Coreference

Activity 1 - Complete the UMR Graph

<u>Instructions:</u> Fill in the blank for each coref relation.

1. Coreference "Man" for Sentence 5 and Sentence 6

```
# :: snt5          :: snt5 and there's a man at the top of the mountain ,
# sentence level graph:
(s5h / have-place-91
  :ARG1 (s5m / man
        :refer-number Singular)
  :ARG2 (s5t / top
        :Part-of (s5l / ladder
              :refer-number Singular)
        :refer-number Singular)
  :Aspect State
  :MODSTR FullAff)
# :: snt6          :: snt6 you can't see him yet .
# sentence level graph:
(s6s / see-01
  :ARG0 (s6p / person
        :refer-person 2nd
        :refer-number Singular)
  :ARG1 (s6p2 / person
        :refer-person 3rd
        :refer-number Singular)
  :mod (s6y / yet)
  :Aspect State
  :MODSTR NeutNeg
  :polarity -)

# document level annotation:
(s6s0 / sentence
  :temporal ((s5e :overlap s6s))
  :modal ((ROOT :MODAL AUTH)
        (AUTH :FullNeg s6s))
  :coref ((s2p2 :same-entity s6p)
        (s5m :same-entity s6p2)))
```

2. Use the :subset-of relation

```
# Sentence 1:   One arrest took place in the Netherlands and another in Germany.
(a/ and
    :op1 (a2/ arrest-01
            :quant 1
        :location (c/ country
                    :wiki "Netherlands"
              :name (n/ name
                          :op1 "Netherlands"))
            :aspect Performance
            :modstr FullAff)
    :op2 (a3/ arrest-01
        :location (c2/ country
                    :wiki "Germany"
              :name (n2/ name
                          :op1 "Germany"))
        :mod (a4/ another)
            :aspect Performance)
            :modstr FullAff)

## DOCUMENT LEVEL
(s/ sentence
        :temporal ((DCT :before a2)
                   (DCT :before a3))
        :modal ((AUTH :FullAff a2)
                (AUTH :FullAff a3)))

# Sentence 2:     The arrests were ordered by anti-terrorism judge fragnoli.
(o/ order-01
    :ARG0 (p/ person
            :wiki -
        :name (n/ name
                    :op1 "Fragnoli")
        :ARG0-of (o2/ oppose-01
            :ARG1 (t/ terrorism))
        :ARG1-of (h/ have-role-91
            :ARG3 (j/ judge-01)))
    :ARG2 (a/ arrest-01
            :aspect Process
            :quot o)
    :aspect Performance
```

```
    :modstr FullAff)

## DOCUMENT LEVEL ANNOTATION
 (s2/ sentence
        :temporal (s2a :before s2o)
        :modal ((AUTH :FullAff s2o)
                (AUTH :FullAff s2p)
                (s2p :PrtAff s2a))

        :coref ((s2a :subset-of s1a2)
                (s2a :subset-of s1a3)))
```

3. Identify the underlying event being coreferenced by different linguistic expressions

```
## Sentence 1:  The Three Gorges project has recently introduced the first foreign capital.
(i/ introduce-01
    :ARG0 (p/ project
          :wiki "Three_Gorges_Dam"
          :name (n/ name
                :op1 "The"
                :op2 "Three"
                :op3 "Gorges")
    :ARG1 (c/ capital
        :mod (f/ foreign)
        :ord (o/ ordinal-entity
                :value 1))
    :temporal (r2/ recent)
    :aspect Performance
    :modstr FullAff)

(s1/ sentence
        :temporal ((PAST_REF :includes s1r2)
                  (s1r2 :includes s1i))
        :modal (AUTH :FullAff s1i))
```

Sentence 2: The loan is an export credit *provided* to the Three Gorges project by the Canadian government.

```
(i/ identity-91
        :ARG1 (t/ thing
                :ARG1-of (l/ loan)
        :ARG2 (c2/ credit
                :mod (e/ export-01)
                :ARG1-of (p/ provide
                        :ARG0 (g/ government-organization
                                :ARG0-of (g/ govern-01
                                        :mod (c3/ country
                                                :wiki "Canada"
                                                :name (n2/ name :op1 "Canada"))))
                        :ARG2 (p2/ project
                                :wiki "Three_Gorges_Dam"
                                :name (n3/ name
                                        :op1 "Three"
                                        :op2 "Gorges"))
                        :aspect Performance
                        :modstr FullAff)
        :aspect State
        :modstr FullAff)

(s2/ sentence
  :temporal ((DCT :overlap s2i)
             (Future_Ref :includes s2u))
  :modal ((AUTH :FullAff s2i)
          (AUTH :FullAff s2p)
          (AUTH :FullAff s2u))

  :coref (s1i :same-event s2p))
```

Temporal Dependencies

Activity - Choosing temporal relations

`:contained` : child is entirely contained within the parent; parent begins before child and parent ends after child (Note: this is called 'Includes' in Zhang & Xue 2018). `:after` : child is after parent. `:before` : child is before parent. `:overlap` : child and parent overlap (either partially or fully).

Instruction: Fill in the blanks with the correct temporal relation and boxes with nodes.

1. Sentence 2

```
# :: snt2          :: snt2 U—m the s the scene opens up with um you see a tree ,
# sentence level graph:
(s2o2 / open-02
  :ARG1 (s2s3 / scene
        :refer-number Singular)
  :manner (s2s4 / see-01
          :ARG0 (s2p2 / person
                :refer-person 2nd
                :refer-number Singular)
          :ARG1 (s2t2 / tree
                :refer-number Singular)
          :Aspect State
          :MODSTR FullAff)
  :Aspect Performance
  :MODSTR FullAff)

# document level annotation:
(s2s0 / sentence

  :temporal ((PRESENT_REF :overlap s2o2)
           (s2o2 :overlap s2s4))
  :modal ((ROOT :MODAL AUTH)
        (AUTH :FullAff s2s4)
        (AUTH :FullAff s2o2)))
```

2. Use previous Sentence 2 and Sentence 4

```
# :: snt4          :: snt4 And there's a ladder coming out . . o of the tree ,
# sentence level graph:
(s4e / exist-91
  :ARG1 (s4c / come-33
        :ARG1 s4l
        :ARG3 (s4t / tree
              :refer-number Singular)
        :Aspect State
        :MODSTR FullAff)
  :ARG2 (s4l / ladder
        :refer-number Singular)
  :Aspect State
  :MODSTR FullAff)

# alignment:s4c: 5-5
s4e: 2-2
s4l: 4-4
s4t: 12-12

# document level annotation:
(s4s0 / sentence

  :temporal ((s2s4 :overlap s4e))

  :modal ((ROOT :MODAL AUTH)
        (AUTH :FullAff s4e))

  :coref ((s2t2 :same-entity s4t)))
```

3. Use previous Sentence 2 and Sentence 7

```
# :: snt7          :: snt7 A-nd then . . it shifts
# sentence level graph:
(s7s / shift-01
  :ARG1 (s7t / thing
        :refer-person 3rd
        :refer-number Singular)
  :temporal (s7t2 / then)
  :Aspect Performance
  :MODSTR FullAff)

# alignment:
s7t: 5-5
s7s: 6-6
s7t2: 2-2

# document level annotation:
(s7s0 / sentence

  :temporal ((s2o2 :after (s7s))

  :modal ((ROOT :MODAL (AUTH)
        (AUTH :FullAff (s7s))

  :coref ((s2s3 :same-entity (s7t)))
```

Modal Dependencies

Activity 2 - Picking :modstr and Completing Modal Annotation

Instructions: Pick the right :modstr values

1. You can go tonight. _____
2. He forbids you from leaving. _____
3. She will go to the beach. _____
4. The birds might be hungry. _____
5. He is probably not upset. _____
6. Janet thinks the house flooded. _____

Instructions: Complete the :modal part of the graph.

They probably decided to leave on Monday.

```
(d/ decide-01
        :ARG0 (p/ person
                :ref-person 3rd
                :ref-number Plural)
        :ARG1 (l/ leave-01
                :ARG0 p
                :temporal (d/ date-entity
                        :weekday (m/ Monday))
                :aspect Performance
                :modpred d)
        :aspect Performance
        :modstr PrtAff)
(s/ sentence
        :temporal ((PAST_REF :contained s1d)
                (s1m :contained s1l))
        :modal ((AUTH :PrtAff s1p)
                (s1p :FullAff s1d)
                (s1d :Unsp s1l)))
```

1. Look at this example sentence about Edmund Pope

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 s3d)
             (AUTH :FullAff s3p)
           (s3p :FullNeg s3d2)
             (s3d :Unsp s3d2))
    :coref(s2p :same-entity s3p))
```