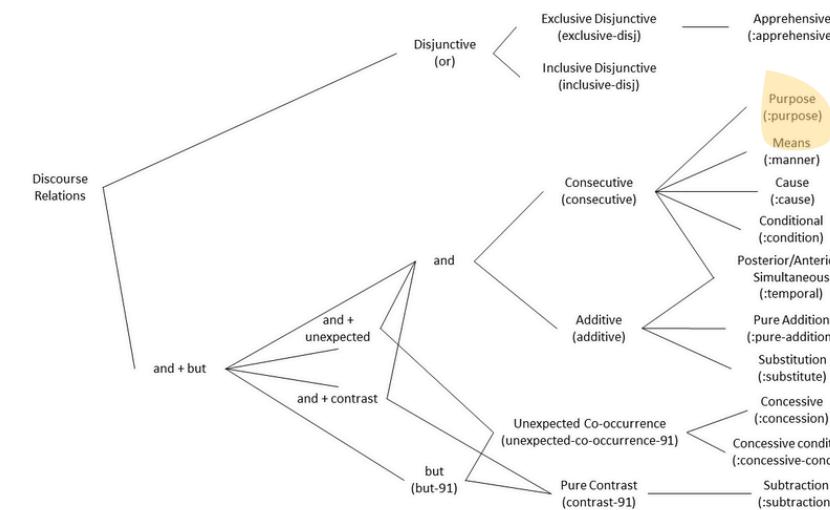
Purpose



Definition: Intention to bring about another event

UMR Concept: :purpose

English Equivalent: so that



I grabbed a stick **in order to defend myself.**
(g/ grab-01)

:ARG0 (p/ person
:ref-person 1st
:ref-number Singular)
:ARG1 (s/ stick
:ref-number Singular)
:purpose (d/ defend
:ARG0 p
:ARG1 p
:aspect Performance
:modstr FullAff)
:Aspect Performance
:modstr FullAff)



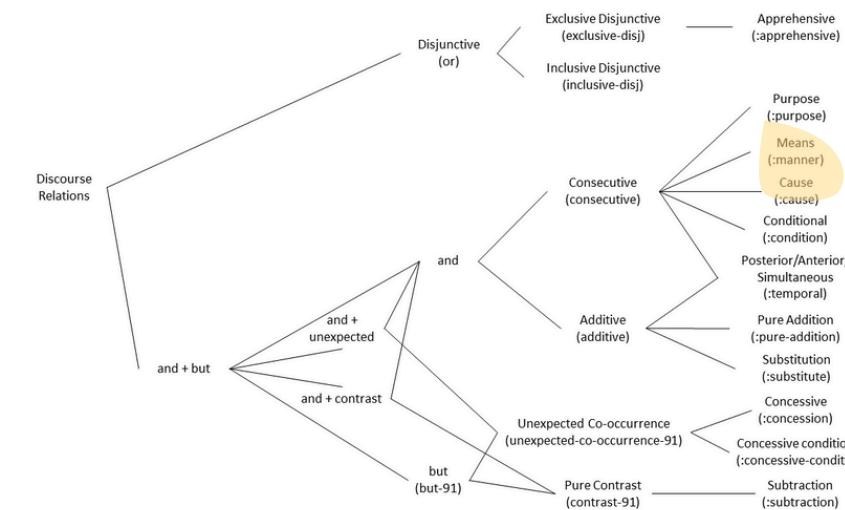
Means



Definition: One event characterizes another

UMR Concept: :manner

English Equivalent: by



He got into the army **by** lying about his age. ,
(g/ get-5)

:ARG1 (p/ person
:ref-person 3rd
:ref-number Singular)
:ARG2 (a/ army)
:manner (l/ lie-01
:ARG0 p
:ARG1 (t/ thing
:ARG2-of (a2/ age-01
:ARG1 p))

:aspect Performance
:modstr FullAff)

:Aspect Performance
:modstr FullAff)



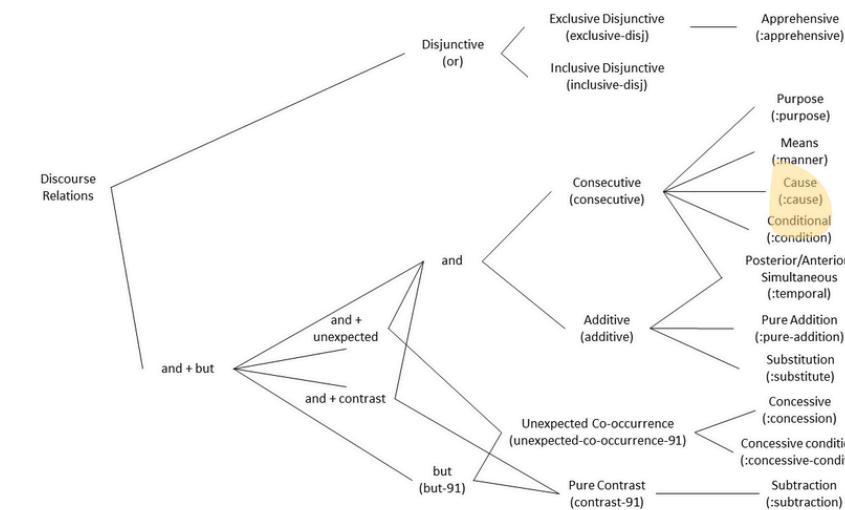
Causal



Definition: One event causes another

UMR Concept: cause-01, :cause

English Equivalent: because



```
Sarah moved back to California because she couldn't find a job in Washington.  
(m/ move-01)  
:ARG0 (p/ person  
:name (n/ name :op1 "Sarah))  
:ARG2 (s/ state :wiki "California"  
:name (n2/ name :op1 "California))  
:mod (b/ back)  
:cause (f/ find-01  
:ARG0 p  
:ARG1 (j/ job)  
:place (s/ state :wiki "Washington"  
:name (n3/ name :op1 "Washington"))  
:aspect Performance  
:modstr FullNeg)  
:aspect Performance  
:modstr FullAff)
```



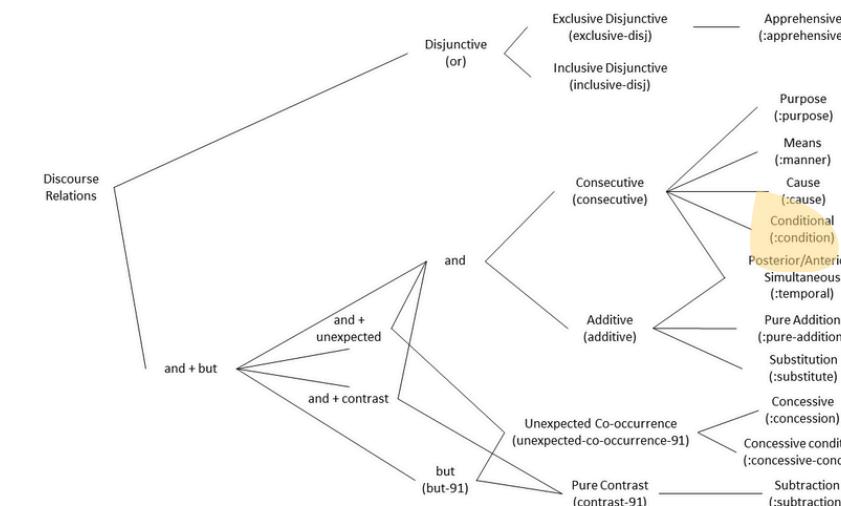
Conditional



Definition: Event contingent on another

UMR Concept: :condition,
have-condition-91

English Equivalent: if



If you touch it, it might explode. /
(e/ explode-01

:ARG1 (t/ thing
:ref-number Singular)
:condition (t2/ touch-01
:ARG0 (p/ person)
:ARG1 t
:aspect Performance
:modstr FullAff)
:aspect Performance
:modstr NeutAff)



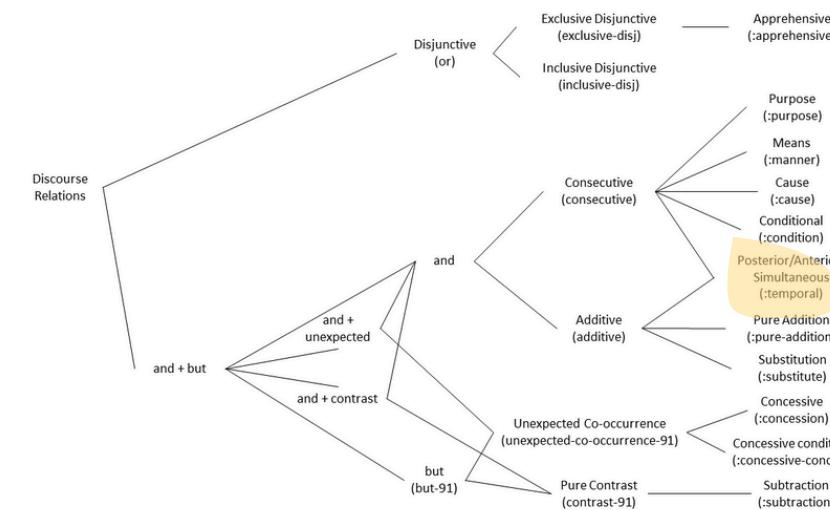
Anterior



Definition: One event before another

UMR Concept: :temporal

English Equivalent: and, before



```
I fed my dog before going to the office. / I fed my do
(f/ feed-01
    :ARG0 (p/ person
        :ref-person 1st
        :ref-number Singular)
    :ARG2 (d/ dog
        :poss p
        :ref-number Singular)
    :temporal (b/ before
        :op1 (g/ go-01
            :ARG1 p
            :ARG4 (o/ office
                :ref-number Singular)
            :aspect Performance
            :modstr FullAff)
        :aspect Performance
        :modstr FullAff)
```



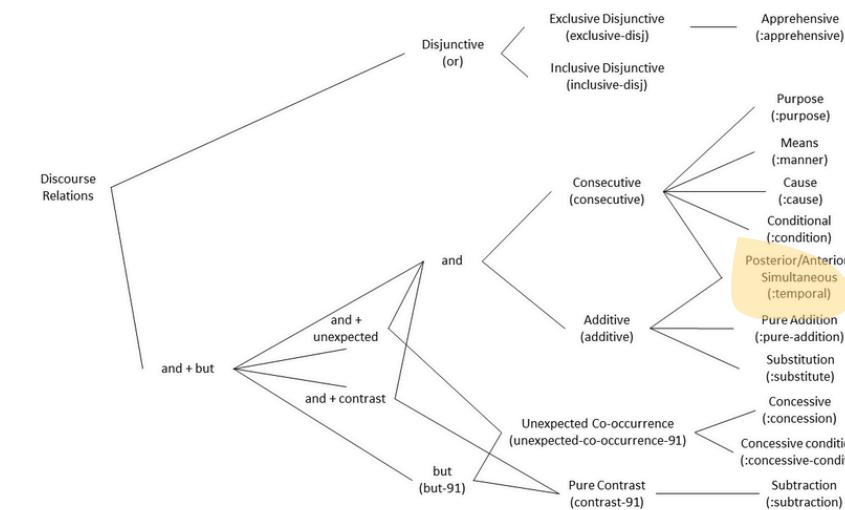
Posterior



Definition: One event after another

UMR Concept: :temporal

English Equivalent: and, after



I went home **after** paying the check. / I paid
(g/ go-01)

:ARG1 (p/ person
:ref-person 1st
:ref-number Singular)

:ARG4 (h/ home)

:temporal (a/ after)

:op1 (p2/ pay-01

:ARG0 p

:ARG3 (c/ check)

:aspect Performance

:modstr FullAff)

:aspect Performance

:modstr FullAff)



Adversative



Definition: Combined conjunctive and contrastive relations

UMR Concept: 'but-91'

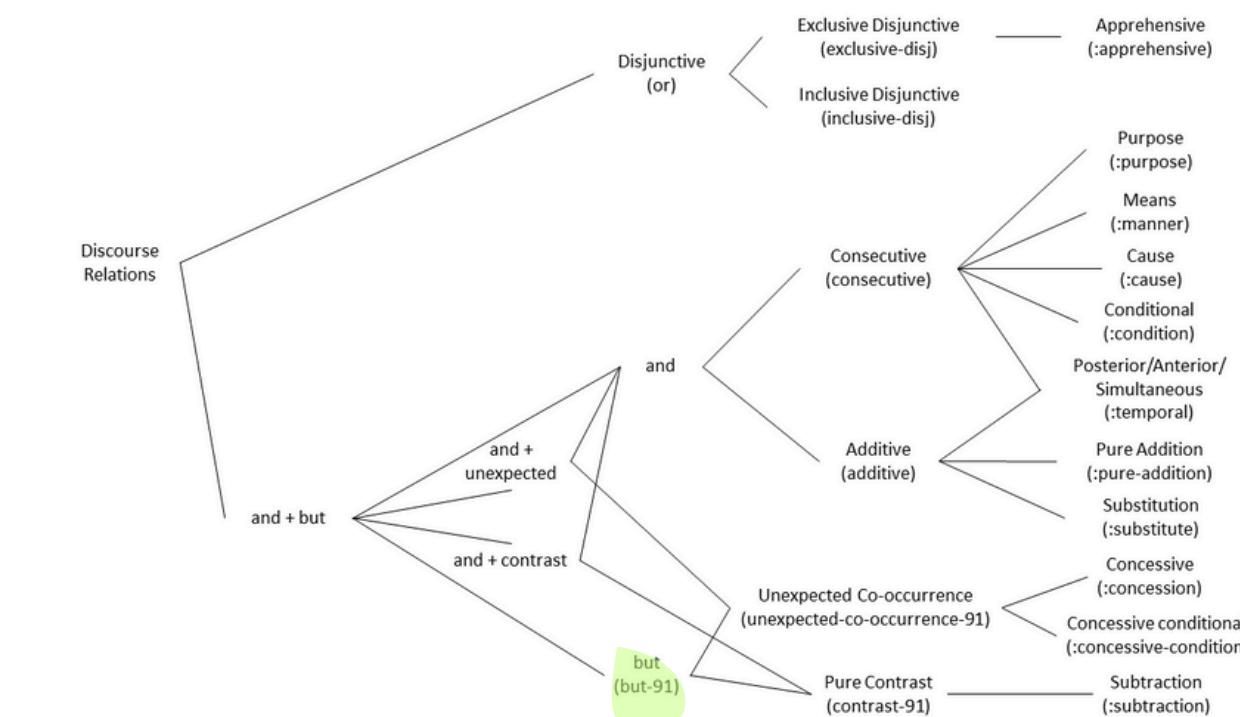
English Equivalent: ~but

Subtypes:

- (Pure) Contrast
- Unexpected co-occurrence

Higher level categories:

- And + Unexpected
- And + Contrast
- And + But



(pure) Contrast



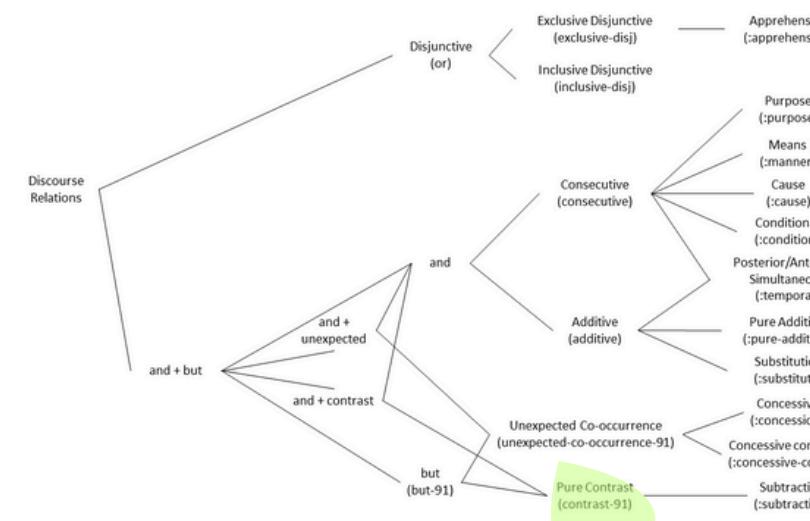
Definition: contrast between events,
no implications

UMR Concept: contrast-91

English Equivalent: not very
equivalent, but

Subtypes:

- Subtraction



```
Petja staratel'nyi, a Vanja lenivyj
Peter diligent CONJ Vanja lazy
'Peter is diligent, but [contrast] Vanja is lazy.'
(c/ contrast-91
:ARG1 (h/ have-mod-91
:ARG1 (p/ person
:name (n/ name :op1 "Peter"))
:ARG2 (s/ staratel'nyi 'diligent')
:aspect State
:modstr FullAff)
:ARG2 (h2/ have-mod-91
:ARG1 (p2/ person
:name (n2/ name :op1 "Vanja"))
:ARG2 (l/ lenivyj 'lazy')
:aspect State
:modstr FullAff))
```



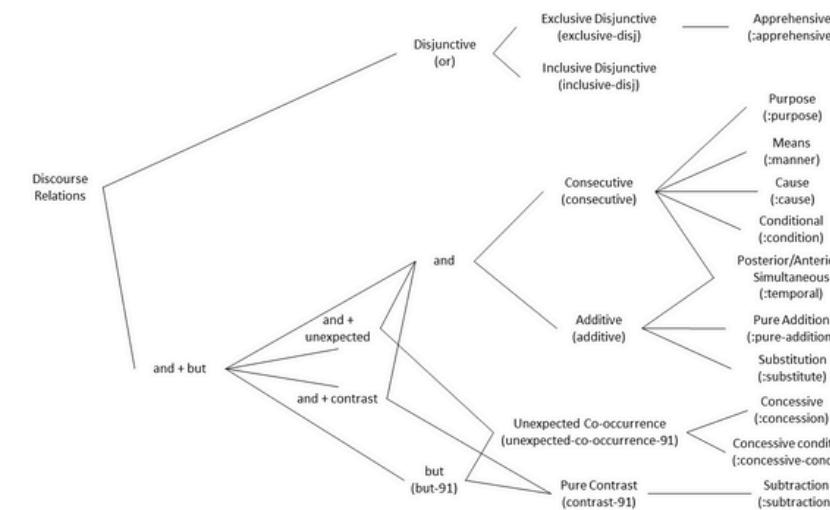
Subtraction



Definition: one event is additionally characterized by absence of another

UMR Concept: :subtraction

English Equivalent: except



People don't own tigers, **except** for Joe Exotic.

(o/ own-01

:polarity -

:ARG0 (p/ person

:refer-person Plural)

:ARG1 (t/ tiger

:ref-number Plural)

:subtraction (p2/ person

:name (n/ name

:op1 "Joe"

:op2 "Exotic"))

:Aspect State

:MODSTR FullNeg)



Unexpected Co-Occurrence



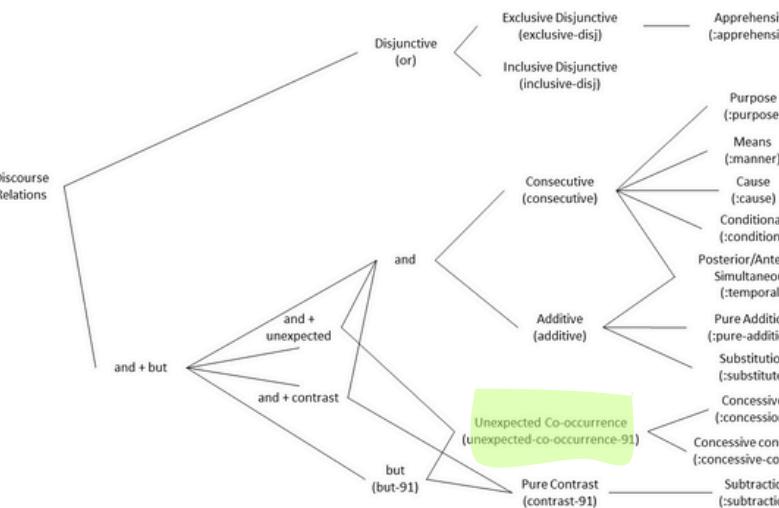
Definition: Juxtaposition with an unexpected second event

UMR Concept: unexpected-co-occurrence-91

English Equivalent: not very equivalent

Subtypes:

- Negative Circumstantial
- Concessive
- Concessive Conditional



```
Vanja prostudilsja, no posh l v shkolu
Vanja caught_cold CONJ went to school
'Vanja caught a cold, but [unexpected] went to school.'
(u/ unexpected-co-occurrence-91
:ARG1 (p/ prostudilsja-00 'catch a cold'
:experiencer (p2/ person
:name (n/ name :op1 "Vanja"))
:aspect Performance
:modstr FullAff)
:ARG2 (p3/ posh l-00 'go'
:actor p2
:goal (s/ shkolu 'school')
:aspect Performance
:modstr FullAff))
```



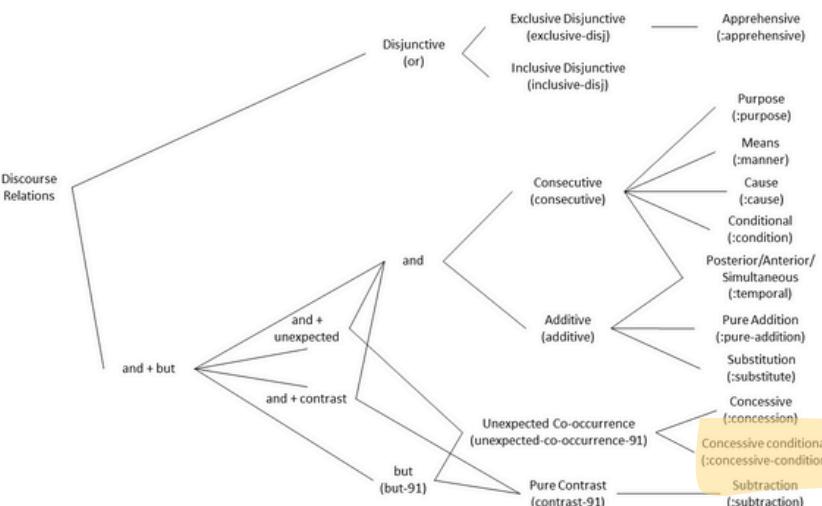
Negative Circumstantial



Definition: Absence of expected event

UMR Concept: :manner with negative polarity

English Equivalent: without



Sarah carried the bowl of punch into the living room **without** doing a somersault.

(c/ carry-01

```
:ARG0 (p/ person  
      :name (n/ name :op1 "Sarah")  
      :ARG1 (p2/ punch  
              :unit (b/ bowl))  
      :goal (r/ room  
             :mod (l/ living))  
      :manner (s/ somersault  
      :ARG0 p  
      :aspect Performance  
      :polarity -  
      :modstr FullNeg)  
      :aspect Performance  
      :modstr FullAff)
```



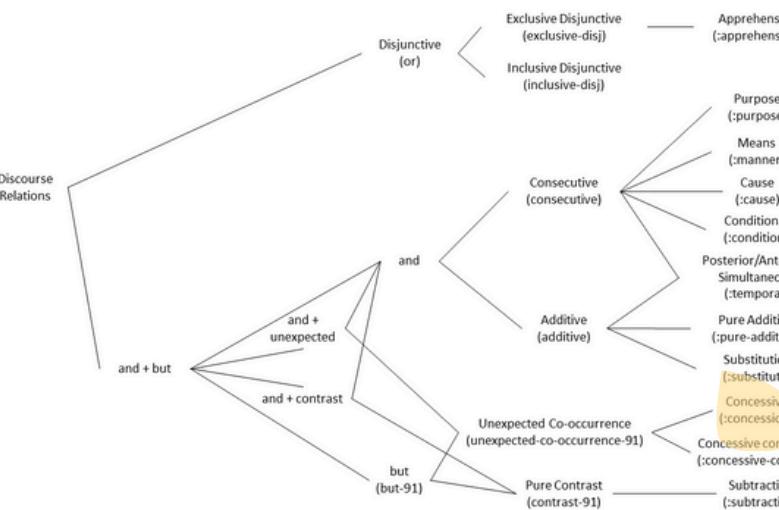
Concessive



Definition: Unexpected co-
occurrence

UMR Concept: have-concession-91,
:concession

English Equivalent: even though



Even though he was broke, he bought a new guitar.

(b/ buy-01

```
:ARG0 (p/ person  
:ref-person 3rd  
:ref-number Singular)  
:ARG1 (g/ guitar  
:mod (n/ new)  
:ref-number Singular)  
:concession (h/ have-mod-91  
:ARG1 p  
:ARG2 (b2/ broke)  
:aspect State  
:modstr FullAff)  
:aspect Performance  
:modstr FullAff)
```



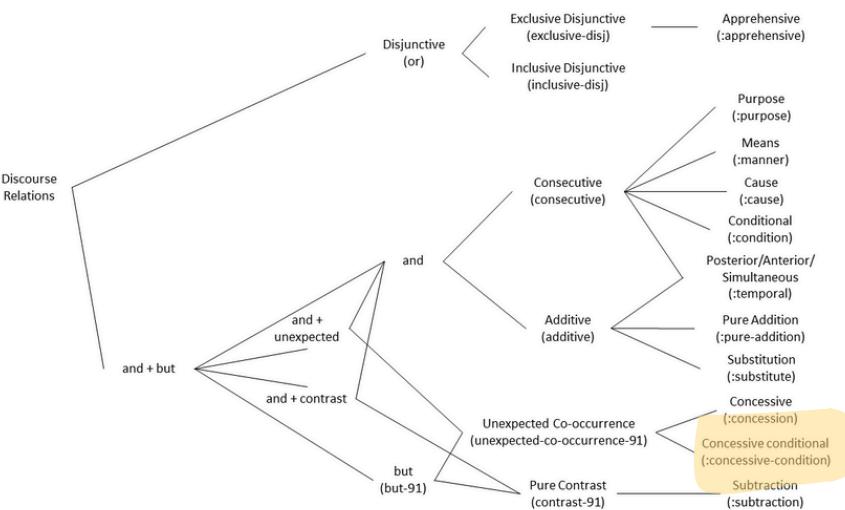
Concessive Conditional



Definition: Event true under unexpected conditions

UMR Concept: :concessive-condition

English Equivalent: even if



```
Even if you arrive only five minutes late, you will be fired.  
(f/ fire-02  
    :ARG1 (p/ person  
            :ref-person 2nd  
            :ref-number Singular)  
    :aspect Performance  
    :modstr FullAff  
    :concessive-condition (a/ arrive-01  
        :ARG1 p  
        :temporal (l/ late  
                    :extent (t/ temporal-quantity  
                                :quant 5  
                                :unit (m/ minute)))  
    :aspect Performance  
    :modstr FullAff))
```



And + But

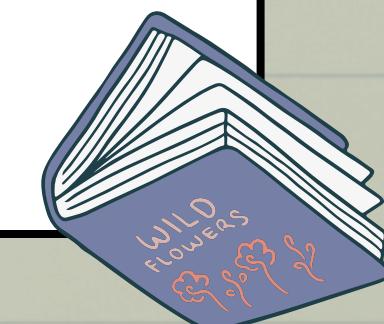
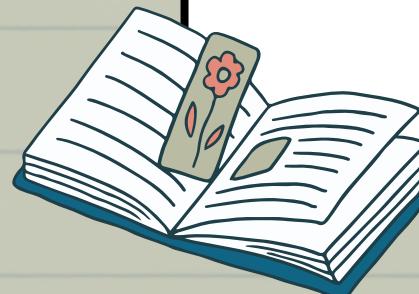
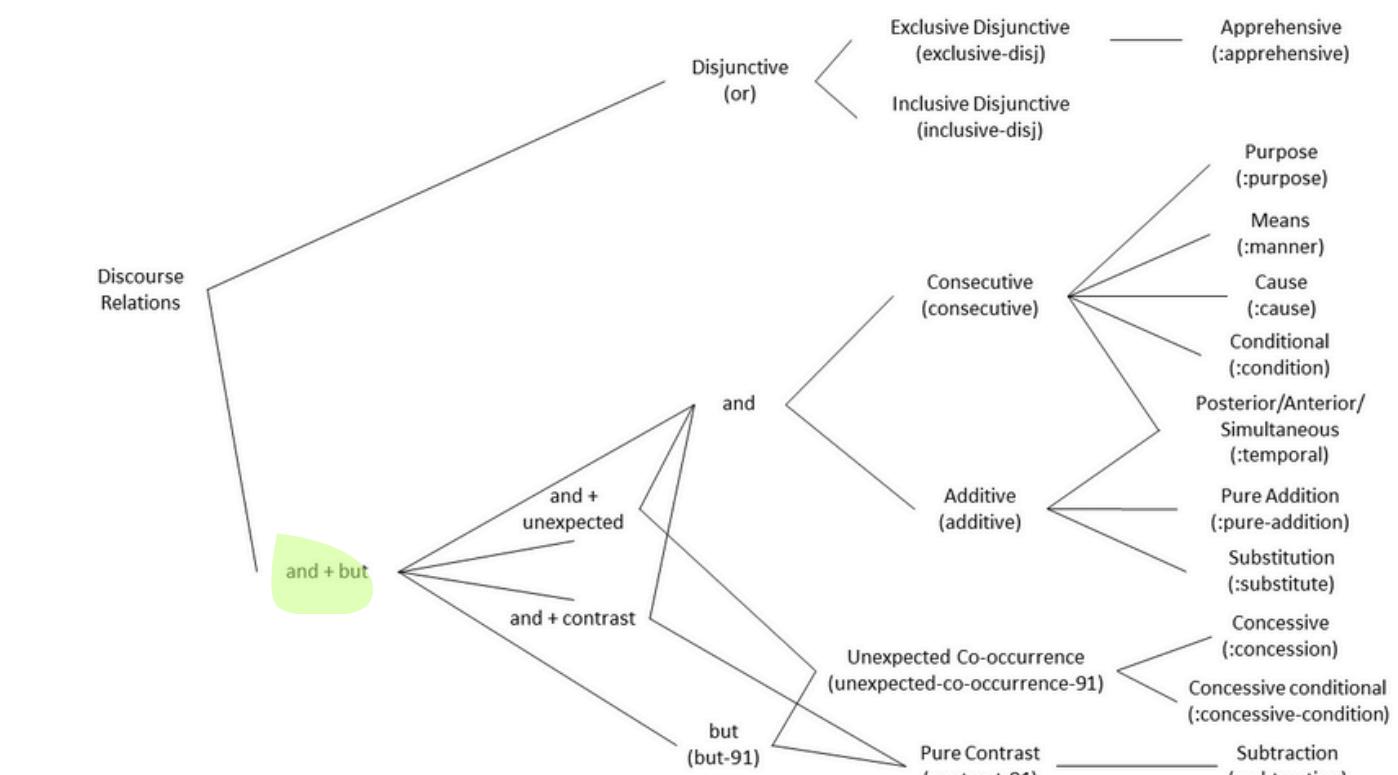


Definition: abstract concept used if language does not formally distinguish conjunctive relations from pure contrast relations

UMR Concept: numbered 'opX' roles

Subtypes:

- and
- and + unexpected
- and + contrast
- but



And



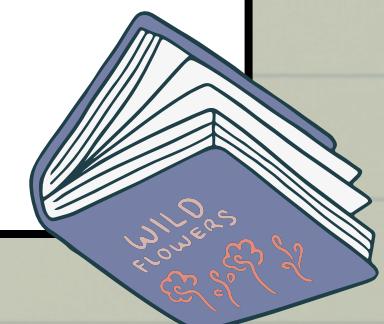
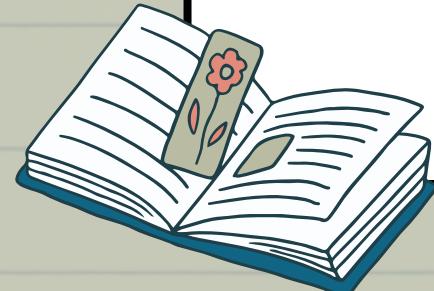
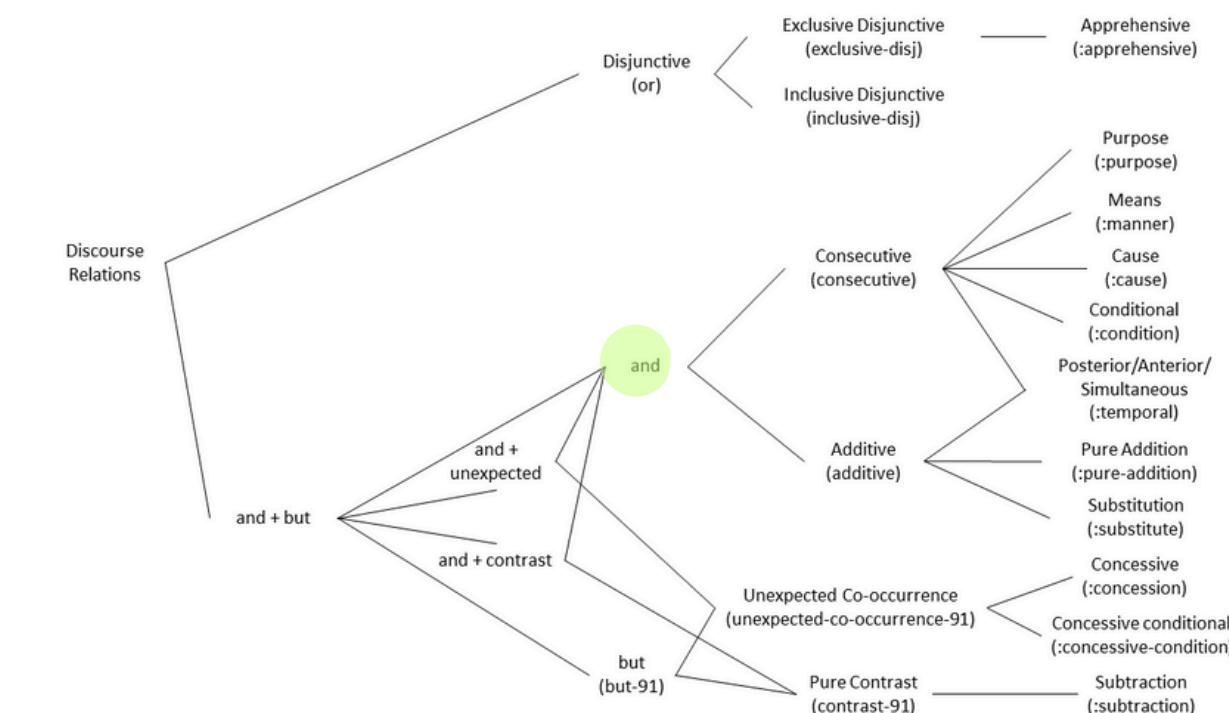
Definition: The more course-grained observation is 'and' for conjunctives

UMR Concept: 'and' and ':opX' roles

English Equivalent: and

Subtypes:

- Consecutive
- Additive



And + Unexpected

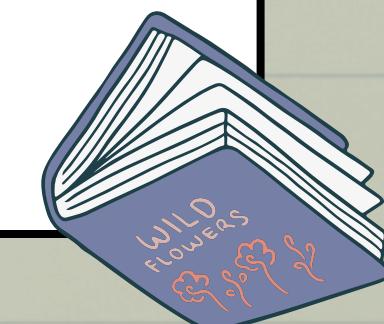
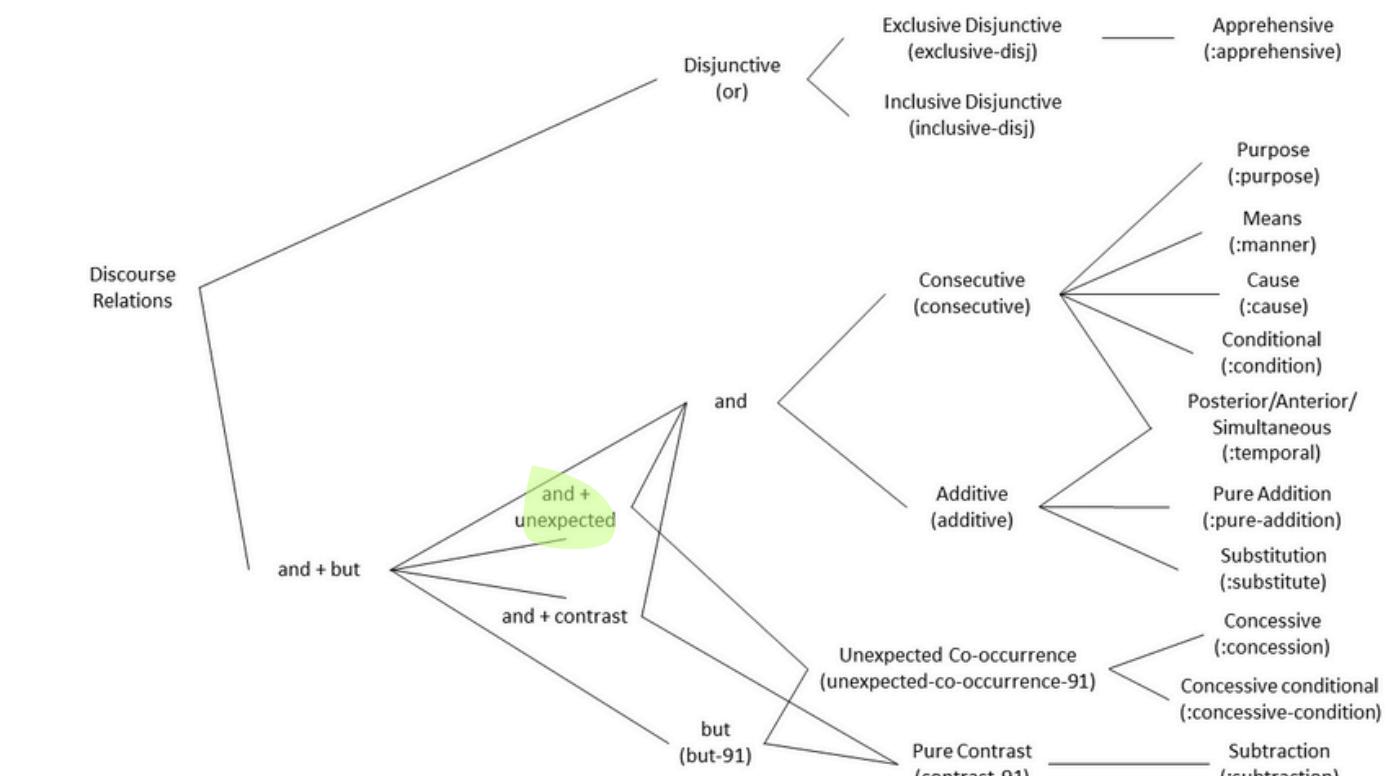


Definition: abstract concept if the language does not formally distinguish conjunctive from unexpected co-occurrence

UMR Concept: numbered ':opX' roles

Subtypes:

- and
- Unexpected co-occurrence



And + Contrast

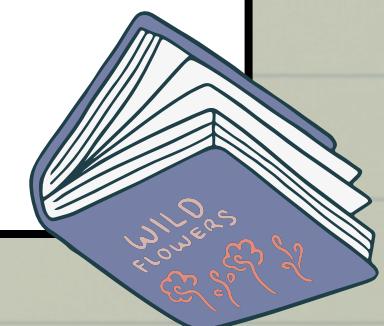
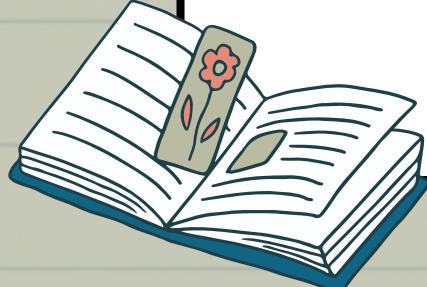
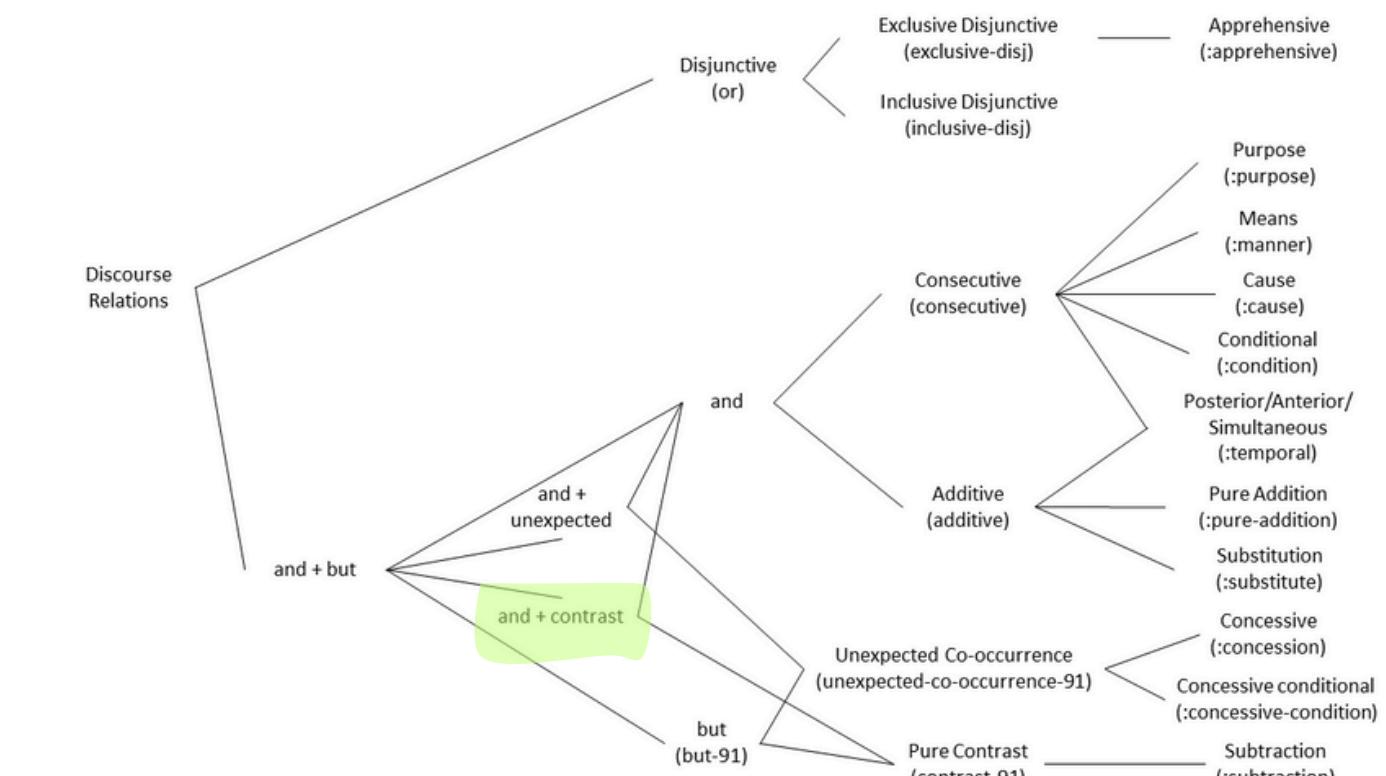


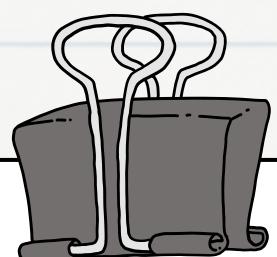
Definition: Abstract concept used if language does not formally distinguish conjunctive relations from pure contrast relations (may have distinct form of expressing unexpected co-occurrence)

UMR Concept: numbered ':opX' roles

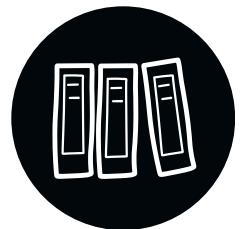
Subtypes:

- and
- (Pure) Contrast

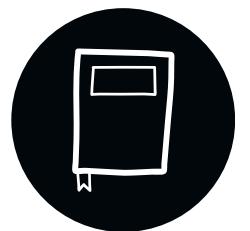




Activity 2

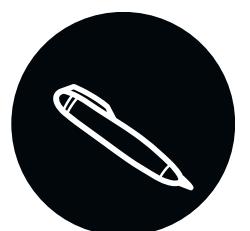


Get into groups



Share/Complete Your Lattices

- Feel free to ask questions or for help



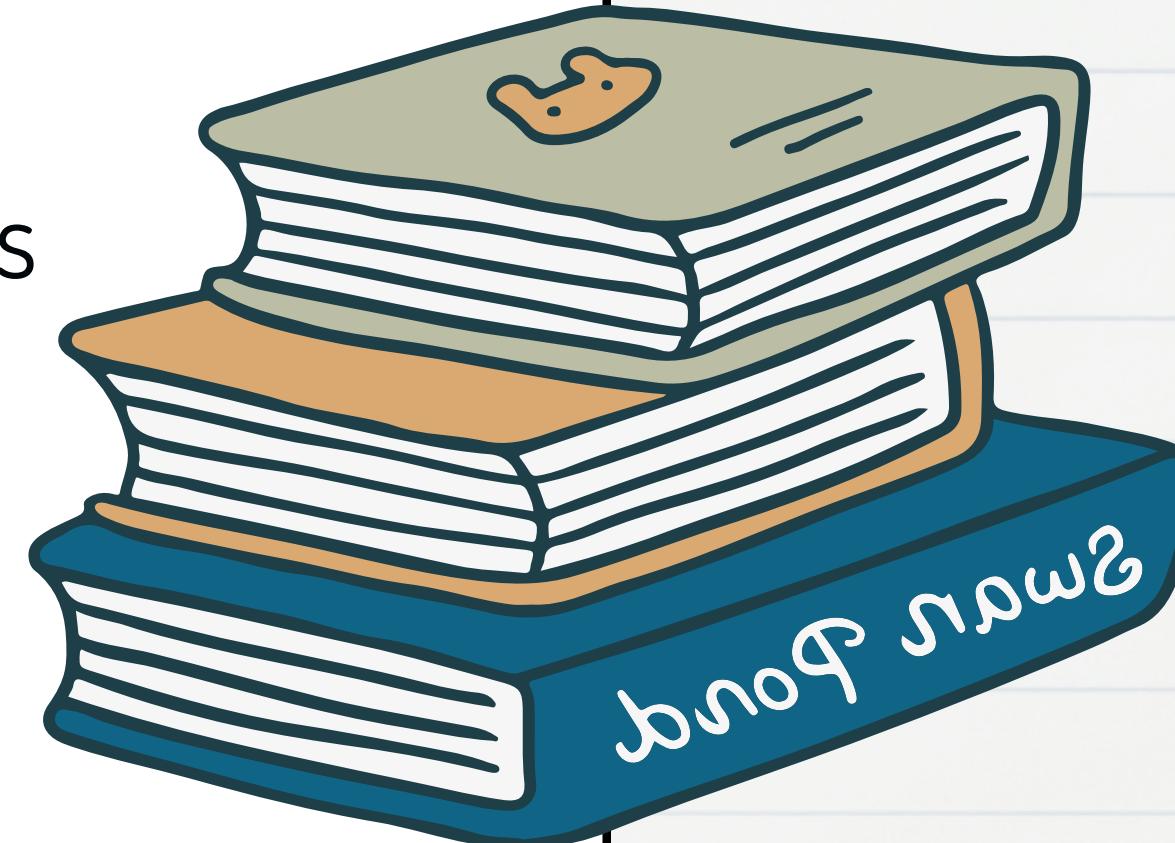
Printed activity

- Fill in the blanks with the correct type of discourse relation
- Help your team mates or divide and conquer!
- Benet will score the teams based on correct answers



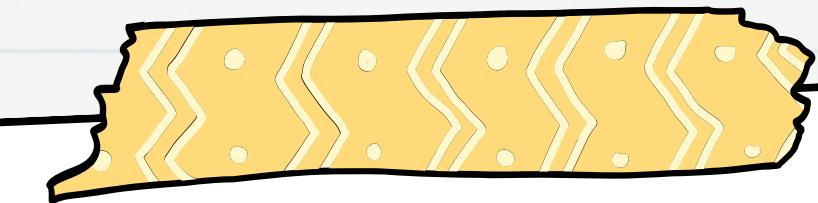
Overview

1. Discourse Relations
2. Coreference
3. Temporal Dependencies
4. Modal Dependencies





Coco the Coreference Cat



Coreference

Overview

- Coreference in UMR involves linking **anaphoric expressions** (aka pronouns) to their referents
- Essential for interpreting pronouns and named entities correctly
- Part of the **document level annotation**
- Two main types of coreference: **Entity Coreference** and **Event Coreference**



Entity Coreference

Definition

Linking anaphoric expressions to named entity

UMR

:coref

2 questions to answer

What counts as anaphoric?

For UMR we are using pronouns

What coreference relations

Most common:
identity relations
meaning 2 expressions have the same referent



Example

01



Sentence 1: Pope is an American.
Sentence 2: "He denied any wrongdoing"

02

Annotation: :same-entity relation
between "Pope" and "He".

03

(s3/ sentence
:coref (s2p :same-entity s3p))





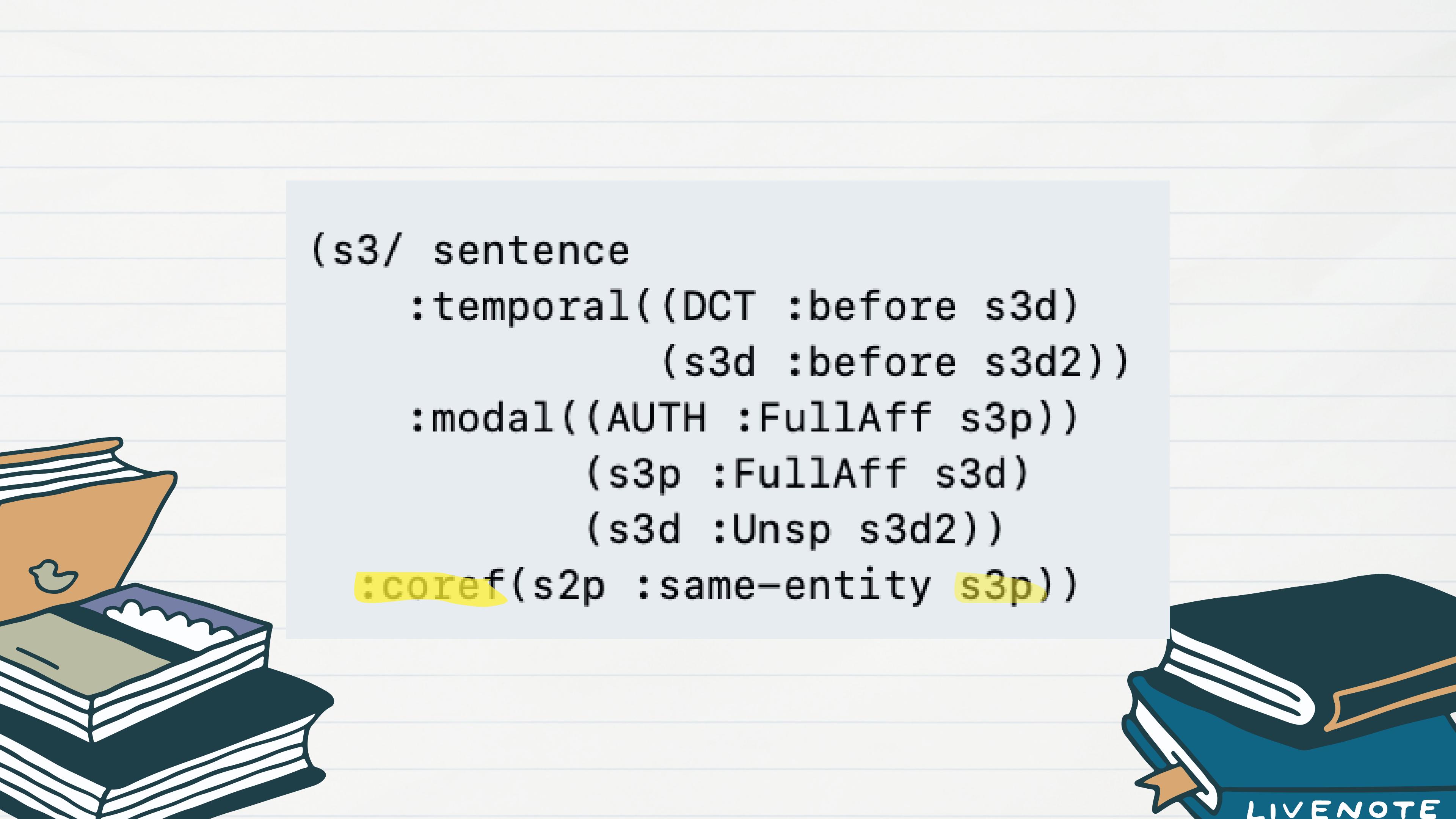
Snt2: Pope is the American businessman who was convicted last week on spying charges and sentenced to prison for 20 years.

(i/ identity-91
 :ARG1 (p/ person :wiki "Edmond_Pope"
 :name (n/ name "op1 "Pope"))
 :ARG2 (b/ businessman
 :mod (n2/ nationality :wiki "United_States"
 :name (n3/ name :op1 "America")))
 :ARG1-of (c/ convict-01
 :ARG2 (c2/ charge-05
 :ARG1 b
 :ARG2 (s/ spy-02
 :ARG0 b
 :modpred c2))
 :temporal (w/ week
 :mod (l/ last))
 :aspect Performance
 :modstr FullAff)
 :ARG1-of (s2/ sentence-01
 :ARG2 (p2/ prison
 :mod (c3/ country :wiki "Russia"
 :name (n4/ name :op1 "Russia"))
 :duration (t/ temporal-quantity
 :quant 20
 :unit (y/ year)))
 :ARG3 s
 :aspect Performance
 :modstr FullAff)
 :aspect State
 :modstr FullAff)

Snt3: He denied any wrongdoing.

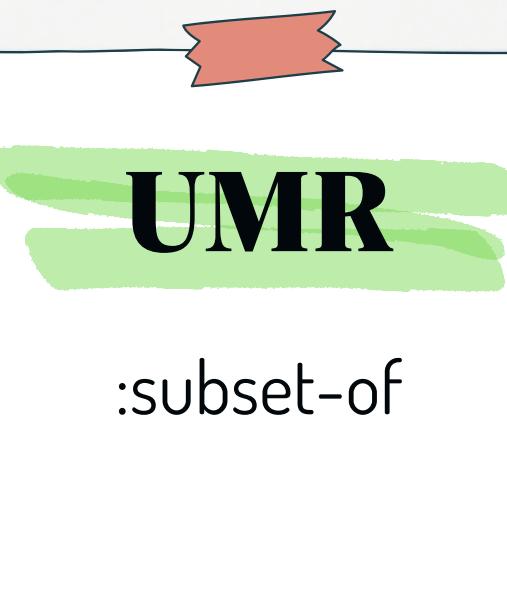
(d/ deny-01
 :ARG0 (p/person
 :ref-person 3rd
 :ref-number Singular)
 :ARG1 (t/ thing
 :ARG1-of (d2/ do-02
 :ARG0 p
 :ARG1-of (w/ wrong-02
 :modpred d))
 :aspect Performance
 :modstr FullAff)





```
(s3/ sentence
  :temporal( (DCT :before s3d)
              (s3d :before s3d2) )
  :modal( (AUTH :FullAff s3p) )
          (s3p :FullAff s3d)
          (s3d :Unsp s3d2) )
  :coref(s2p :same-entity s3p) )
```

Subset-of



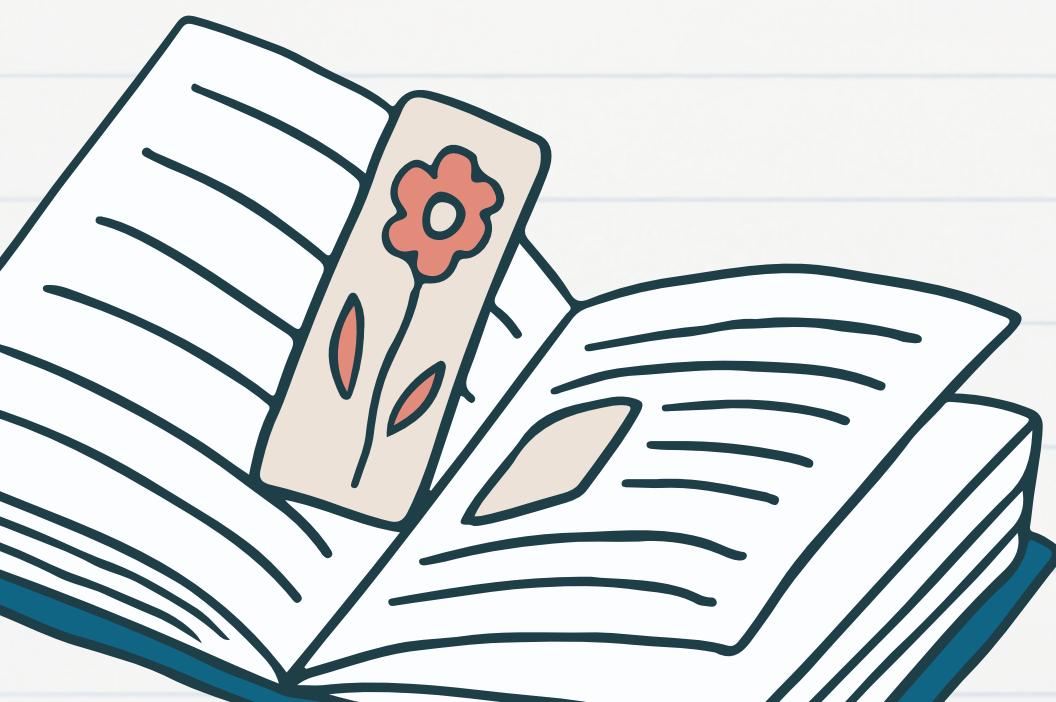
Definition

- Additional info about entity coreference
- Role used to relate mentions of sets of entities to mentions of entities belonging to such a set

Example

The word **we** include reference to **two entities** - author and the person who is “possessive and controlling.” Pronoun **he**, given constant p2, refers to one of two entities. Thus, p2 is :subset-of p3 node.

“**He** is very possessive and controlling but **he** has no right to be as **we** are not together.”



He is very possessive and controlling but he has no r
(c/ contrast-01

```
:ARG1 (a/ and
      :op1 (p/ possessive-03
            :ARG0 (p/ person
                  :ref-person 3rd
                  :ref-number Singular)
            :degree (v/ very)
            :aspect State
            :modstr FullAff)
      :op2 (h/ have-mod-91
            :ARG1 p
            :ARG2 (c2/ controlling)
            :degree (v/ very)
            :aspect State
            :modstr FullAff))
:ARG2 (r/ right-05
      :ARG1 p
      :ARG2 a
      :ARG1-of (c3/ cause-01
                :ARG0 (h2/ have-mod-91
                      :ARG1 (p2/ person
                            :ref-person 1st
                            :ref-number Plural)
                      :ARG2 (t/ together)
                      :aspect State
                      :modstr FullNeg))
      :aspect State
      :modstr FullNeg))
```

(s/ sentence

```
:temporal ((DCT :overlap s1p)
            (s1p :overlap s1h)
            (s1h :overlap s1r)
            (s1r :overlap s1h2))
:modal ((AUTH :FullAff s1p)
        (AUTH :FullAff s1h)
        (AUTH :FullNeg s1r)
        (AUTH :FullNeg s1h2))
:coref (p :subset-of p2))
```



Event Coreference

Definition

Linking different mentions of the same event

Details

The :subset-of relation is also used to annotate the subset relations between two event mentions, with one referring to a subset of another

UMR

:same-event



Example Part 1

El-Shater and Malek's property was confiscated and is believed to be worth millions of dollars.

```
(a/ and
  :op1 (c/ confiscate-01
    :ARG1 (a2/ and
      :op1 (p/ property
        :poss (p2/ person
          :wiki "Khairat_el-Shater"
          :name (n/ name :op1 "El-Shater")))
      :op2 (p3/ property
        :poss (p4/ person
          :wiki -
          :name (n2/ name :op1 "Malek")))
    :aspect Performance
    :modstr FullAff)
  :op2 (b/ believe-01
    :ARG1 (w/ worth-01
      :ARG1 a2
      :ARG2 (m/ multiple
        :op1 (m2/ monetary-quantity
          :quant 1000000
          :unit (d/ dollar)))
    :aspect State
    :modpred b)
  :aspect State
  :modstr FullAff))
```

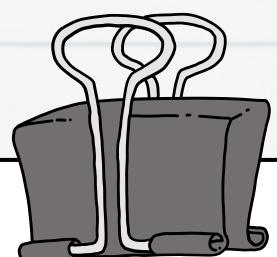
```
(s1/ sentence
  :temporal ((PAST_REF :includes s1c)
    (DCT :overlap s1b)
    (s1b :overlap s1w))
  :modal ((AUTH :FullAff s1c)
    (AUTH :FullAff NULL_BELIEVER)
    (NULL_BELIEVER :FullAff s1b)
    (s1b :Unsp s1w)))
```

Example Part 2

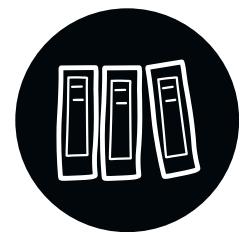
Abdel-Maksoud stated the **confiscation** will affect the Brotherhood's financial bases.

```
(s/ state-01
  :ARG0 (p/ person
    :wiki "Hamdeen_Sabahi"
    :name (n/ name :op1 "Abdel-Maksoud"))
  :ARG1 (a/ affect-01
    :ARG0 (c/ confiscate-01)
    :ARG1 (b/ base
      :poss (o/ organization
        :wiki "Muslim_Brotherhood"
        :name (n2/ name :op1 "Brotherhood"))
      :mod (f/ finance))
    :aspect Performance
    :quot s
    :modstr FullAff)
  :aspect Performance
  :modstr FullAff)
```

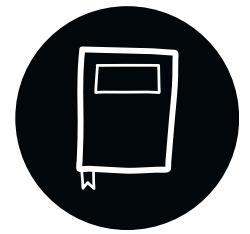
```
(s2/ sentence
  :temporal ((s1c :after s2s)
              (s2s :after s2a))
  :modal ((AUTH :FullAff s2s)
          (AUTH :FullAff s2p)
          (s2p :FullAff s2c)
          (s2p :FullAff s2a)))
  :coref (s1c :same-event s2c))
```



Activity 3



Get into your groups



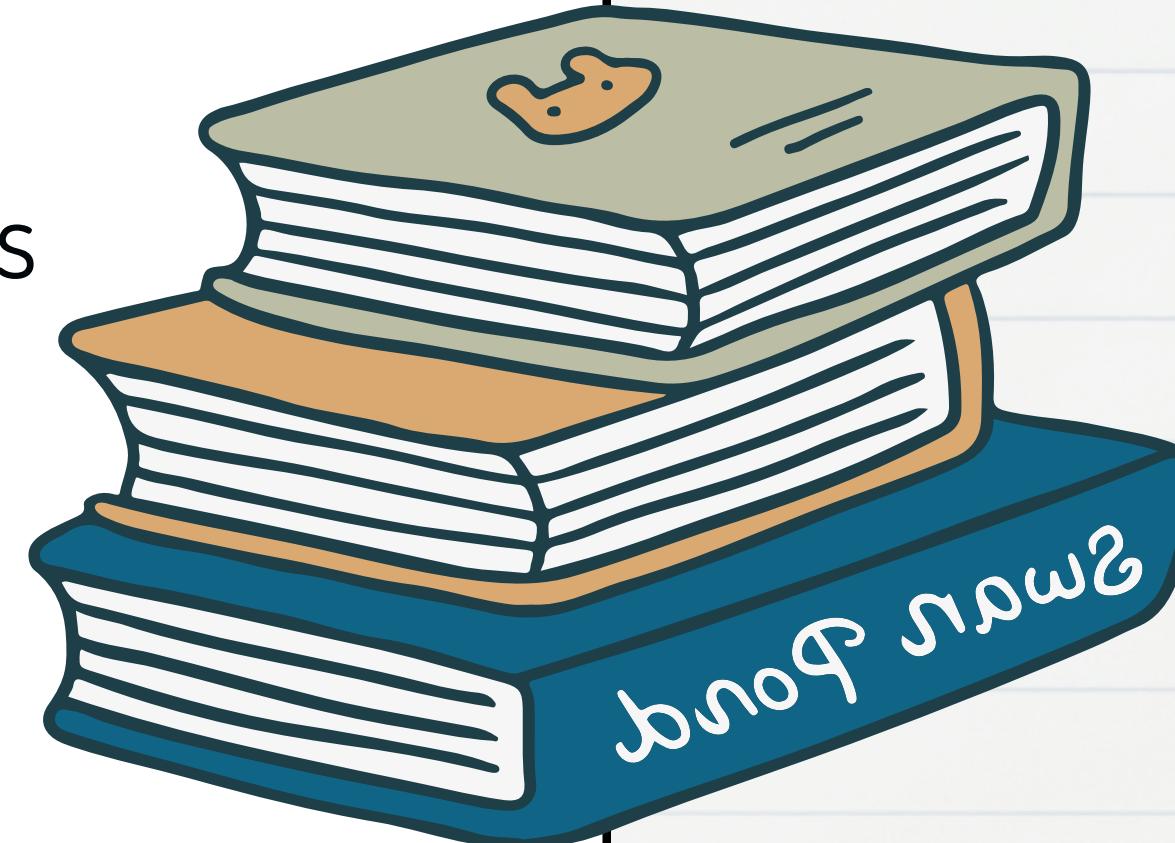
Complete Activity for Coreference

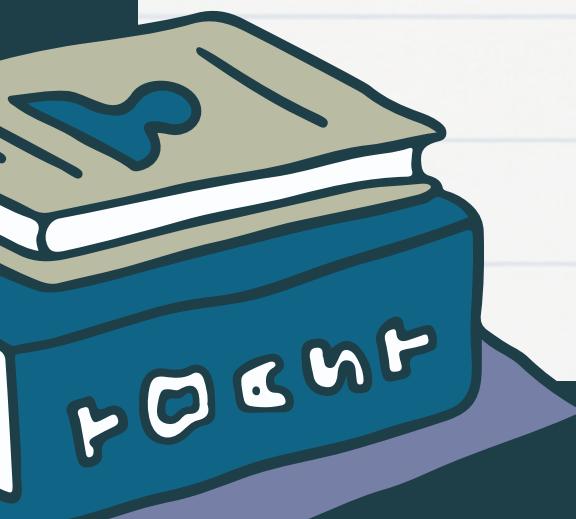
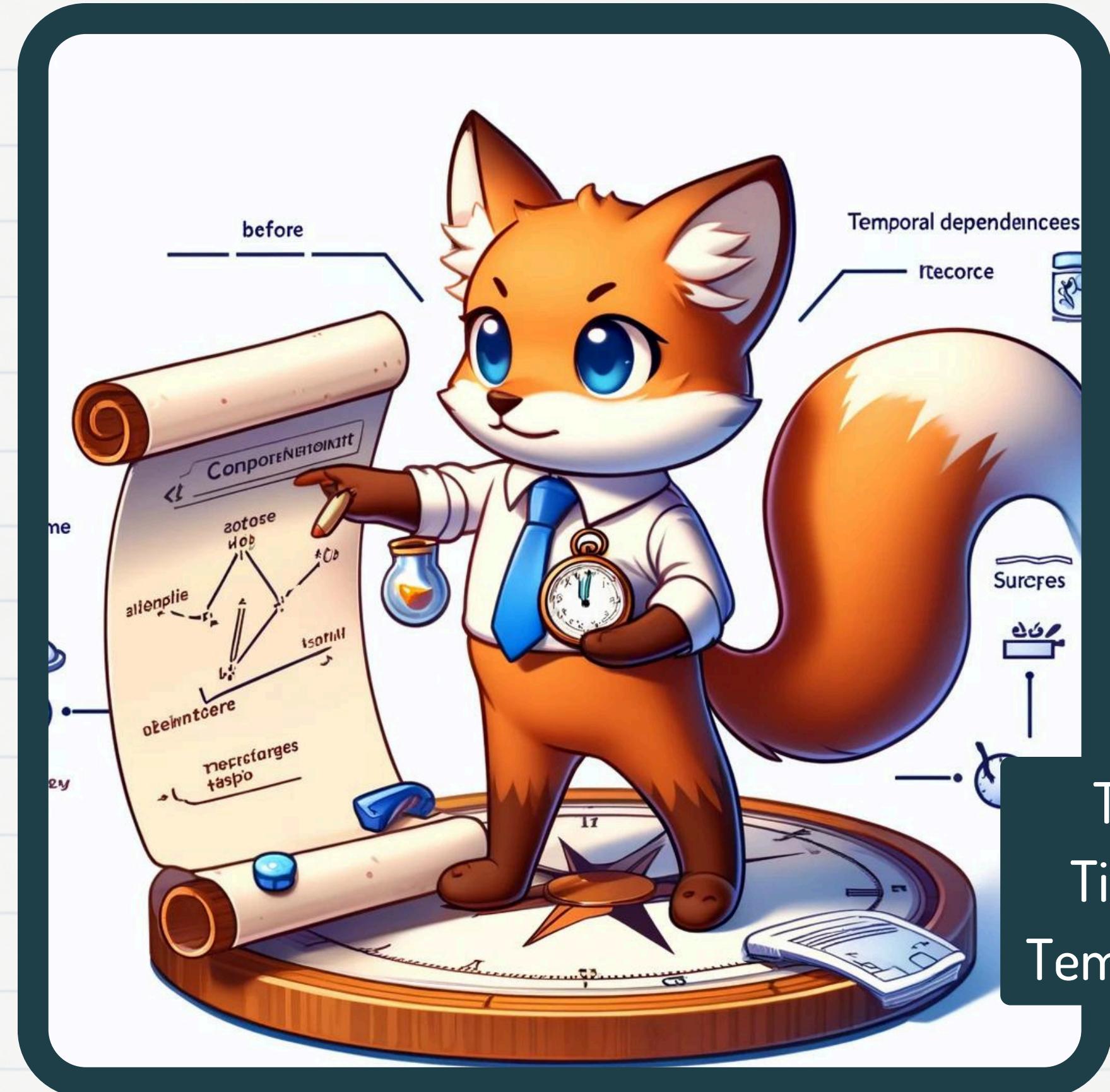
- Feel free to ask questions or for help

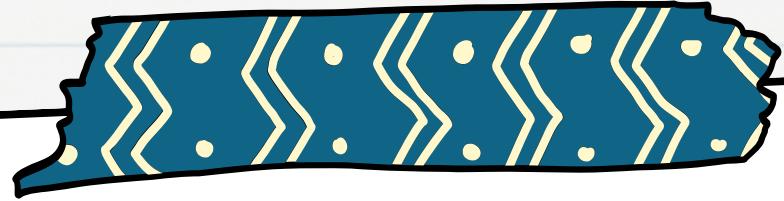


Overview

1. Discourse Relations
2. Coreference
3. Temporal Dependencies
4. Modal Dependencies







Temporal Dependencies

Overview

- Temporal annotation in UMR is done at both the sentence level and the document level.
- Sentence level:
 - Annotates **time expressions** as modifiers of predicates and their relations to the document creation time (**DCT**).
- Document level:
 - Focuses on event-event and time-time relations, creating a **temporal superstructure**.



Note



- Note that the temporal dependency annotations are read **right to left**
 - B :before A means A occurred before B.
 - :**overlap** relation indicates partial or exact overlap, but does not indicate in which direction the overlap occurs
 - :**contains** relation indicates that all of one event occurred during another event/state
 - B :**contains** A means that A contains all of the B event

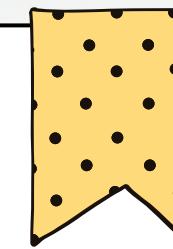
Sentence Level Example

In April 1998 Arab countries signed an anti-terrorism agreement that binds the signatories to coordinate to fight terrorism.

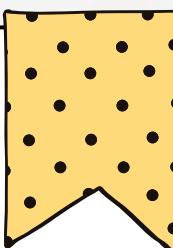
```
(s/ sign-02
  :ARG0 (c/ country
    :mod (e/ ethnic-group
      :wiki "Arabs"
      :name (n/ name :op1 "Arab")))
  :ARG1 (a/ agree-01
    :topic (c2/ counter-01
      :ARG1 (t/ terrorism))
    :ARG0-of (b/ bind-01
      :ARG1 c
      :ARG2 (c3/ coordinate-01
        :ARG1 c
        :purpose (f/ fight-01
          :ARG0 c
          :ARG1 t
          :aspect Activity
          :modstr FullAff)
        :aspect Activity
        :modpred b)
        :aspect Activity
        :modstr FullAff))
    :temporal (d/ date-entity :year 1998 :month 4)
    :temporal (b2/before :op (n/now))
    :aspect Performance))
```



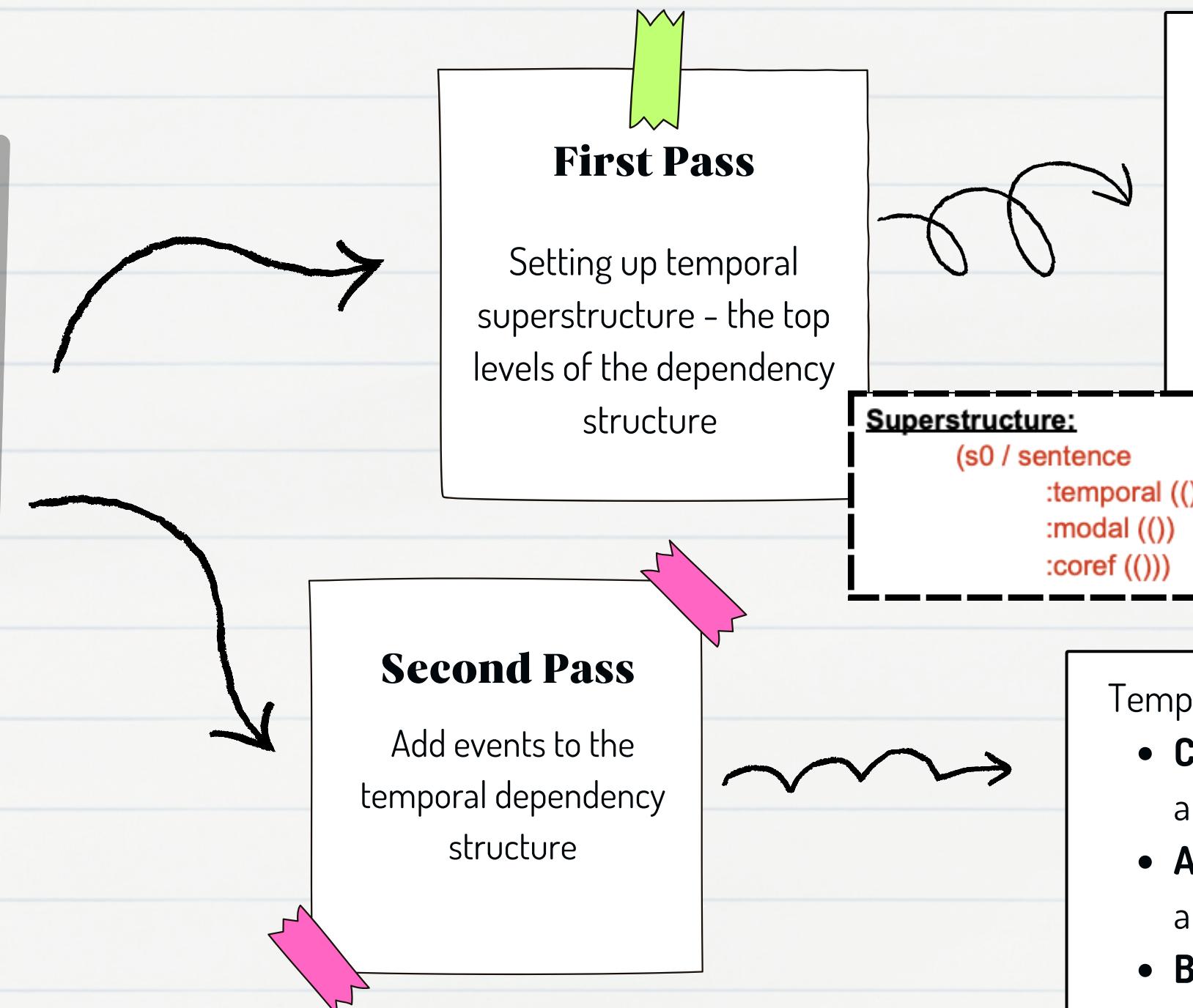
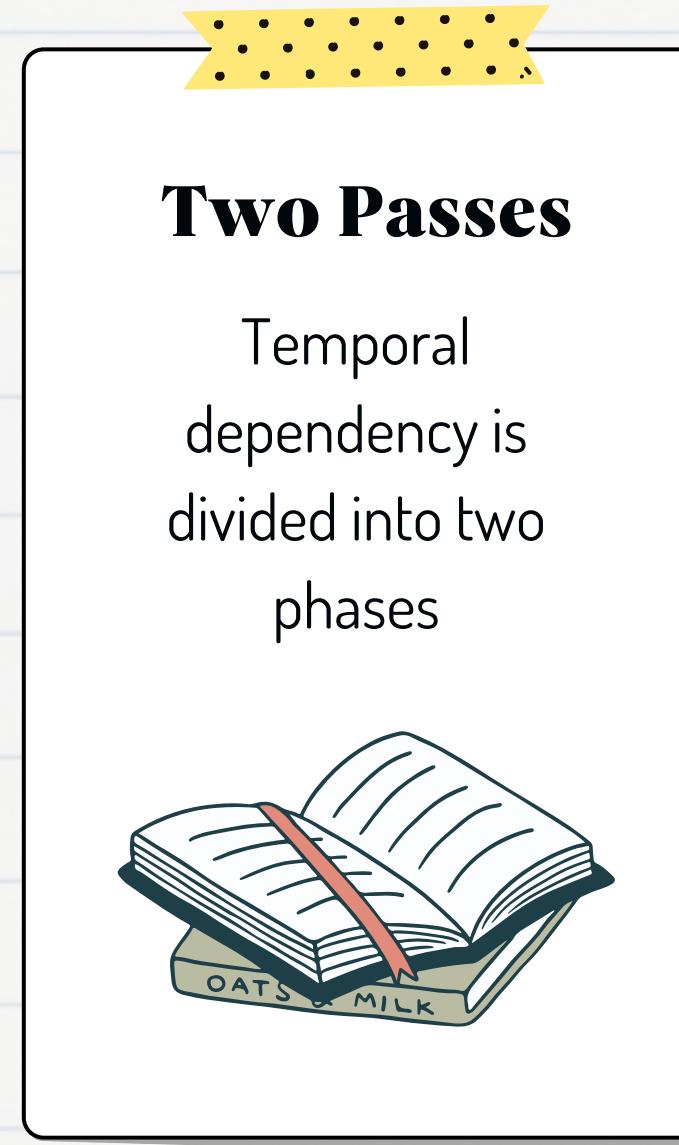
Annotation: :temporal (d/
date-entity :year 1998
:month 4),



:temporal (b2/before :op
(n/now))



Annotating Doc Level Temporal Dependencies



Temporal Superstructure

- Nodes:** Three types - Pre-defined metanodes, Time expressions, Events.
- Pre-defined Metanodes:** PAST_REF, PRESENT_REF, FUTURE_REF, DCT
- Time Expressions:** Locatable (Concrete, Vague) and Unlocatable.
- Events:** Linked to time expressions or other events.

Temporal Relations:

- Contained:** Event is fully within another.
- After:** Event occurs after another.
- Before:** Event occurs before another.
- Overlap:** Events partially or fully overlap.

Nodes

01

Pre-defined Metanodes

- nodes at the top of every temporal dependency structure connected directly to the ROOT node
- Nodes:
 - PAST_REF
 - PRESENT_REF
 - FUTURE_REF
 - DCT



02

Time Expressions

- They are identified and annotated before linking events.
- Categorized into locatable and unlocatable types.
- Will go into more detail in later slides

03

Events

- All events need to receive a temporal annotation
- Each event is annotated as the child of either a time expression in the superstructure or another event



Nodes: Types of Time Expressions

01

Locatable Time Expressions

- Can be placed on a timeline.
- Subcategories:
 - **Concrete**: Specific points in time.
 - **Absolute**: Directly refer to a specific date or time.
 - Example: "May 2015"
 - Reference: ROOT
 - **Relative**: Depend on another time expression for interpretation.
 - Example: "today," "two days later"
 - Reference: DCT or another concrete time expression.
 - **Vague**: General time periods without specific dates.
 - Example: "nowadays"
 - Reference: PRESENT_REF, PAST_REF, or FUTURE_REF

Taxonomy				Examples	Possible Reference Times
Time Expressions	Locatable Time Expressions	Absolute	May 2015	ROOT	
		Concrete	today, two days later	DCT, another Concrete	
		Relative	nowadays	Present/Past/Future_Ref	
	Vague	every month		--	
Unlocatable Time Expressions					

- DCT -- document time
- PAST_REF -- vague time expressions in the past
- PRESENT_REF -- vague time expressions in the present
- FUTURE_REF -- vague time expressions in the future

Nodes: Types of Time Expressions

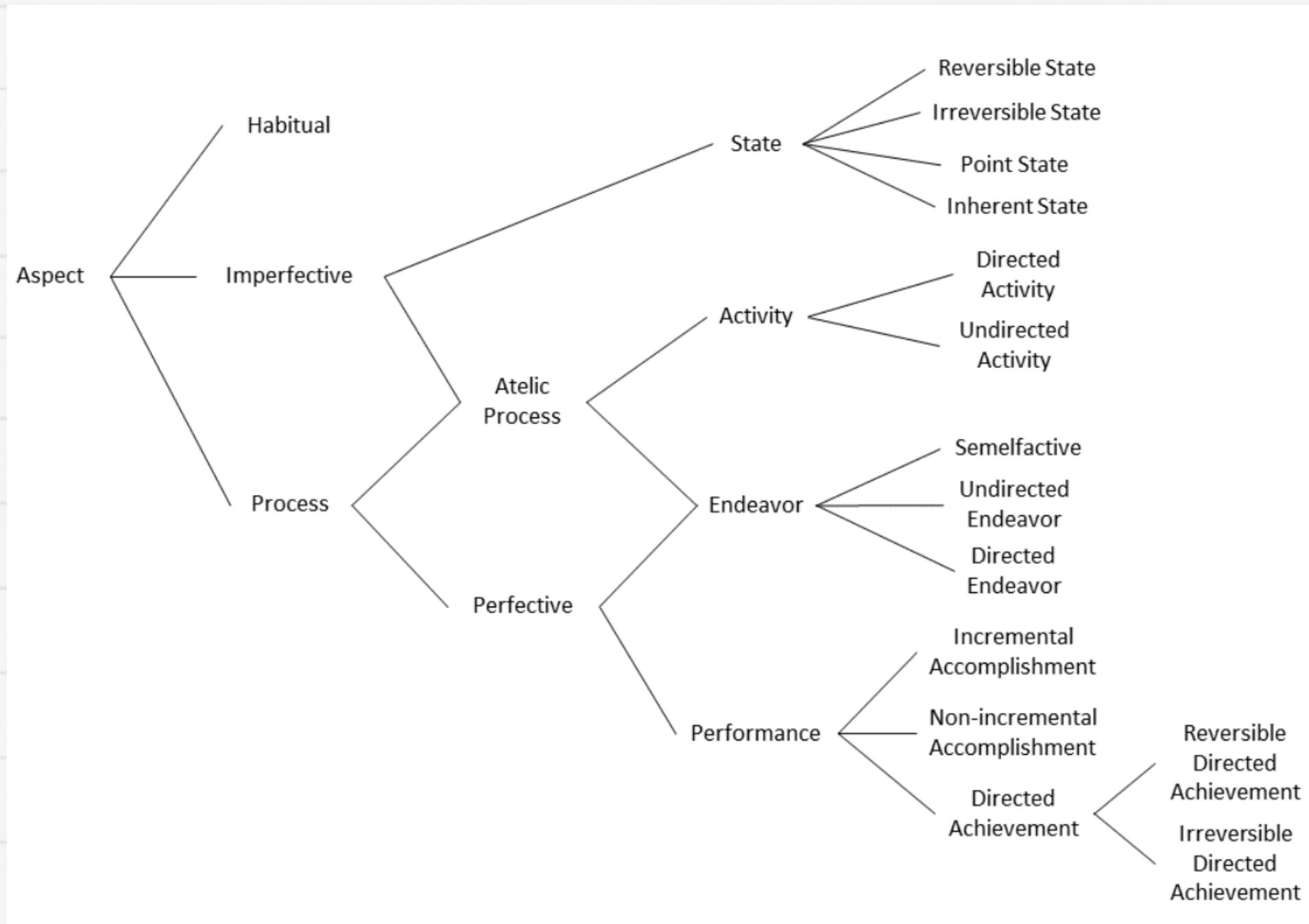
02

Unlocatable Time Expressions

- Refer to durations or recurring intervals.
- Not represented in the temporal dependency structure.
 - However, they do influence the aspect notation (covered in dif section)
- Examples: "for three hours," "every day"

Taxonomy			Examples		Possible Reference Times
Time Expressions	Locatable Time Expressions	Concrete	Absolute	May 2015	ROOT
			Relative	today, two days later	DCT, another Concrete
		Vague	nowadays	Present/Past/Future_Ref	
	Unlocatable Time Expressions			every month	--

Quick Look



Aspect Notation

I am not covering this aspect, but just want to reference the lattice concerning unlocatable time expressions.

Match the Time Expressions up!

Absolute

Vague

Relative

Unlocatable

Sentences

A: on Christmas Day

B: Two days later C: 12:30pm

D: During that summer

E: In prehistoric times

F: bi-weekly review

G: The Jurassic Period

H: everyday

I: In the old days

J: after the
announcement



Match the Time Expressions up!

Absolute

A: on Christmas Day

C: 12:30pm

G: The Jurassic Period

Relative

B: Two days later

J: after the
announcement

Vague

D: During that summer

E: In prehistoric times

I: In the old days

Sentences

Unlocatable

F: bi-weekly review

H: everyday



Steps for Annotating Temporal Relations

1. Identify Time Expressions

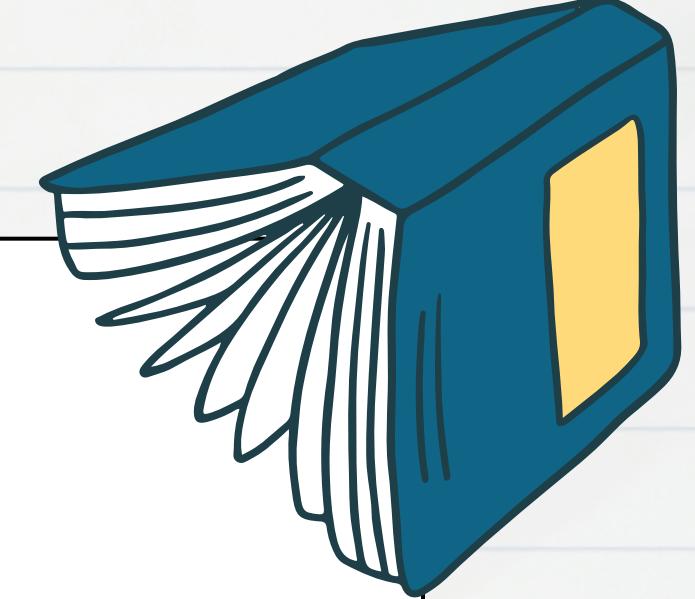
- Annotate events **linked to time expressions** within the same line
- Example: "In April 1998, Arab countries signed..."
- Annotation:
 - :temporal (d/ date-entity :year 1998 :month 4)

2. Link to Previous Events

- Consider time expressions in other lines if not linked in the first step.
- If no time expressions:
 - link to the **immediately preceding** event.
- Example: "He dumps all his pears into the basket."
- Annotation:
 - :temporal ((s15g :after s16d))

3. Link to Tense Metanodes

- If unable to link to time expressions or events, link to appropriate tense metanodes
 - PAST_REF,
 - PRESENT_REF,
 - FUTURE_REF
 - DCT

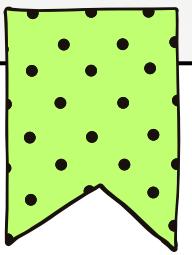


Additional Details:

Temporal Relations

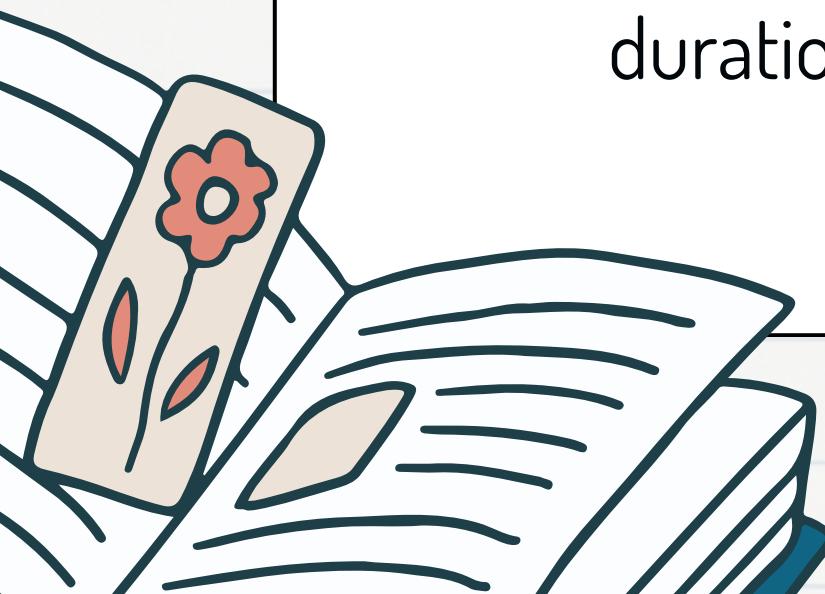
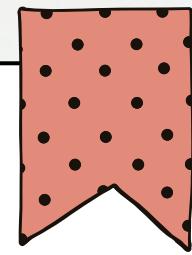
Contained or Overlap

- Use :overlap
 - for events beginning and ending simultaneously.
- Use :contained
 - for events within the temporal duration of another.



Causally-related events

- Always annotated as :after.
- Example: "The crops grew well because it rained enough."
- Annotation:
 - (rain :after grow)
- Even though the event (raining) and caused event (growing) overlap.



Pear Story

snt15 A-and u-h and then he gets down out of the tree,
(s15g / get-05
:ARG1 (s15p / person
:ref-person 3rd
:ref-number Singular)
:ARG2 (s15d / down)
:source (s15t / tree
:refer-number Singular)
:aspect Performance
:temporal (s15t2 / then)
:modstr FullAff)

(s15s0 / sentence
:modal ((ROOT :MODAL AUTH)
:AUTH :FullAff s15g))
:temporal ((PAST_REF :contained s15g)))

snt16 and he dumps all his pears into the basket
(s16d / dump-01
:ARG0 (s16p / person
:ref-person 3rd
:ref-number Singular)
:ARG1 (s16p2 / pear
:quant (s16a / all)
:poss s16p)
:aspect Performance
:goal (s16b / basket
:ref-number Singular)
:modstr FullAff)

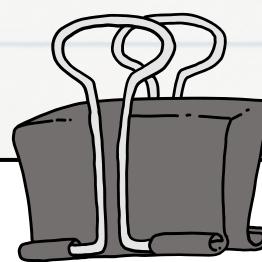
(s16s0 / sentence
:modal ((ROOT :MODAL AUTH)
:AUTH :FullAff s16d))
:temporal ((s15g :after s16d)))

snt17 and the basket's full,
(s17h / have-mod-91
:ARG1 (s17b / basket
:ref-number Singular)
:ARG2 (s17f / full)
:aspect State
:modstr FullAff)

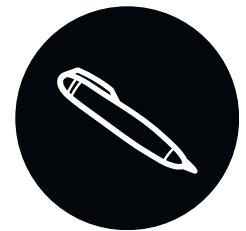
(s17s0 / sentence
:modal ((ROOT :MODAL AUTH)
:AUTH :FullAff s17h))
:temporal ((s16d :overlap s17h)))

snt18 and one of the pears drops down to the floor,
(s18d / drop-01
:ARG1 (s18p / pear
:quant 1
:ARG2-of (s18i2 / include-91
:ARG1 (s18p2 / pear
:ref-number Plural)))
:ARG4 (s18f / floor
:ref-number Singular)
:direction (s18d2 / down)
:aspect Performance
:modstr FullAff)

(s18s0 / sentence
:modal ((ROOT :MODAL AUTH)
:AUTH :FullAff s18d))
:temporal ((s16d :after s18d)))



Step 1



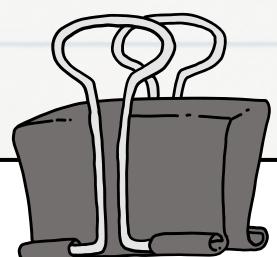
Annotate Events with Time Expressions

- Rule: If there is a time expression in the same line as the event, link the event to it.
- Consider time expressions in other lines after

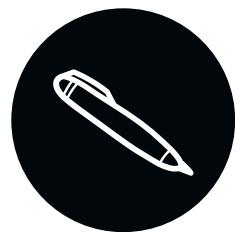
```
snt15 A-and u-h and then he gets down out of the tree,  
(s15g / get-05  
:ARG1 (s15p / person  
:ref-person 3rd  
:ref-number Singular)  
:ARG2 (s15d / down)  
:source (s15t / tree  
:refer-number Singular)  
:aspect Performance  
:temporal XXXXXXXXXX  
:modstr FullAff)
```

- What is the event?
- What should the temporal annotation be?





Step 1



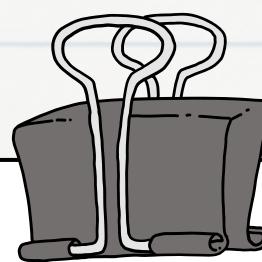
Annotate Events with Time Expressions

- Rule: If there is a time expression in the same line as the event, link the event to it.
- Consider time expressions in other lines after

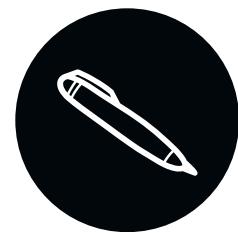
```
snt15 A-and u-h and then he gets down out of the tree,  
  (s15g / get-05  
  :ARG1 (s15p / person  
    :ref-person 3rd  
    :ref-number Singular)  
  :ARG2 (s15d / down)  
  :source (s15t / tree  
    :refer-number Singular)  
  :aspect Performance  
  :temporal (s15t2 / then)  
  :modstr FullAff)
```

- Event:
 - (s15g / get-05)
- Annotation
 - :temporal (s15t / then)





Step 2



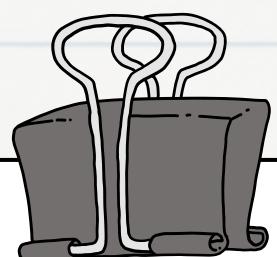
Consider Time Expressions in Other Lines

- Rule: If not linked in Step 1, link to another relevant time expression in the document.
- Rule: If the event has a :contained relation to any time expression in the text, annotate that relation

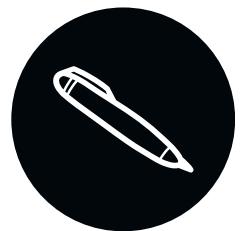
```
snt15 A-and u-h and then he gets down out of the tree,  
(s15g / get-05  
:ARG1 (s15p / person  
:ref-person 3rd  
:ref-number Singular)  
:ARG2 (s15d / down)  
:source (s15t / tree  
:refer-number Singular)  
:aspect Performance  
:temporal (s15t2 / then)  
:modstr FullAff)  
  
(s15s0 / sentence  
:modal ((ROOT :MODAL AUTH)  
          (AUTH :FullAff s15g))  
:temporal ((PAST_REF :contained s15g)))
```

- snt15 is an event contained since has a time expression contained within the text
- Annotation
 - :temporal ((PAST_REF :contained s15g))





Step 3



Link events to Other Events

- Rule: If not linked to time expressions, link to a preceding event that fits criteria.
- Criteria:
 - Parent event is a Process.
 - Compatible modal annotation.
 - Clear temporal relation.

```
snt15  A-and u-h and then he gets down out of the tree,  
(s15g / get-05  
  
snt16  and he dumps all his pears into the basket  
(s16d / dump-01  
  :ARG0 (s16p / person  
    :ref-person 3rd  
    :ref-number Singular)  
  :ARG1 (s16p2 / pear  
    :quant (s16a / all)  
    :poss s16p)  
  :aspect Performance  
  :goal (s16b / basket  
    :ref-number Singular)  
  :modstr FullAff)  
  
(s16s0 / sentence  
  :modal ((ROOT :MODAL AUTH)  
    (AUTH :FullAff s16d))  
  :temporal ((s15g :after s16d)))
```

```
snt15  A-and u-h and then he gets down out of the tree,  
(s15g / get-05
```

- What is the event in snt16?
- What is the temporal relation to event in snt15?
- How would you annotate it?



Step 3



Link events to Other Events

- Rule: If not linked to time expressions, link to a preceding event that fits criteria.
- Criteria:
 - Parent event is a Process.
 - Compatible modal annotation.
 - Clear temporal relation.

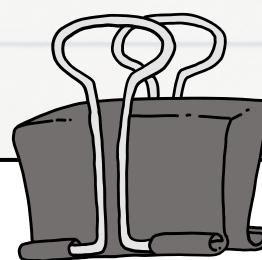
```
snt16 and he dumps all his pears into the basket
(s16d / dump-01
  :ARG0 (s16p / person
    :ref-person 3rd
    :ref-number Singular)
  :ARG1 (s16p2 / pear
    :quant (s16a / all)
    :poss s16p)
  :aspect Performance
  :goal (s16b / basket
    :ref-number Singular)
  :modstr FullAff)

(s16s0 / sentence
  :modal ((ROOT :MODAL AUTH)
    (AUTH :FullAff s16d))
  :temporal ((s15g :after s16d)))
```

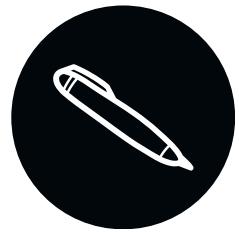
- Event:
 - (s16d / dump-01)
- Annotation
 - :temporal ((s15g :after s16d))
- Dumps pears (s16d) happens **after** get out of tree (s15g).

```
snt15 A-nd u-h and then he gets down out of the tree,
(s15g / get-05)
```





Step 3



Link events to Other Events

- Rule: If not linked to time expressions, link to a preceding event that fits criteria.
- Criteria:
 - Parent event is a Process.
 - Compatible modal annotation.
 - Clear temporal relation.

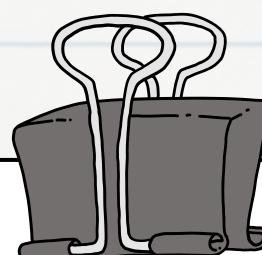
```
snt18 and one of the pears drops down to the floor,  
(s18d / drop-01  
:ARG1 (s18p / pear  
:quant 1  
:ARG2-of (s18i2 / include-91  
:ARG1 (s18p2 / pear  
:ref-number Plural)))  
:ARG4 (s18f / floor  
:ref-number Singular)  
:direction (s18d2 / down)  
:aspect Performance  
:modstr FullAff)  
  
(s18s0 / sentence  
:modal ((ROOT :MODAL AUTH)  
(:AUTH :FullAff s18d))  
:temporal ((s16d :after s18d)))
```

snt16 and he dumps all his pears into the basket
(s16d / dump-01)

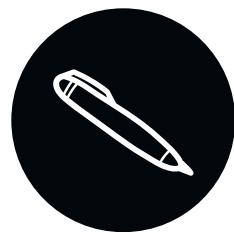
snt17 and the basket's full,
(s17h / have-mod-91)

- Event:
 - (s18d / drop01)
- Annotation
 - :temporal ([s16d :after s18d])

Since s17h is a State (not a Process), it does not fit the criteria for a good reference time. Therefore, s18d is linked to the next event back in the text s16d.



Step 4



Add Additional Annotations if Needed

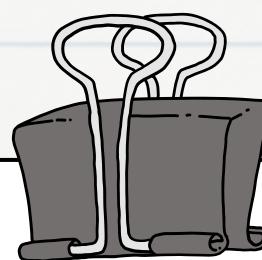
- Rule: If an event has a relation to a time expression, add another annotation if needed to specify its relation to another event.

```
snt17 and the basket's full,  
(s17h / have-mod-91  
  :ARG1 (s17b / basket  
          :ref-number Singular)  
  :ARG2 (s17f / full)  
  :aspect State  
  :modstr FullAff)  
  
(s17s0 / sentence  
  :modal ((ROOT :MODAL AUTH)  
          (AUTH :FullAff s17h))  
  :temporal ((s16d :overlap s17h)))
```

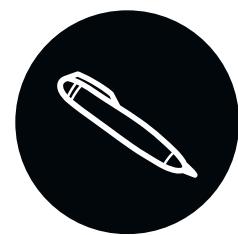
- Event:
 - (s17h / have-mod-91)
- Annotation
 - :temporal ((s16d :overlap s17h))

```
snt16 and he dumps all his pears into the basket  
(s16d / dump-01
```





Step 5



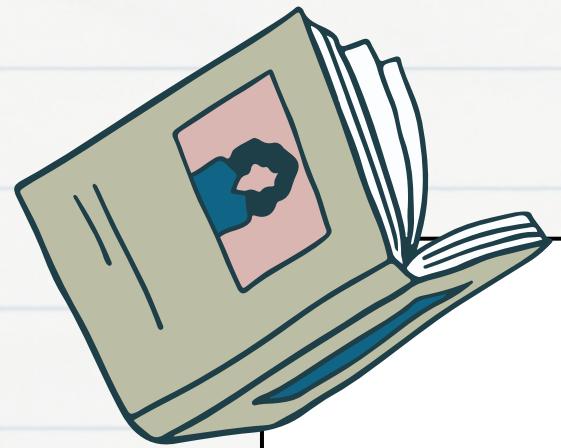
Link to Tense Metanodes if Necessary

- Rule: If no time expression or event is appropriate, link to a tense metanode (PAST_REF, PRESENT_REF, FUTURE_REF).

```
snt15 A-and u-h and then he gets down out of the tree,  
(s15g / get-05  
  :ARG1 (s15p / person  
    :ref-person 3rd  
    :ref-number Singular)  
  :ARG2 (s15d / down)  
  :source (s15t / tree  
    :refer-number Singular)  
  :aspect Performance  
  :temporal (s15t2 / then)  
  :modstr FullAff)  
  
(s15s0 / sentence  
  :modal ((ROOT :MODAL AUTH)  
    (AUTH :FullAff s15g))  
  :temporal ((PAST_REF :contained s15g)))
```

- snt15 is an event contained since has a time expression contained within the text
- Annotation
 - :temporal ((PAST_REF :contained s15g))





Special Cases: Temporal Relations

1. Complement-Taking Predicates

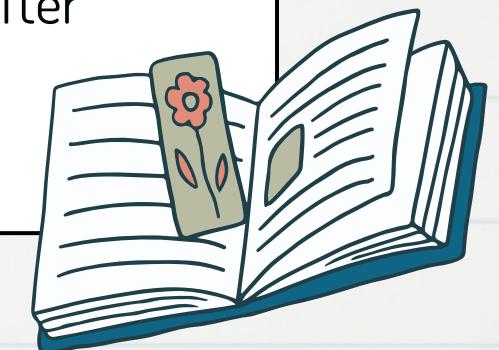
- Events linked by a :modpred relation at the sentence level should also be linked in the temporal dependency.
- Process:
 - Complement-taking predicate acts as the reference time for its complement.
 - Annotate the complement as a child of the complement-taking predicate in the temporal dependency.

2. Reporting Events

- Reported events linked to the reporting predicate.
- Process:
 - Link reported events to the reporting predicate.
 - If multiple reported events exist, consider their order.

3. Purpose Clauses

- Events in purpose clauses are linked to the main clause event with a temporal relation :after
- Example: "He went home to wash the dishes."
- Annotation:
 - :temporal ((PAST_REF :contained s1g) (s1g :after s1w))



Complement-taking Predicates

Complement with Overlap Relation



Sentence

“I saw him knock
on the door”

Annotation

- has “modpred” in sent
- Predicate: (s1s/ see-01)
- Complement: (s1k/ knock-01)
- Temporal: :temporal ((PAST_REF
:contained s1s) (s1s :overlap s1k))

I saw him knock on the door.

(s1s/ see-01

:ARG0 (s1p/ person
:ref-person 1st
:ref-number Singular)

:ARG1 (s1k/ knock-01

:ARG0 (s1p2/ person
:ref-person 3rd
:ref-number Singular)

:ARG1 (s1d/ door
:ref-number Singular))

:aspect Performance
:modpred s1s)

:aspect State
:modstr FullAff)

(s/ sentence

:temporal ((PAST_REF :contained s1s)
(s1s :overlap s1k)))

Complement-taking Predicates

Complement with after Relation



Sentence

"I want to cook dinner."

Annotation

- has “modpred” in sent
- Predicate: (s1w/ want-01)
- Complement: (s1c/ cook-01)
- Temporal: :temporal ((PRESENT_REF
:contained s1w) [s1w :after s1c]))

I want to cook dinner.

(s1w/ want-01)

:ARG0 (s1p/ person

:ref-person 1st

:ref-number Singular)

:ARG1 (s1c/ cook-01

:ARG0 s1p

:ARG1 (s1d/ dinner)

:aspect Performance

:modpred s1w)

:aspect State

:modstr FullAff)

(s/ sentence

:temporal ((PRESENT_REF :contained s1w))

(s1w :after s1c)))

Complement-taking Predicates

Multiple Complements

Sentence

"I want to go to the city
and visit a museum."

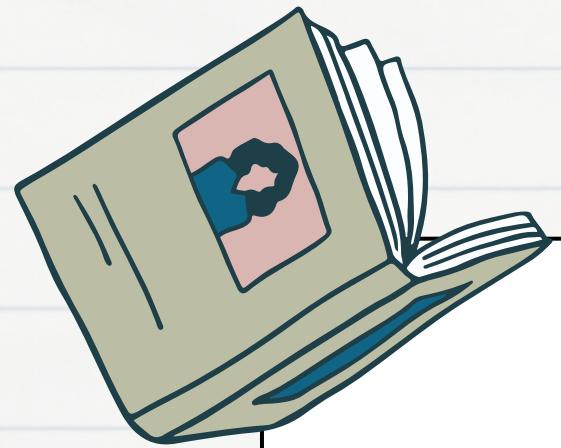
Rules

- If multiple complements exist, consider their order.
- Only one complement is annotated as the child of the main predicate.
- Remaining complements are linked to other complements.

Annotation

- has "modpred" in sent
- Predicate: (s1w/ want-01)
- Complement: (s1g/ go-01), (s1v/ visit-01)
- Temporal: :temporal ((PRESENT_REF :contained s1w)
:contained s1w) (s1w :after s1c)

```
I want to go to the city and visit, a museum.  
(s1w/ want-01  
:ARG0 (s1p/ person  
:ref-person 1st  
:ref-number Singular)  
:ARG1 (s1g/ go-01  
:ARG1 s1p  
:ARG4 (s1c/ city  
:ref-number Singular)  
:aspect Performance  
:modpred s1w)  
:aspect State  
:modstr FullAff)  
  
(s/ sentence  
:temporal ((PRESENT_REF :contained s1w)  
:contained s1w) (s1w :after s1c))
```



Special Cases: Temporal Relations

1. Complement-Taking Predicates

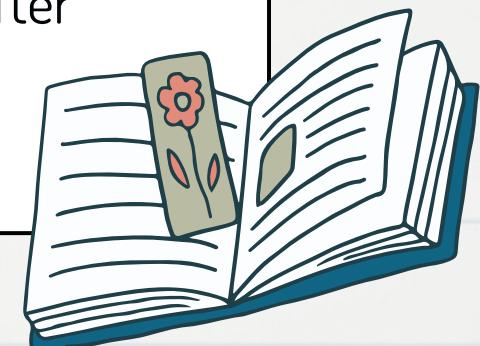
- Events linked by a :modpred relation at the sentence level should also be linked in the temporal dependency.
- Process:
 - Complement-taking predicate acts as the reference time for its complement.
 - Annotate the complement as a child of the complement-taking predicate in the temporal dependency.

2. Reporting Events

- Reported events linked to the reporting predicate.
- Process:
 - Link reported events to the reporting predicate.
 - If multiple reported events exist, consider their order.

3. Purpose Clauses

- Events in purpose clauses are linked to the main clause event with a temporal relation :after
- Example: "He went home to wash the dishes."
- Annotation:
 - :temporal ((PAST_REF :contained s1g) (s1g :after s1w))



Reporting Events

Multiple reported events



Sentence

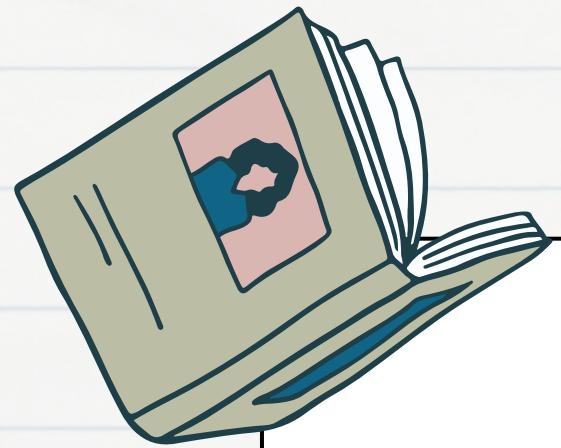
"Magdalena said she arrived home, ate dinner, and will meet us at the theater."



Annotation

- Predicate: (s1s/ say-01)
- Reported Events: (s1a2/ arrive-01), (s1e/ eat-01), (s1m/ meet-03)
- Temporal: :temporal ((PAST_REF :contained s1s) (s1s :before s1e) (s1e :before s1a) (s1s :after s1m))

```
Magdalena said she arrived home, ate dinner, and will meet us at the theater.  
(s1s/ say-01  
:ARG0 (s1p/ person  
:name (s1n/ name :op1 "Magdalena"))  
:ARG1 (s1a/ and  
:op1 (s1a2/ arrive-01  
:ARG1 s1p  
:ARG4 (s1h/ home)  
:aspect Performance  
:modstr FullAff  
:QUOT s1s)  
:op2 (s1e/ eat-01  
:ARG0 s1p  
:ARG1 (s1d/ dinner)  
:aspect Performance  
:modstr FullAff  
:QUOT s1s)  
:op3 (s1m/ meet-03  
:ARG0 s1p  
:ARG1 (s1p2/ person  
:ref-person 1st  
:ref-number Plural)  
:place (s1t/ theater  
:ref-number Singular)  
:aspect Performance  
:modstr FullAff  
:QUOT s1s))  
:aspect Performance  
:modstr FullAff)  
  
(s/ sentence  
:temporal ((PAST_REF :contained s1s)  
(s1s :before s1e)  
(s1e :before s1a)  
(s1s :after s1m)))
```



Special Cases: Temporal Relations

1. Complement-Taking Predicates

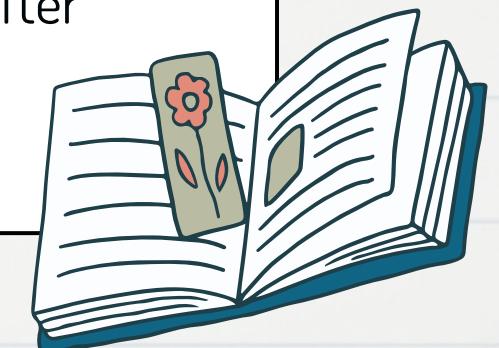
- Events linked by a :modpred relation at the sentence level should also be linked in the temporal dependency.
- Process:
 - Complement-taking predicate acts as the reference time for its complement.
 - Annotate the complement as a child of the complement-taking predicate in the temporal dependency.

2. Reporting Events

- Reported events linked to the reporting predicate.
- Process:
 - Link reported events to the reporting predicate.
 - If multiple reported events exist, consider their order.

3. Purpose Clauses

- Events in purpose clauses are linked to the main clause event with a temporal relation :after
- Example: "He went home to wash the dishes."
- Annotation:
 - :temporal ((PAST_REF :contained s1g) (s1g :after s1w))



Purpose Clauses

Example



Sentence

He went home (in order)
to wash the dishes.



Annotation

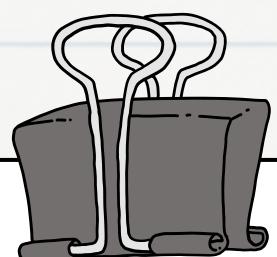
- Main Event: (s1g/ go-01)
- Purpose Event: (s1w/ wash-01)
- Temporal:
 - :temporal ((PAST_REF :contained s1g)
(s1g :after s1w))

He went home (in order) to wash the dishes.
(s1g/ go-01)

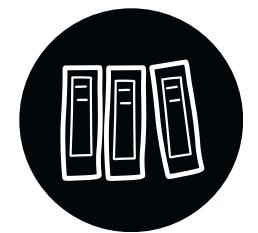
linked
with
:after

:ARG1 (s1p/ person
:ref-person 3rd
:ref-number Singular)
:ARG4 (s1h/ home)
:purpose (s1w/ wash-01
:ARG0 s1p
:ARG1 (s1d/ dish
:ref-number Plural)
:aspect Performance
:modstr FullAff)
:aspect Performance
:modstr FullAff)

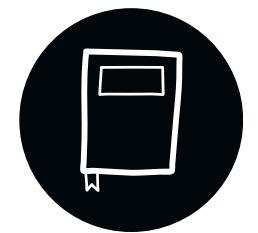
(s/ sentence
:temporal ((PAST_REF :contained s1g)
(s1g :after s1w)))



Activity 4



Get into your groups



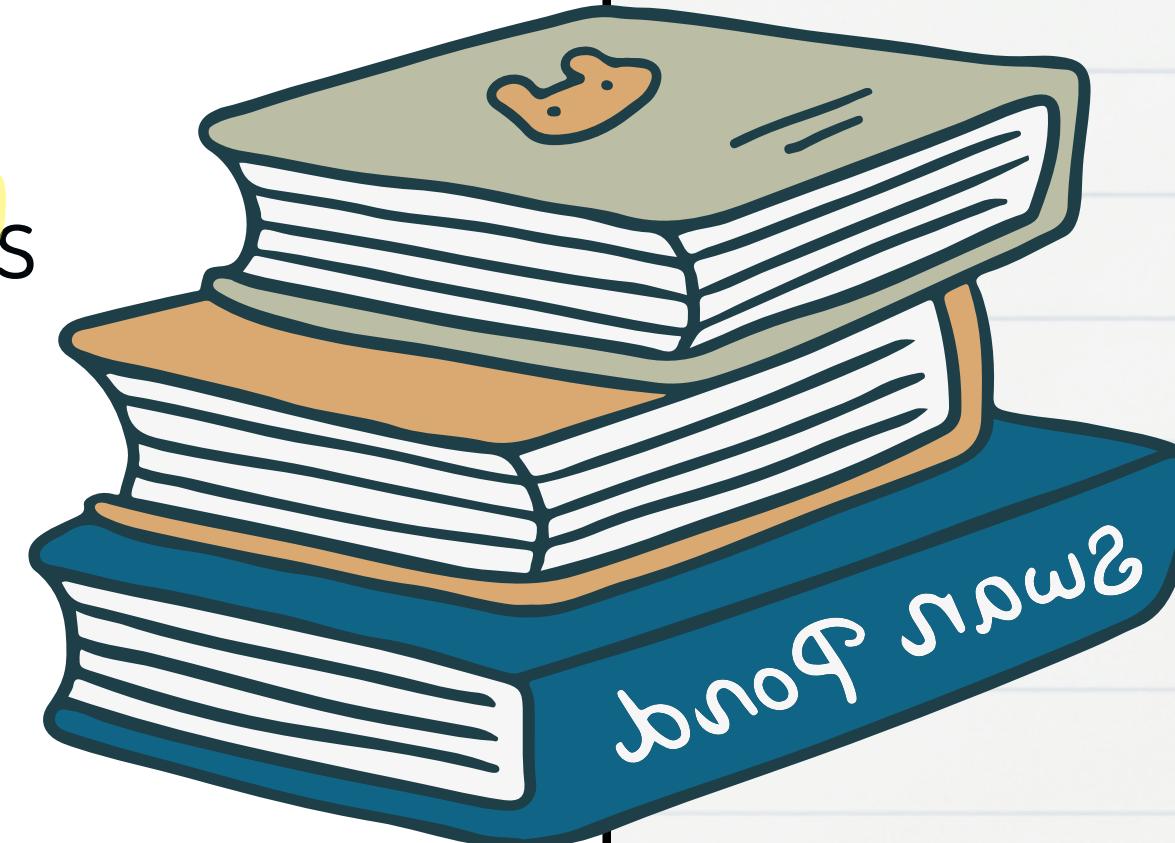
**Complete Activity for Temporal
Relations**

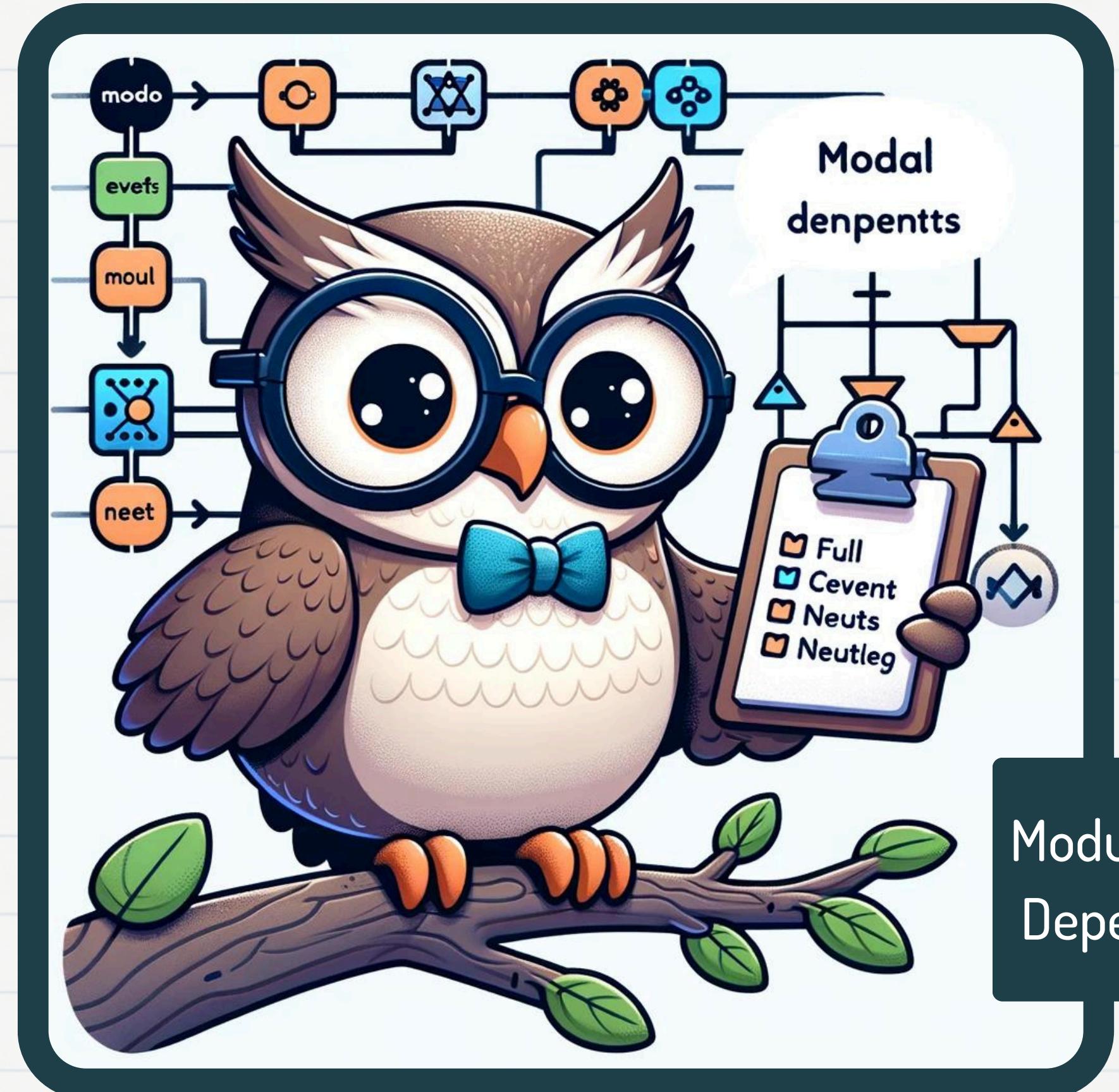
- Feel free to ask questions or for help
- Benet will pass out worksheets



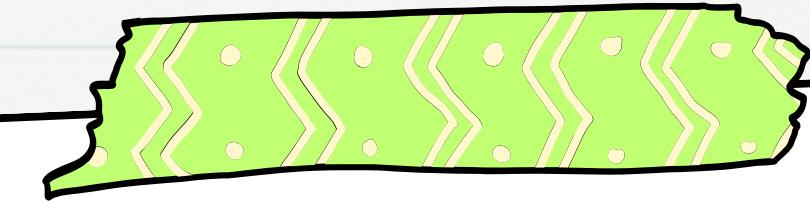
Overview

1. Discourse Relations
2. Coreference
3. Temporal Dependencies
4. Modal Dependencies





Modulo the Modal
Dependency Owl



Mode Dependencies

Overview

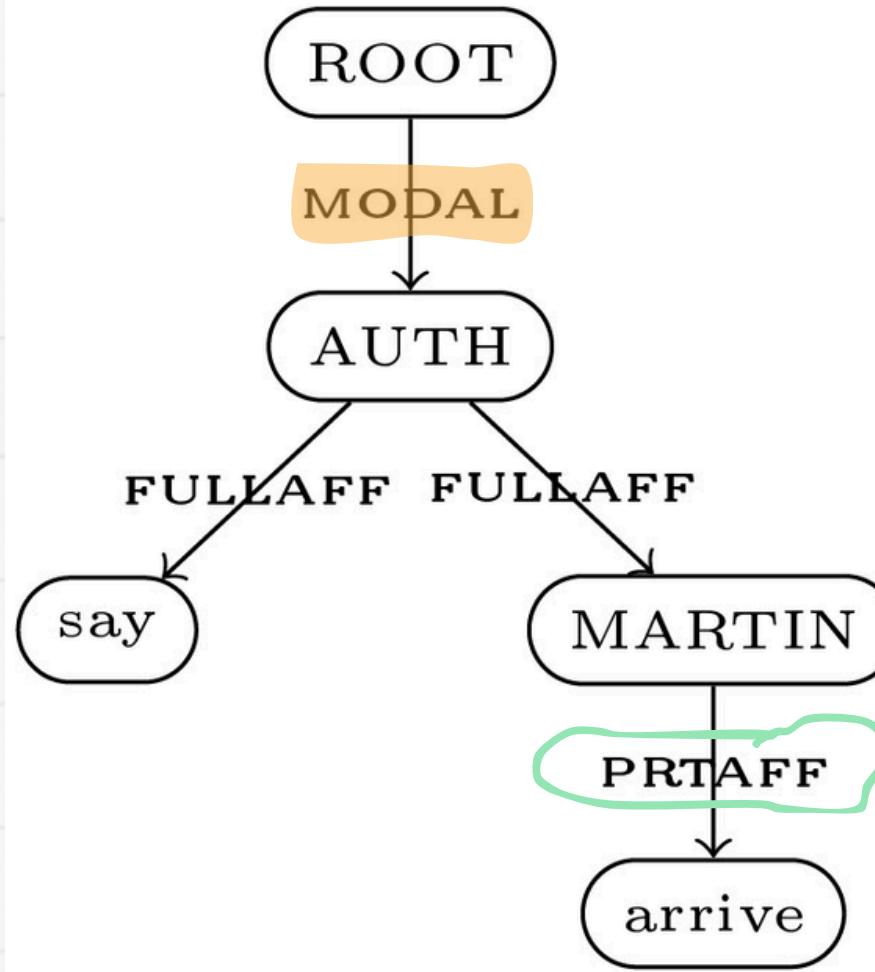
- UMR captures **modal strength** and **polarity** as a **dependency structure**, involving events or conceivers.
- The structure shows **how certain** a **conceiver** is about an **event occurring**.
- Annotators **do not** need to construct the structure directly;
 - it is built automatically through modal/polarity annotation and participant roles.



Modal Strength & Polarity

Key Points

- Modal strength and polarity are represented as edges in the dependency structure.
- Modal strength values indicate the degree of certainty about an event.
- :modstr
 - annotation captures the modal strength of events.
- :quot
 - value shows an event is reported.

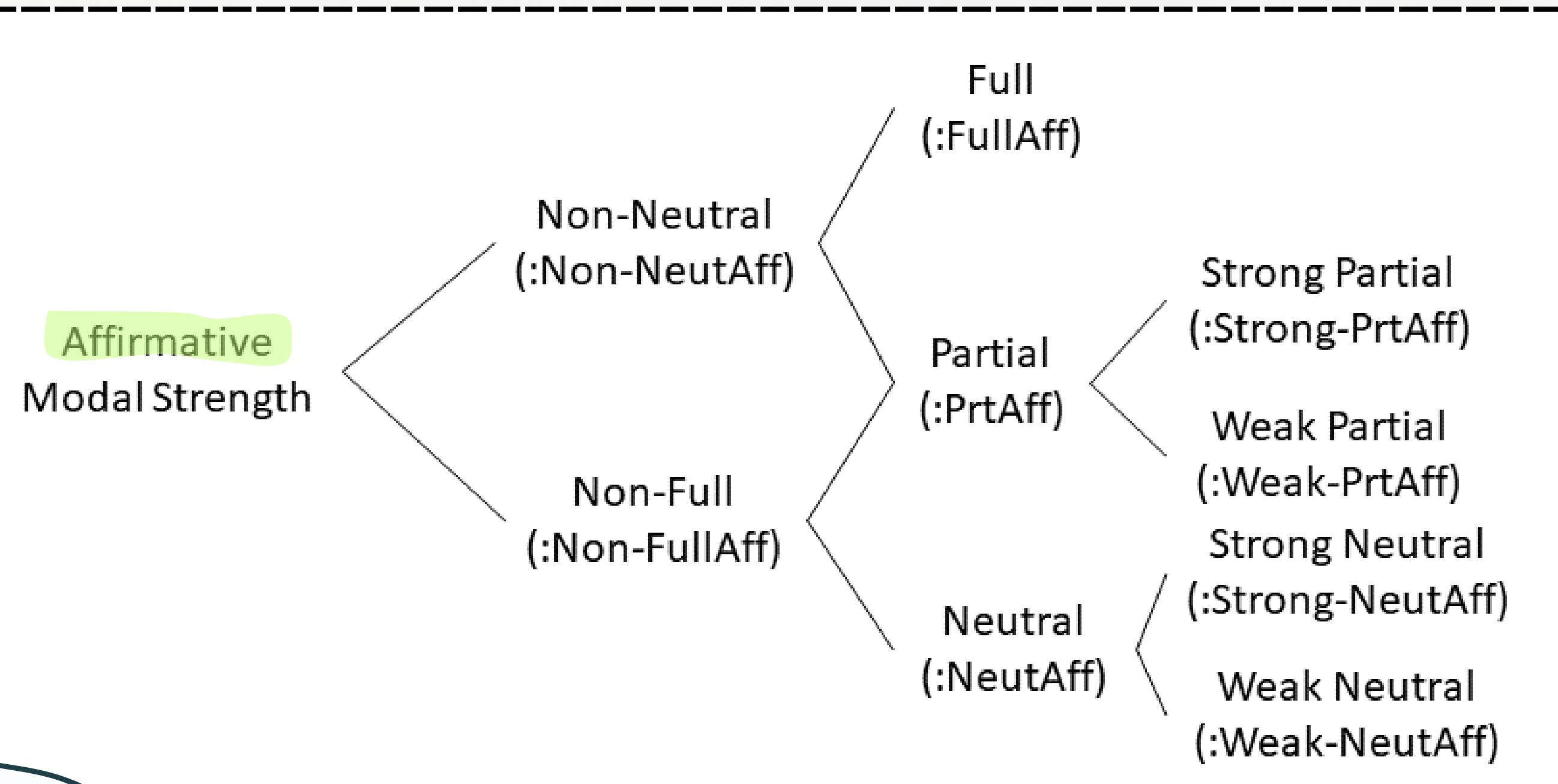


Modal constants for first slot:

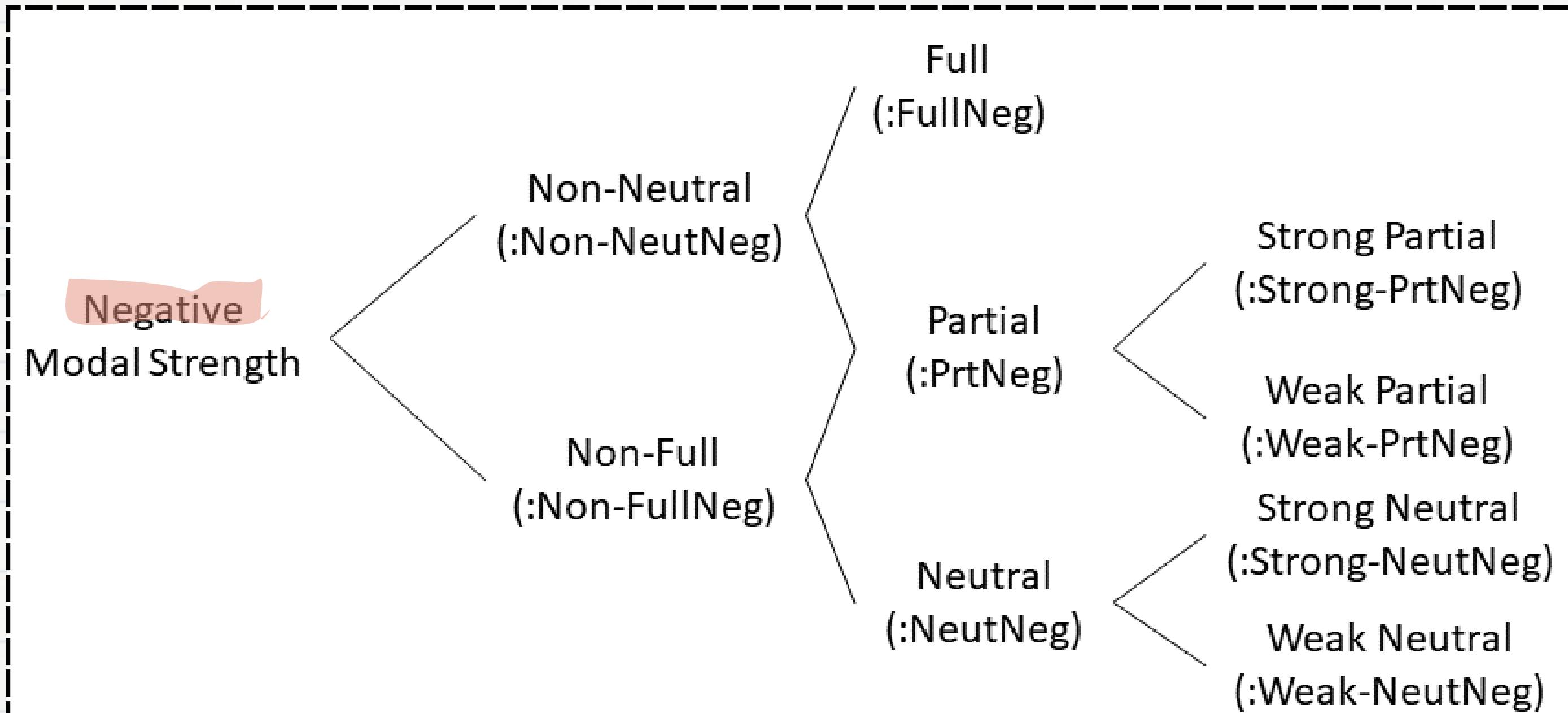
- AUTH -- author
- ROOT -- root of the chain, establishes AUTH-- AUTH establishes all other asserters

Martin said that the package has probably already arrived.
(s/ say-01
:ARG0 (p/ person
:name (n/ name :op1 "Martin")
:ARG1 (a/ arrive-01
:ARG1 (p/ package
:ref-number Singular)
quot: s
modstr: PrtAff)
modstr: Aff)
(s/ sentence
:temporal ((PAST_REF :contained s1s)
(s1s :before s1a))
:modal ((AUTH :FullAff s1p)
(AUTH :FullAff s1s)
(s1p :PrtAff s1a)))

What are the :modstr values?



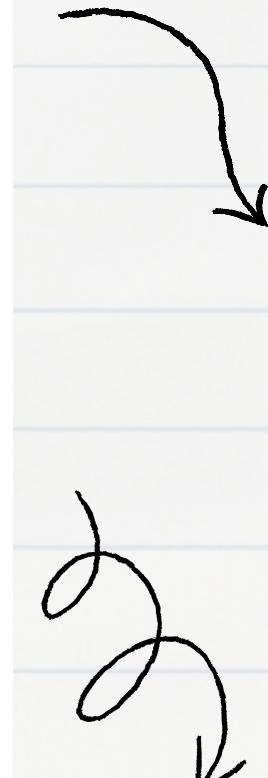
What are the :modstr values?





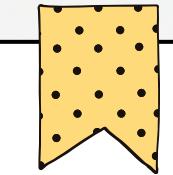
Modal Strength Values

Label	Value
FullAff	full strength, affirmative polarity
PrtAff	partial strength, affirmative polarity
NeutAff	neutral strength, affirmative polarity
NeutNeg	neutral strength, negative polarity
PrtNeg	partial strength, negative polarity
FullNeg	full strength, negative polarity



Examples

- **Full:** The cat already ate breakfast.
- **Partial:** The cat probably already ate breakfast.
- **Neutral:** The cat might have already eaten breakfast.



FullAff : full affirmative support; complete certainty that the event occurs (CT+)
PrtAff : partial affirmative support; there is strong, but not definitive certainty that the event occurs (PR+)
NeutAff : affirmative neutral support; there is neutral certainty that the event occurs/doesn't occur; event is expressed positively (PS+)
NeutNeg : negative neutral support; there is neutral certainty that the event occurs/doesn't occur; negation of event is expressed (PS-)
PrtNeg : partial negative support; there is strong but not definitive certainty that the event does not occur (PR-)
FullNeg : full negative support; complete certainty that the event does not occur (CT-)



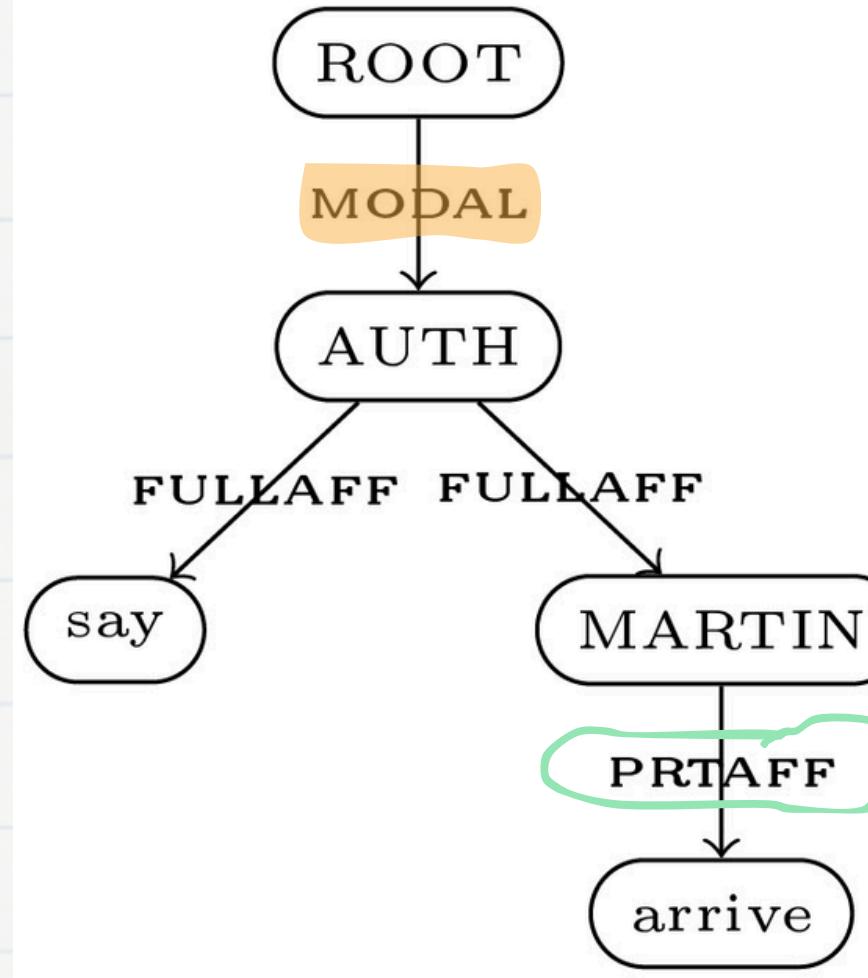
- apply to all events
- determine level of certainty



Modal Strength & Polarity

Key Points

- Modal strength and polarity are represented as edges in the dependency structure.
- Modal strength values indicate the degree of certainty about an event.
- :modstr
 - annotation captures the modal strength of events.
- :quot
 - value shows an event is reported.



What is :quot?

Martin said that the package has probably already arrived.
(s/ say-01
:ARG0 (p/ person
:name (n/ name :op1 "Martin")
:ARG1 (a/ arrive-01
:ARG1 (p/ package
:ref-number Singular)
quot: s
modstr: PrtAff)
modstr: Aff)
(s/ sentence
:temporal ((PAST_REF :contained s1s)
(s1s :before s1a))
:modal ((AUTH :FullAff s1p)
(AUTH :FullAff s1s)
(s1p :PrtAff s1a)))

:Quot

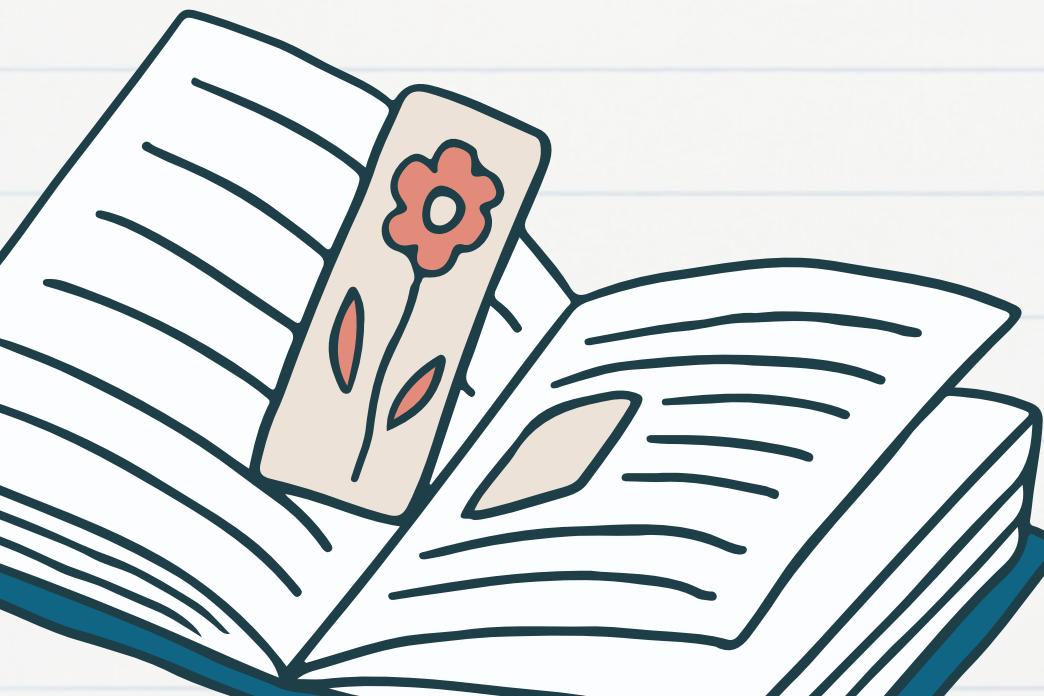
Rules

For each predicate that documents an utterance of some sort, add a :quot relation to the utterance argument

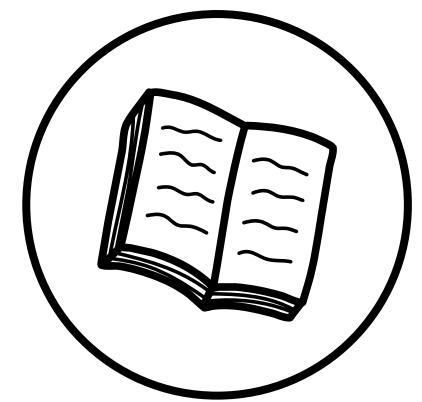
Example

if the predicate is 'say-01', then the ARG1-utterance gets a :quot argument added to it. Its value is the variable of the 'say' predicate in the graph.

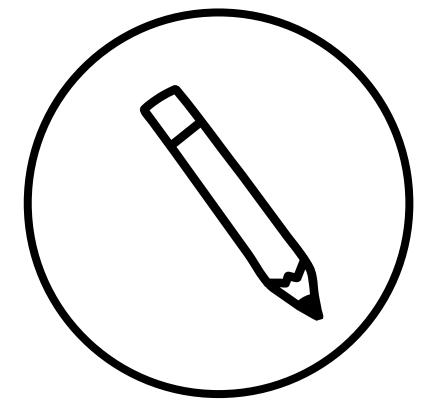
```
Mary didn't say that she went to Santa Fe.  
(s/ say-01  
    :ARG0 (p/ person  
            :name (n/ name :op1 "Mary"))  
    :ARG1 (g/ go-01  
            :ARG1 p  
            :ARG4 (c/ city  
                    :wiki "Santa_Fe,_New_Mexico"  
                    :name (n2/ name  
                            :op1 "Santa"  
                            :op2 "Fe))  
            :aspect Performance  
            :modstr FullAff  
            :quot s  
    :aspect Performance  
    :modstr FullNeg)  
(s/ sentence  
    :temporal ((PAST_REF :contained s1s)  
                (s1s :before s1g))  
    :modal ((AUTH :FullNeg s1s)  
            (AUTH :FullNeg s1p)  
            (s1p :FullAff s1g)))
```



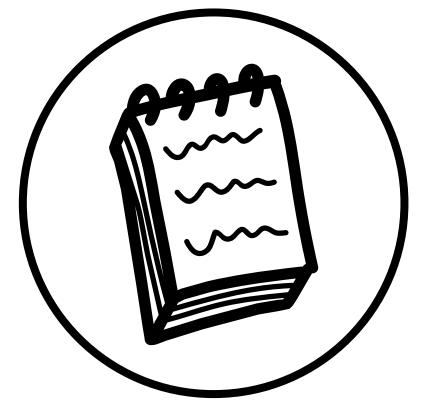
Activity 5



Get back into your groups
and start on the Modal
Dependencies Activity



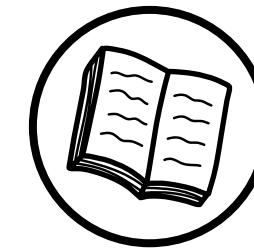
Consider examples of how
different Modstr values are
used in Activity 1 on the
sheet



We will share once you all
have finished!

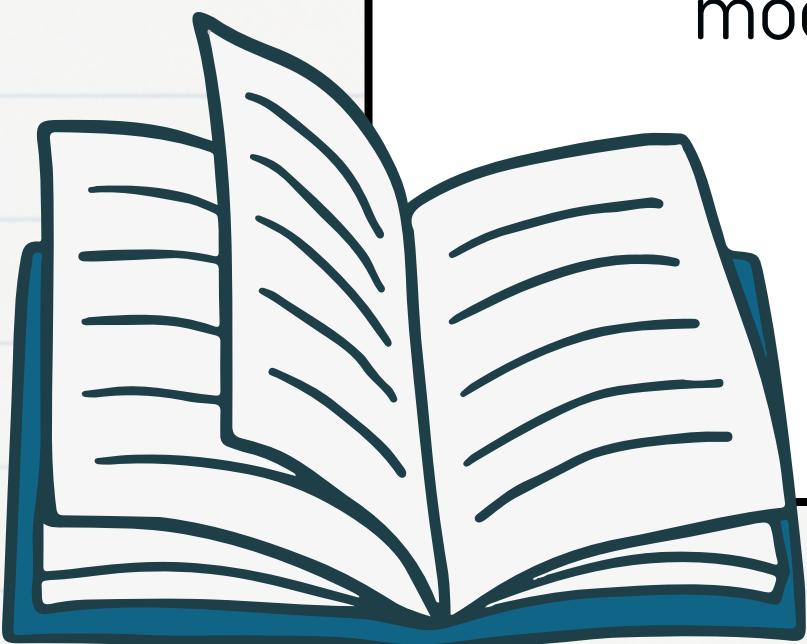


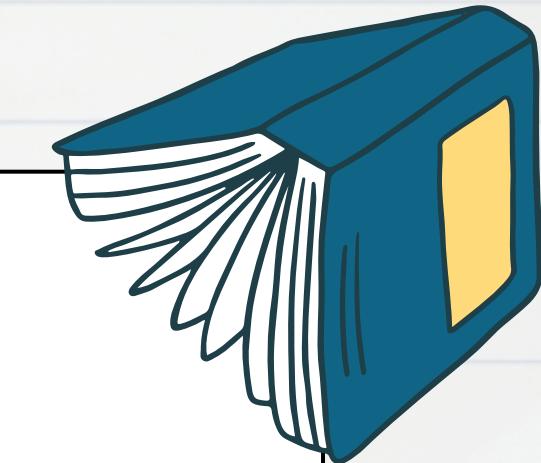
Stage 0 Annotations



Overview:

- Stage 0 annotations form the foundation of the modal dependency structure.
- They include **modal strength values** and **dependency relations**.
- These annotations are essential for constructing a detailed and accurate modal representation of the text.





Manual and Automated Step 0

Manual Annotations:

- Annotators manually perform Stage 0 annotations:
 - **:modstr Values:** FullAff, PrtAff, NeutAff, FullNeg, PrtNeg, NeutNeg.
 - Dependency Relations:
 - **:modpred:** Link between a modal event and its modalized event.
 - **:quot:** Link between a reporting event and the reported event.
 - **Purpose Clauses:** Annotated with :purpose.
 - **Conditionals:** Annotated with :condition or have-condition-91 node.

Automation Assistance:

- Pre-existing Lexicon:
 - Automatically assign :modstr values based on known modal verbs.
- Rule-based Systems:
 - Detect common modal and reporting verbs to annotate :modpred and :quot relations.
- Template Matching:
 - Match sentences against templates for common modal constructions.
 - Suggest appropriate annotations.



Detailed Steps for Annotators

01

Identify Event Type:

- Determine if the event is under the scope of a modal or reporting verb.

02

Assign :modstr Value:

- Use the appropriate modal strength value.

03

Annotate Dependency Relations:

- Link events with :modpred or :quot as needed.

04

Leverage Automation:

- Utilize system suggestions for :modstr values and dependency relations.
- Validate and adjust as necessary.



Special Cases

Evidential Justification

Concept

- Evidential justification refers to the type of evidence or support a conceiver has for an event.
- It can be direct (sensory perception) or indirect (inference, hearsay).

Direct Justification: author has direct knowledge of event, so feed gets annotated with Aff modal strength

Indirect Justification: the must signals the author is inferring, so fed is annotated with Prt

(I saw) Mary feed the cat.
(s/ see-01
:ARG0 (p/ person
:ref-person 1st
:ref-number Singular)
:ARG1 (f/ feed-01
:ARG0 (p2/ person
:name (n/ name :op1 "Mary"))
:ARG2 (c/ cat
:ref-number Singular)
:aspect Performance
:modstr FullAff)
:aspect State
:modstr FullAff)

Mary must have fed the cat.
(f/ feed-01
:ARG0 (p/ person
:name (n/ name :op1 "Mary"))
:ARG2 (c/ cat
:ref-number Singular)
:aspect Performance
:modstr PrtAff)

Special Cases

Future Events and Deontic Modality

Concept

- For events presented as potentially happening in the **future**, the **:modstr** value reflects the likelihood of the event occurring.
- This includes predictive future events, intentions, commands, and permissions.

these are “deontic meanings”

Predictive Future Event is expected with high certainty to ‘go’ gets FullAff

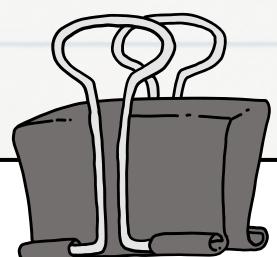
Intentions and Commands: get PrtAff or PrtNeg, in this case its PrtAff

Permissions: get NeutAff or NeutNeg. indicating speaker is neutral about event

```
I will go to Santa Fe.  
(g/ go-01  
  :ARG1 (p/ person  
    :ref-person 1st  
    :ref-number Singular)  
  :ARG4 (c/ city  
    :wiki "Santa_Fe,_New_Mexico"  
    :name (n/ name  
      :op1 "Santa"  
      :op2 "Fe"))  
  :aspect Performance  
  :modstr FullAff)
```

```
You must go to Santa Fe.  
(g/ go-01  
  :ARG1 (p/ person  
    :ref-person 2nd  
    :ref-number Singular)  
  :ARG4 (c/ city  
    :wiki "Santa_Fe,_New_Mexico"  
    :name (n/ name  
      :op1 "Santa"  
      :op2 "Fe"))  
  :aspect Performance  
  :modstr PrtAff  
  :mode Imperative)
```

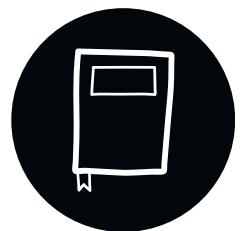
```
You can go to Santa Fe.  
(g/ go-01  
  :ARG1 (p/ person  
    :ref-person 2nd  
    :ref-number Singular)  
  :ARG4 (c/ city  
    :wiki "Santa_Fe,_New_Mexico"  
    :name (n/ name  
      :op1 "Santa"  
      :op2 "Fe"))  
  :aspect Performance  
  :modstr NeutAff)
```



Activity 5

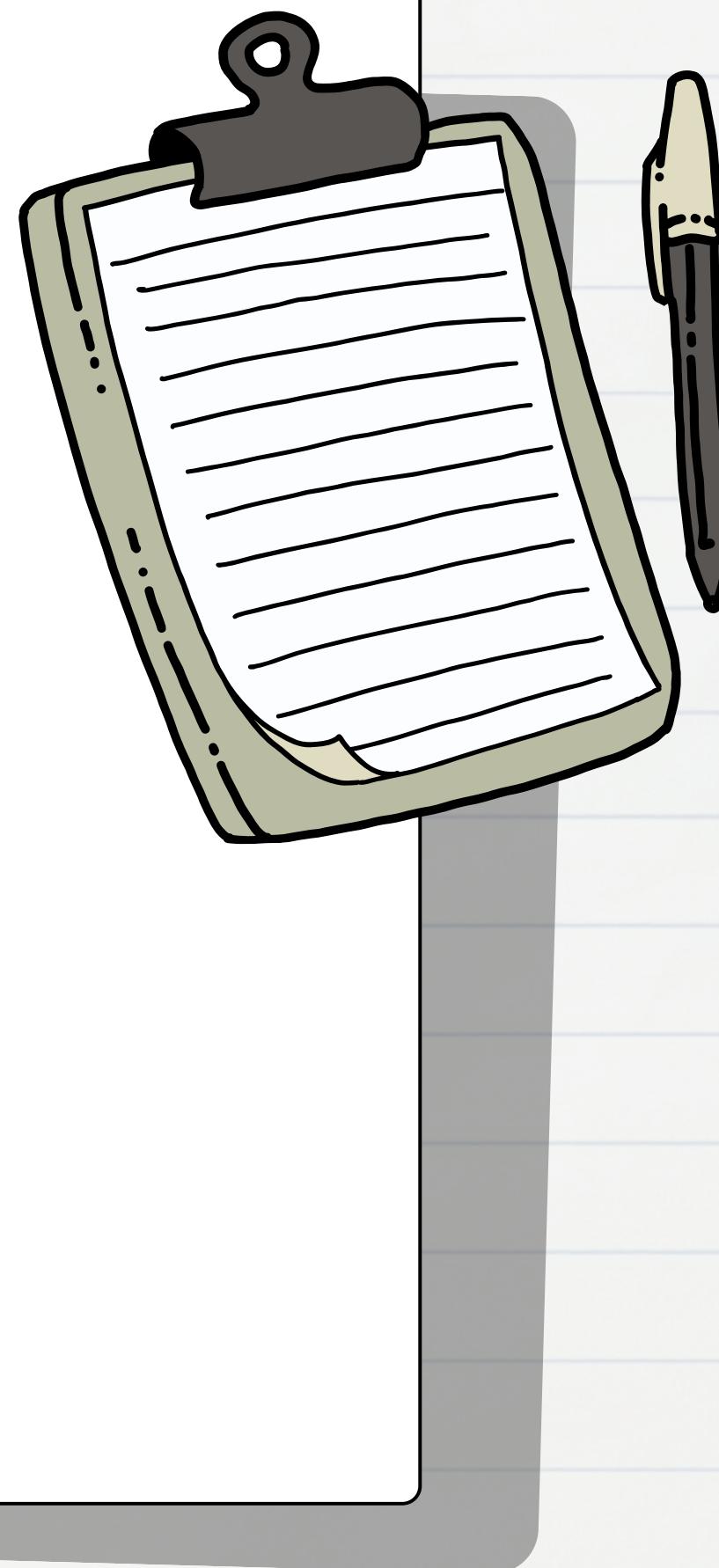


Get into your groups



**Complete Activity for Modal
Dependency**

- Feel free to ask questions or for help
- Benet will pass out worksheets



Thank You!

Questions?

