NLP Coursework 3: Coreference with the Stanford CORE NLP System

Lyapina Ekaterina
Student ID: 160984099

Task 1

The system can carry out the following types of interpretation (annotators):

- Tokenization - splits the text into words
- Sentence splitting - splits a sequence of tokens into sentences
- Parts of speech - labels tokens with their POS tag
- Lemmatization - generates the word lemmas for all tokens in the corpus.
- Named entity recognition - recognizes named (PERSON, LOCATION, ORGANIZATION, MISC), numerical (MONEY, NUMBER, ORDINAL, PERCENT), and temporal (DATE, TIME, DURATION, SET) entities
- RegexNER (Named Entity Recognition) - implements a rule-based NER over token sequences using regular expressions to provide a simple framework to incorporate NE labels that are not annotated in traditional NL corpora
- Constituency parsing - provides full syntactic analysis, using both the constituent and the dependency representations
- Dependency parsing - provides a fast syntactic dependency parser.
- Coreference resolution - implements nominal coreference resolution
- Natural logic - marks quantifier scope and token polarity, according to natural logic semantics
- Open information extraction - extracts open-domain relation triples, representing a subject, a relation, and the object of the relation
- Sentiment - implements Socher et al's sentiment model
- Relation extraction - finds relations between two entities
- Quote annotator - picks out quotes delimited by "or" from a text
- CleanXML annotator - removes xml tokens from the document
- True case annotator - recognizes the true case of tokens in text where this information was lost, e.g., all upper case text
- Entity mentions annotator - generates a list of the mentions, identified by NER, it produces whole entity mentions
- UD features - labels tokens with their Universal Dependencies universal part of speech (UPOS) and features
- Adding a new annotator - adds custom annotators

It handles languages:

- Arabic
- Chinese
- English
- French
- German
- Spanish

It is used for:

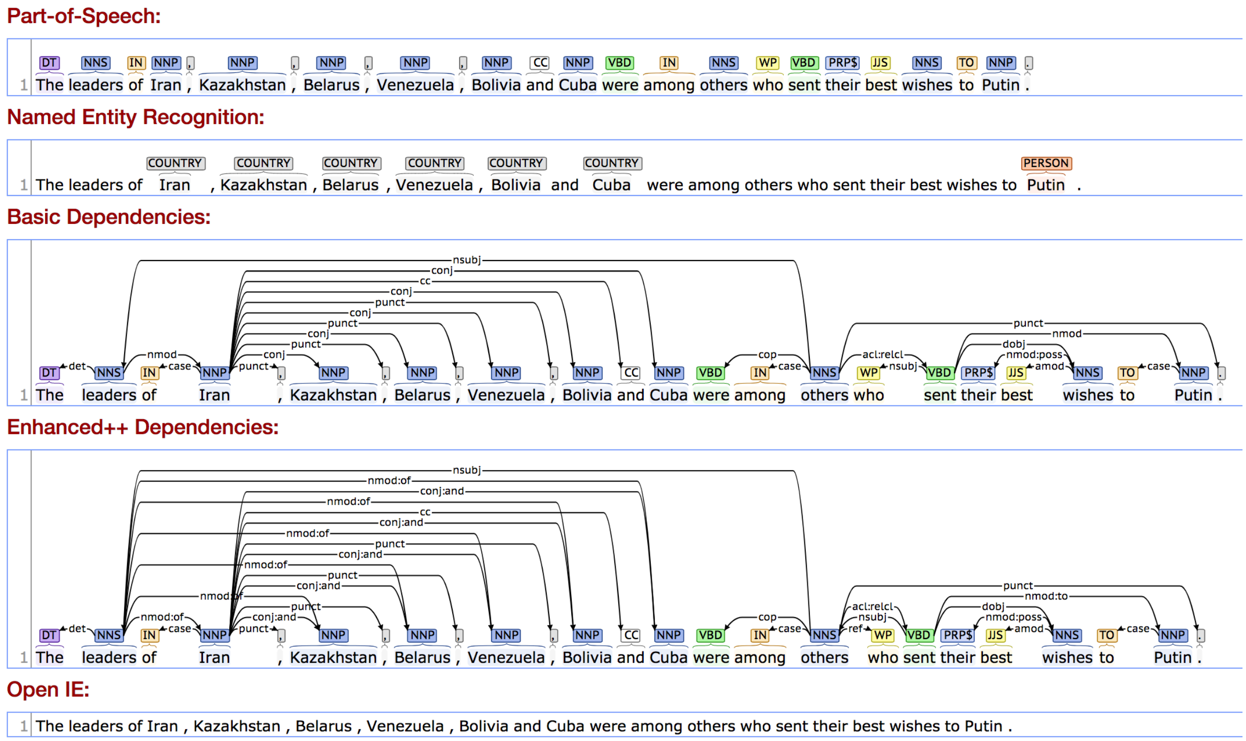
1) A broad range of grammatical text analysis of major (human) languages

2) An annotation for arbitrary texts
3) Running a simple web service

Task 2

Initial sentence:

“The leaders of Iran, Kazakhstan, Belarus, Venezuela, Bolivia and Cuba were among others who sent their best wishes to Putin.”



Part-of-Speech:

1 The leaders of Iran , Kazakhstan , Belarus , Venezuela , Bolivia and Cuba were among others who sent their best wishes to Putin .

DT NNS IN NNP , NNP , NNP , NNP , NNP CC NNP VBD IN NNS WP VBD PRP$ JJS NNS TO NNP .

Named Entity Recognition:

1 The leaders of Iran , Kazakhstan , Belarus , Venezuela , Bolivia and Cuba were among others who sent their best wishes to Putin .

COUNTRY
COUNTRY
COUNTRY
COUNTRY
COUNTRY
COUNTRY
PERSON

Basic Dependencies:

1 The leaders of Iran , Kazakhstan , Belarus , Venezuela , Bolivia and Cuba were among others who sent their best wishes to Putin .

Enhanced++ Dependencies:

1 The leaders of Iran , Kazakhstan , Belarus , Venezuela , Bolivia and Cuba were among others who sent their best wishes to Putin .

Open IE:

1 The leaders of Iran , Kazakhstan , Belarus , Venezuela , Bolivia and Cuba were among others who sent their best wishes to Putin .

1) Part-of-Speech

The same idea with doing Context Free Grammar. It assigns parts of speech to each word (and other token), such as:

- **DT** - determinant
- **NNS** - a plural noun
- **IN** - preposition/sub-conjunction
- **NNP** - singular proper noun
- **CC** - coordinating conjunction
- **VBD** - verb in a past tense
- **WP** - wh-pronoun
- **PRP$** - possessive pronoun
- **JJS** - superlative adjective
- **TO** - separate preposition “to”

2) Named Entity Recognition

NER module recognised the words in a text which are a person (Putin) and countries (Iran, Kazakhstan, Belarus, Venezuela, Bolivia, Cuba).

3) Basic Dependencies

On top of the part-of-speach labeling, the dependencies of words to each other are marked and punctuation marks are also defined. As a result, module provides direct relations between content words.

- **det** - determinant regardint to the ise of "the" ate the beginning of the sentence
- **nmod** - nominal modifier, which links two connected nouns: "leaders" and "Iran"
- **case** - connects prepositions with the noun it refers to:

"of" with "Iran"
"among" with "others"
"to" with "Putin"

- **punc** - commas and full-stop were identified.
- **conj, cc** - labels conjunction and mentioning of the same words, which refer to the same context
- **nsubj** - subjects in the sentence, consisting of 2 small sentences like "leaders" with "others", connection between some kind of proverb "who" and verb "sent"
- **cop** - connection linkin the subject of a sentence with a predicate (a subject complement): "were" and "others" in our case
- **acl:relcl** - connection between the subject and controlled verb like "others" and "sent"
- **nmod** - link between two words, equals to the cases in some languages, meaning posession or reference
- **dobj** - direct object in a noun phrase which accusatives object of the verb: "sent" with "wishes"
- **amod** - adjectival modifier in an adjectival phrase that serves to modify the meaning of the noun phrase: "best" and "wishes"

4) Enhanced++ Dependencies

This type of analysis represents more detailed description, which includes mentioning each conjuncted modificator.

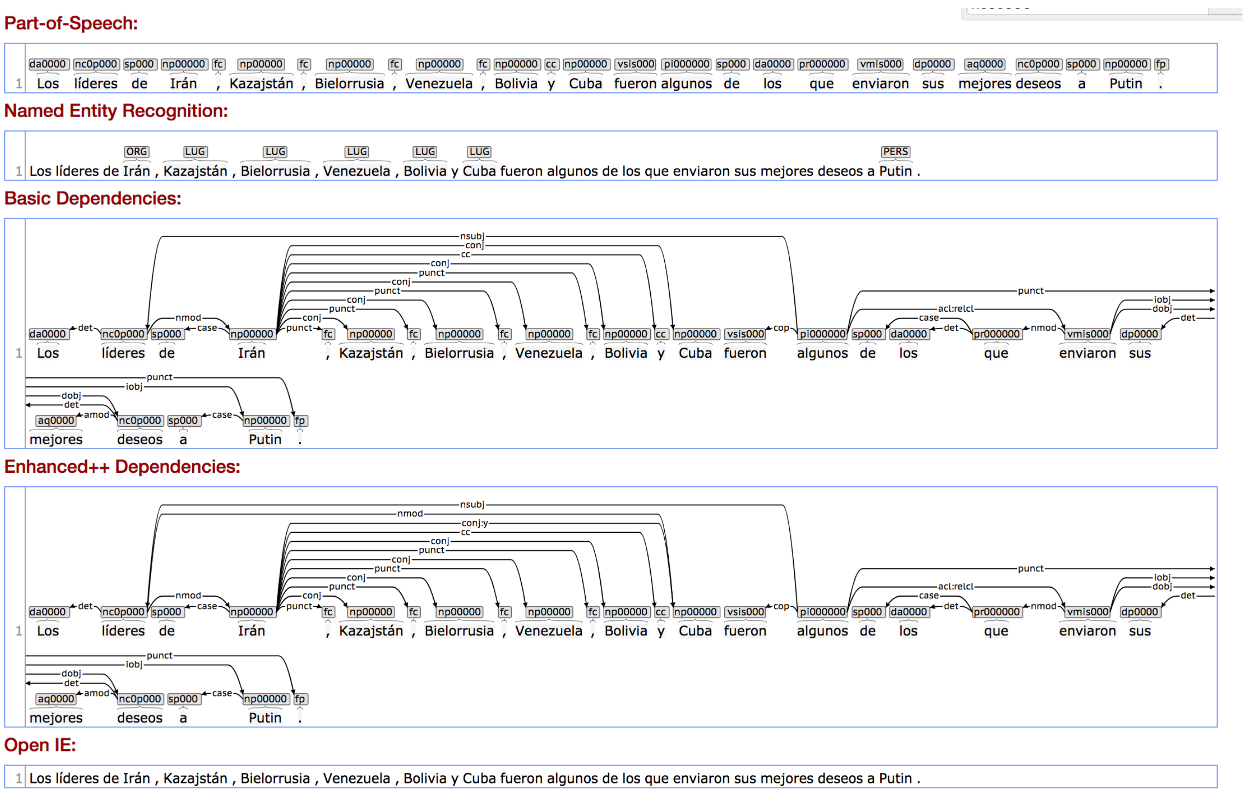
5) Open IE

Open information extraction, which should extract the relation tuples, typically binary relations, from plain text did not work at all for some reason. May be this happen because the sentence is too long and have non-standard reference or Stanford NLP prefers not to mess with Putin.

Task 3

Initial sentence:

“Los líderes de Irán, Kazajstán, Bielorrusia, Venezuela, Bolivia y Cuba fueron algunos de los que enviaron sus mejores deseos a Putin.”



Part-of-Speech:

	da0000	nc0p000	sp000	np00000	fc	np00000	fc	np00000	fc	np00000	fc	np00000	cc	np00000	vsis000	pi000000	sp000	da0000	pr000000	vmis000	dp0000	aq0000	nc0p000	sp000	np00000	fp
1	Los	líderes	de	Irán	,	Kazajstán	,	Bielorrusia	,	Venezuela	,	Bolivia	y	Cuba	fueron	algunos	de	los	que	enviaron	sus	mejores	deseos	a	Putin	.

Named Entity Recognition:

1 Los líderes de Irán , Kazajstán , Bielorrusia , Venezuela , Bolivia y Cuba fueron algunos de los que enviaron sus mejores deseos a Putin .

Irán	Kazajstán	Bielorrusia	Venezuela	Bolivia	Cuba	Putin
ORG	LUG	LUG	LUG	LUG	LUG	PERS

Basic Dependencies:

1 Los líderes de Irán , Kazajstán , Bielorrusia , Venezuela , Bolivia y Cuba fueron algunos de los que enviaron sus mejores deseos a Putin .

Enhanced++ Dependencies:

1 Los líderes de Irán , Kazajstán , Bielorrusia , Venezuela , Bolivia y Cuba fueron algunos de los que enviaron sus mejores deseos a Putin .

Open IE:

1 Los líderes de Irán , Kazajstán , Bielorrusia , Venezuela , Bolivia y Cuba fueron algunos de los que enviaron sus mejores deseos a Putin .

- **da0000** - article (definite)
- **sp000** - preposition
- **nc0p000** - common noun (plural)
- **np00000** - proper noun
- **fc** - comma
- **vsis000** - verb (semiauxiliary, indicative, preterite)
- **dp0000** - possessive
- **vmip000** - verb (main, indicative, present)
- **aq0000** - adjective (descriptive)
- **fp** - period/full-stop

Open information extraction, which should extract the relation tuples, typically binary relations, from plain text did not work even here. May be this happen because the sentence is too long and have non-standard reference or Stanford NLP prefers not to mess with Putin.

Task 6

```
NLP> Adjectives in English absolutely have to be in this order: opinion-size-age-shape-colour-
origin-material-purpose Noun. So you can have a lovely little old rectangular green French silver
whittling knife.

Sentence #1 (14 tokens):
Adjectives in English absolutely have to be in this order: opinion-size-age-shape-colour-origin-
material-purpose Noun.

Tokens:
[Text=Adjectives CharacterOffsetBegin=0 CharacterOffsetEnd=10 PartOfSpeech=NNS
Lemma=adjective NamedEntityTag=O]
[Text=in CharacterOffsetBegin=11 CharacterOffsetEnd=13 PartOfSpeech=IN Lemma=in
NamedEntityTag=O]
[Text=English CharacterOffsetBegin=14 CharacterOffsetEnd=21 PartOfSpeech=NNP
Lemma=English NamedEntityTag=NATIONALITY]
[Text=absolutely CharacterOffsetBegin=22 CharacterOffsetEnd=32 PartOfSpeech=RB
Lemma=absolutely NamedEntityTag=O]
[Text=have CharacterOffsetBegin=33 CharacterOffsetEnd=37 PartOfSpeech=VBP Lemma=have
NamedEntityTag=O]
[Text=to CharacterOffsetBegin=38 CharacterOffsetEnd=40 PartOfSpeech=TO Lemma=to
NamedEntityTag=O]
[Text=be CharacterOffsetBegin=41 CharacterOffsetEnd=43 PartOfSpeech=VB Lemma=be
NamedEntityTag=O]
[Text=in CharacterOffsetBegin=44 CharacterOffsetEnd=46 PartOfSpeech=IN Lemma=in
NamedEntityTag=O]
[Text=this CharacterOffsetBegin=47 CharacterOffsetEnd=51 PartOfSpeech=DT Lemma=this
NamedEntityTag=O]
[Text=order CharacterOffsetBegin=52 CharacterOffsetEnd=57 PartOfSpeech=NN Lemma=order
NamedEntityTag=O]
[Text=: CharacterOffsetBegin=57 CharacterOffsetEnd=58 PartOfSpeech=: Lemma=:
NamedEntityTag=O]
[Text=opinion-size-age-shape-colour-origin-material-purpose CharacterOffsetBegin=59
CharacterOffsetEnd=112 PartOfSpeech=NN Lemma=opinion-size-age-shape-colour-origin-
material-purpose NamedEntityTag=O]
[Text=Noun CharacterOffsetBegin=113 CharacterOffsetEnd=117 PartOfSpeech=NN
Lemma=noun NamedEntityTag=O]
[Text=. CharacterOffsetBegin=117 CharacterOffsetEnd=118 PartOfSpeech=. Lemma=.
NamedEntityTag=O]

Constituency parse:
(ROOT
  (S
    (NP
      (NP (NNS Adjectives))
      (PP (IN in)
        (NP (NNP English))))
    (ADVP (RB absolutely))
    (VP (VBP have)
      (S
        (VP (TO to)
          (VP (VB be)
            (PP (IN in)
              (NP
                (NP (DT this) (NN order))
                (: :)
                (NP (NN opinion-size-age-shape-colour-origin-material-purpose) (NN Noun))))))))
    (. .)))

Dependency Parse (enhanced plus dependencies):
root(ROOT-0, have-5)
nsubj(have-5, Adjectives-1)
nsubj:xsubj(order-10, Adjectives-1)
case(English-3, in-2)
nmod:in(Adjectives-1, English-3)
advmod(have-5, absolutely-4)
mark(order-10, to-6)
cop(order-10, be-7)
case(order-10, in-8)
det(order-10, this-9)
xcomp(have-5, order-10)
punct(order-10, :-11)
compound(Noun-13, opinion-size-age-shape-colour-origin-material-purpose-12)
dep(order-10, Noun-13)
punct(have-5, .-14)

Extracted the following NER entity mentions:
English	NATIONALITY

Sentence #2 (15 tokens):
So you can have a lovely little old rectangular green French silver whittling knife.

Tokens:
[Text=So CharacterOffsetBegin=119 CharacterOffsetEnd=121 PartOfSpeech=IN Lemma=so NamedEntityTag=O]
[Text=you CharacterOffsetBegin=122 CharacterOffsetEnd=125 PartOfSpeech=PRP Lemma=you NamedEntityTag=O]
```

```
[Text=can CharacterOffsetBegin=126 CharacterOffsetEnd=129 PartOfSpeech=MD Lemma=can NamedEntityTag=O]
[Text=have CharacterOffsetBegin=130 CharacterOffsetEnd=134 PartOfSpeech=VB Lemma=have NamedEntityTag=O]
[Text=a CharacterOffsetBegin=135 CharacterOffsetEnd=136 PartOfSpeech=DT Lemma=a NamedEntityTag=O]
[Text=lovely CharacterOffsetBegin=137 CharacterOffsetEnd=143 PartOfSpeech=JJ Lemma=lovely NamedEntityTag=O]
[Text=little CharacterOffsetBegin=144 CharacterOffsetEnd=150 PartOfSpeech=JJ Lemma=little NamedEntityTag=O]
[Text=old CharacterOffsetBegin=151 CharacterOffsetEnd=154 PartOfSpeech=JJ Lemma=old NamedEntityTag=O]
[Text=rectangular CharacterOffsetBegin=155 CharacterOffsetEnd=166 PartOfSpeech=JJ Lemma=rectangular NamedEntityTag=O]
[Text=green CharacterOffsetBegin=167 CharacterOffsetEnd=172 PartOfSpeech=JJ Lemma=green NamedEntityTag=O]
[Text=French CharacterOffsetBegin=173 CharacterOffsetEnd=179 PartOfSpeech=JJ Lemma=french NamedEntityTag=NATIONALITY]
[Text=silver CharacterOffsetBegin=180 CharacterOffsetEnd=186 PartOfSpeech=NN Lemma=silver NamedEntityTag=O]
[Text=whittling CharacterOffsetBegin=187 CharacterOffsetEnd=196 PartOfSpeech=VBG Lemma=whittle NamedEntityTag=O]
[Text=knife CharacterOffsetBegin=197 CharacterOffsetEnd=202 PartOfSpeech=NN Lemma=knife NamedEntityTag=O]
[Text=. CharacterOffsetBegin=202 CharacterOffsetEnd=203 PartOfSpeech=. Lemma=. NamedEntityTag=O]
```

Constituency parse:

```
(ROOT
  (S (IN So)
    (NP (PRP you))
    (VP (MD can)
      (VP (VB have)
        (S
          (NP (DT a) (JJ lovely) (JJ little) (JJ old) (JJ rectangular) (JJ green) (JJ French) (NN silver) (VBG whittling) (NN knife)))))
    (. .)))
```

Dependency Parse (enhanced plus plus dependencies):

```
root(ROOT-0, have-4)
dep(have-4, So-1)
nsubj(have-4, you-2)
aux(have-4, can-3)
det(knife-14, a-5)
```

```
amod(knife-14, lovely-6)
amod(knife-14, little-7)
amod(knife-14, old-8)
amod(knife-14, rectangular-9)
amod(knife-14, green-10)
amod(knife-14, French-11)
compound(knife-14, silver-12)
amod(knife-14, whittling-13)
xcomp(have-4, knife-14)
punct(have-4, .-15)

Extracted the following NER entity mentions:
French NATIONALITY
NLP>
```

Comparing to the website version, Stanford CORE NLP module produces the same result, only in a command prompt interface. For some reasons author's word "opinion-size-age-shape-colour-origin-material-purpose" was not split for smaller parts, probably, because grammar was trained using some double words (which considered to be one word). The word French was also recognised as nationality, but nationality is rarely associated with non-alive objects.

Task 6

How can you process a list of files?

To process a list of files there is a need to use the -filelist parameter: java -cp "*" -Xmx2g edu.stanford.nlp.pipeline.StanfordCoreNLP [-props myprops.props] -filelist filelist.txt where the -filelist parameter points to a file whose content lists all files to be processed (one per line).

What is a .properties file?

Properties file is a configuration file (a Java Properties file), which is recommended to be created manually before using Stanford CoreNLP. Minimally, this file should contain the "annotators" property, which contains a comma-separated list of Annotators to use. Also can be included:

- segment (for Chineese language),
- sentence split parameters and directories for a pos (Part Of Speach) model tagger,
- ner (Named Entity Recognition),
- regexner (pattern-based interface for doing Named Entity Recognition),
- parse (Dependencies Annotation),
- depparse (Direct Dependency Parser Annotator),
- coref (Pronominal and Nominal Coreference Resolution),
- kbp (Knowledge Base Population),
- entitylink (Entity Linking)

Task 7

Run the package from the command line:

```
NLP> Queen Mary University is located in London. It is a great university.

Sentence #1 (8 tokens):
Queen Mary University is located in London.

Tokens:
[Text=Queen CharacterOffsetBegin=0 CharacterOffsetEnd=5 PartOfSpeech=NNP
Lemma=Queen NamedEntityTag=ORGANIZATION]
[Text=Mary CharacterOffsetBegin=6 CharacterOffsetEnd=10 PartOfSpeech=NNP Lemma=Mary
NamedEntityTag=ORGANIZATION]
[Text=University CharacterOffsetBegin=11 CharacterOffsetEnd=21 PartOfSpeech=NNP
Lemma=University NamedEntityTag=ORGANIZATION]
[Text=is CharacterOffsetBegin=22 CharacterOffsetEnd=24 PartOfSpeech=VBZ Lemma=be
NamedEntityTag=O]
[Text=located CharacterOffsetBegin=25 CharacterOffsetEnd=32 PartOfSpeech=JJ
Lemma=located NamedEntityTag=O]
[Text=in CharacterOffsetBegin=33 CharacterOffsetEnd=35 PartOfSpeech=IN Lemma=in
NamedEntityTag=O]
[Text=London CharacterOffsetBegin=36 CharacterOffsetEnd=42 PartOfSpeech=NNP
Lemma=London NamedEntityTag=CITY]
[Text=. CharacterOffsetBegin=42 CharacterOffsetEnd=43 PartOfSpeech=. Lemma=.
NamedEntityTag=O]

Dependency Parse (enhanced plus plus dependencies):
root(ROOT-0, located-5)
compound(University-3, Queen-1)
compound(University-3, Mary-2)
nsubjpass(located-5, University-3)
auxpass(located-5, is-4)
case(London-7, in-6)
nmod:in(located-5, London-7)
punct(located-5, .-8)

Extracted the following NER entity mentions:
Queen Mary University       ORGANIZATION
London          CITY

Sentence #2 (6 tokens):
It is a great university.

Tokens:
```

```
[Text=It CharacterOffsetBegin=44 CharacterOffsetEnd=46 PartOfSpeech=PRP Lemma=it
NamedEntityTag=O]
[Text=is CharacterOffsetBegin=47 CharacterOffsetEnd=49 PartOfSpeech=VBZ Lemma=be
NamedEntityTag=O]
[Text=a CharacterOffsetBegin=50 CharacterOffsetEnd=51 PartOfSpeech=DT Lemma=a
NamedEntityTag=O]
[Text=great CharacterOffsetBegin=52 CharacterOffsetEnd=57 PartOfSpeech=JJ Lemma=great
NamedEntityTag=O]
[Text=university CharacterOffsetBegin=58 CharacterOffsetEnd=68 PartOfSpeech=NN
Lemma=university NamedEntityTag=O]
[Text=. CharacterOffsetBegin=68 CharacterOffsetEnd=69 PartOfSpeech=. Lemma=.
NamedEntityTag=O]

Dependency Parse (enhanced plus plus dependencies):
root(ROOT-0, university-5)
nsubj(university-5, It-1)
cop(university-5, is-2)
det(university-5, a-3)
amod(university-5, great-4)
punct(university-5, .-6)

Extracted the following NER entity mentions:

Coreference set:
        (2,1,[1,2]) -> (1,3,[1,4]), that is: "It" -> "Queen Mary University"
```

Run the package using the annotators to produce xml package:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
  <document>
    <docId>input.txt</docId>
    <sentences>
      <sentence id="1">
        <tokens>
          <token id="1">
            <word>Queen</word>
            <lemma>Queen</lemma>
            <CharacterOffsetBegin>0</CharacterOffsetBegin>
            <CharacterOffsetEnd>5</CharacterOffsetEnd>
            <POS>NNP</POS>
            <NER>ORGANIZATION</NER>
            <Speaker>PER0</Speaker>
```

```
</token>
<token id="2">
  <word>Mary</word>
  <lemma>Mary</lemma>
  <CharacterOffsetBegin>6</CharacterOffsetBegin>
  <CharacterOffsetEnd>10</CharacterOffsetEnd>
  <POS>NNP</POS>
  <NER>ORGANIZATION</NER>
  <Speaker>PER0</Speaker>
</token>
<token id="3">
  <word>University</word>
  <lemma>University</lemma>
  <CharacterOffsetBegin>11</CharacterOffsetBegin>
  <CharacterOffsetEnd>21</CharacterOffsetEnd>
  <POS>NNP</POS>
  <NER>ORGANIZATION</NER>
  <Speaker>PER0</Speaker>
</token>
<token id="4">
  <word>is</word>
  <lemma>be</lemma>
  <CharacterOffsetBegin>22</CharacterOffsetBegin>
  <CharacterOffsetEnd>24</CharacterOffsetEnd>
  <POS>VBZ</POS>
  <NER>O</NER>
  <Speaker>PER0</Speaker>
</token>
<token id="5">
  <word>located</word>
  <lemma>located</lemma>
  <CharacterOffsetBegin>25</CharacterOffsetBegin>
  <CharacterOffsetEnd>32</CharacterOffsetEnd>
  <POS>JJ</POS>
  <NER>O</NER>
  <Speaker>PER0</Speaker>
</token>
<token id="6">
  <word>in</word>
  <lemma>in</lemma>
  <CharacterOffsetBegin>33</CharacterOffsetBegin>
  <CharacterOffsetEnd>35</CharacterOffsetEnd>
  <POS>IN</POS>
  <NER>O</NER>
```

```
          <Speaker>PER0</Speaker>
        </token>
        <token id="7">
          <word>London</word>
          <lemma>London</lemma>
          <CharacterOffsetBegin>36</CharacterOffsetBegin>
          <CharacterOffsetEnd>42</CharacterOffsetEnd>
          <POS>NNP</POS>
          <NER>CITY</NER>
          <Speaker>PER0</Speaker>
        </token>
        <token id="8">
          <word>.</word>
          <lemma>.</lemma>
          <CharacterOffsetBegin>42</CharacterOffsetBegin>
          <CharacterOffsetEnd>43</CharacterOffsetEnd>
          <POS>.</POS>
          <NER>O</NER>
          <Speaker>PER0</Speaker>
        </token>
      </tokens>
      <parse>(ROOT (S (NP (NNP Queen) (NNP Mary) (NNP University)) (VP (VBZ is) (ADJP (JJ
located) (PP (IN in) (NP (NNP London))))) (. .))) </parse>
      <dependencies type="basic-dependencies">
        <dep type="root">
          <governor idx="0">ROOT</governor>
          <dependent idx="5">located</dependent>
        </dep>
        <dep type="compound">
          <governor idx="3">University</governor>
          <dependent idx="1">Queen</dependent>
        </dep>
        <dep type="compound">
          <governor idx="3">University</governor>
          <dependent idx="2">Mary</dependent>
        </dep>
        <dep type="nsubj">
          <governor idx="5">located</governor>
          <dependent idx="3">University</dependent>
        </dep>
        <dep type="cop">
          <governor idx="5">located</governor>
          <dependent idx="4">is</dependent>
        </dep>
      <dep type="case">
        <governor idx="7">London</governor>
        <dependent idx="6">in</dependent>
      </dep>
      <dep type="nmod">
        <governor idx="5">located</governor>
        <dependent idx="7">London</dependent>
      </dep>
      <dep type="punct">
        <governor idx="5">located</governor>
        <dependent idx="8">.</dependent>
      </dep>
    </dependencies>
    <dependencies type="collapsed-dependencies">
      <dep type="root">
        <governor idx="0">ROOT</governor>
        <dependent idx="5">located</dependent>
      </dep>
      <dep type="compound">
        <governor idx="3">University</governor>
        <dependent idx="1">Queen</dependent>
      </dep>
      <dep type="compound">
        <governor idx="3">University</governor>
        <dependent idx="2">Mary</dependent>
      </dep>
      <dep type="nsubj">
        <governor idx="5">located</governor>
        <dependent idx="3">University</dependent>
      </dep>
      <dep type="cop">
        <governor idx="5">located</governor>
        <dependent idx="4">is</dependent>
      </dep>
      <dep type="case">
        <governor idx="7">London</governor>
        <dependent idx="6">in</dependent>
      </dep>
      <dep type="nmod:in">
        <governor idx="5">located</governor>
        <dependent idx="7">London</dependent>
      </dep>
      <dep type="punct">
        <governor idx="5">located</governor>
    <dependent idx="8">.</dependent>
  </dep>
</dependencies>
<dependencies type="collapsed-ccprocessed-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="5">located</dependent>
  </dep>
  <dep type="compound">
    <governor idx="3">University</governor>
    <dependent idx="1">Queen</dependent>
  </dep>
  <dep type="compound">
    <governor idx="3">University</governor>
    <dependent idx="2">Mary</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="5">located</governor>
    <dependent idx="3">University</dependent>
  </dep>
  <dep type="cop">
    <governor idx="5">located</governor>
    <dependent idx="4">is</dependent>
  </dep>
  <dep type="case">
    <governor idx="7">London</governor>
    <dependent idx="6">in</dependent>
  </dep>
  <dep type="nmod:in">
    <governor idx="5">located</governor>
    <dependent idx="7">London</dependent>
  </dep>
  <dep type="punct">
    <governor idx="5">located</governor>
    <dependent idx="8">.</dependent>
  </dep>
</dependencies>
<dependencies type="enhanced-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="5">located</dependent>
  </dep>
  <dep type="compound">
    <governor idx="3">University</governor>
    <dependent idx="1">Queen</dependent>
  </dep>
  <dep type="compound">
    <governor idx="3">University</governor>
    <dependent idx="2">Mary</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="5">located</governor>
    <dependent idx="3">University</dependent>
  </dep>
  <dep type="cop">
    <governor idx="5">located</governor>
    <dependent idx="4">is</dependent>
  </dep>
  <dep type="case">
    <governor idx="7">London</governor>
    <dependent idx="6">in</dependent>
  </dep>
  <dep type="nmod:in">
    <governor idx="5">located</governor>
    <dependent idx="7">London</dependent>
  </dep>
  <dep type="punct">
    <governor idx="5">located</governor>
    <dependent idx="8">.</dependent>
  </dep>
</dependencies>
<dependencies type="enhanced-plus-plus-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="5">located</dependent>
  </dep>
  <dep type="compound">
    <governor idx="3">University</governor>
    <dependent idx="1">Queen</dependent>
  </dep>
  <dep type="compound">
    <governor idx="3">University</governor>
    <dependent idx="2">Mary</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="5">located</governor>
    <dependent idx="3">University</dependent>
  </dep>
```

```
        <dep type="cop">
          <governor idx="5">located</governor>
          <dependent idx="4">is</dependent>
        </dep>
        <dep type="case">
          <governor idx="7">London</governor>
          <dependent idx="6">in</dependent>
        </dep>
        <dep type="nmod:in">
          <governor idx="5">located</governor>
          <dependent idx="7">London</dependent>
        </dep>
        <dep type="punct">
          <governor idx="5">located</governor>
          <dependent idx="8">.</dependent>
        </dep>
      </dependencies>
    </sentence>
    <sentence id="2">
      <tokens>
        <token id="1">
          <word>It</word>
          <lemma>it</lemma>
          <CharacterOffsetBegin>44</CharacterOffsetBegin>
          <CharacterOffsetEnd>46</CharacterOffsetEnd>
          <POS>PRP</POS>
          <NER>O</NER>
          <Speaker>PER0</Speaker>
        </token>
        <token id="2">
          <word>is</word>
          <lemma>be</lemma>
          <CharacterOffsetBegin>47</CharacterOffsetBegin>
          <CharacterOffsetEnd>49</CharacterOffsetEnd>
          <POS>VBZ</POS>
          <NER>O</NER>
          <Speaker>PER0</Speaker>
        </token>
        <token id="3">
          <word>a</word>
          <lemma>a</lemma>
          <CharacterOffsetBegin>50</CharacterOffsetBegin>
          <CharacterOffsetEnd>51</CharacterOffsetEnd>
          <POS>DT</POS>
      <NER>O</NER>
      <Speaker>PER0</Speaker>
     </token>
     <token id="4">
      <word>great</word>
      <lemma>great</lemma>
      <CharacterOffsetBegin>52</CharacterOffsetBegin>
      <CharacterOffsetEnd>57</CharacterOffsetEnd>
      <POS>JJ</POS>
      <NER>O</NER>
      <Speaker>PER0</Speaker>
     </token>
     <token id="5">
      <word>university</word>
      <lemma>university</lemma>
      <CharacterOffsetBegin>58</CharacterOffsetBegin>
      <CharacterOffsetEnd>68</CharacterOffsetEnd>
      <POS>NN</POS>
      <NER>O</NER>
      <Speaker>PER0</Speaker>
     </token>
     <token id="6">
      <word>.</word>
      <lemma>.</lemma>
      <CharacterOffsetBegin>68</CharacterOffsetBegin>
      <CharacterOffsetEnd>69</CharacterOffsetEnd>
      <POS>.</POS>
      <NER>O</NER>
      <Speaker>PER0</Speaker>
     </token>
    </tokens>
    <parse>(ROOT (S (NP (PRP It)) (VP (VBZ is) (NP (DT a) (JJ great) (NN university))) (. .)))
</parse>
    <dependencies type="basic-dependencies">
     <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="5">university</dependent>
     </dep>
     <dep type="nsubj">
      <governor idx="5">university</governor>
      <dependent idx="1">It</dependent>
     </dep>
     <dep type="cop">
      <governor idx="5">university</governor>
    <dependent idx="2">is</dependent>
  </dep>
  <dep type="det">
    <governor idx="5">university</governor>
    <dependent idx="3">a</dependent>
  </dep>
  <dep type="amod">
    <governor idx="5">university</governor>
    <dependent idx="4">great</dependent>
  </dep>
  <dep type="punct">
    <governor idx="5">university</governor>
    <dependent idx="6">.</dependent>
  </dep>
</dependencies>
<dependencies type="collapsed-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="5">university</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="5">university</governor>
    <dependent idx="1">It</dependent>
  </dep>
  <dep type="cop">
    <governor idx="5">university</governor>
    <dependent idx="2">is</dependent>
  </dep>
  <dep type="det">
    <governor idx="5">university</governor>
    <dependent idx="3">a</dependent>
  </dep>
  <dep type="amod">
    <governor idx="5">university</governor>
    <dependent idx="4">great</dependent>
  </dep>
  <dep type="punct">
    <governor idx="5">university</governor>
    <dependent idx="6">.</dependent>
  </dep>
</dependencies>
<dependencies type="collapsed-ccprocessed-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="5">university</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="5">university</governor>
    <dependent idx="1">It</dependent>
  </dep>
  <dep type="cop">
    <governor idx="5">university</governor>
    <dependent idx="2">is</dependent>
  </dep>
  <dep type="det">
    <governor idx="5">university</governor>
    <dependent idx="3">a</dependent>
  </dep>
  <dep type="amod">
    <governor idx="5">university</governor>
    <dependent idx="4">great</dependent>
  </dep>
  <dep type="punct">
    <governor idx="5">university</governor>
    <dependent idx="6">.</dependent>
  </dep>
</dependencies>
<dependencies type="enhanced-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="5">university</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="5">university</governor>
    <dependent idx="1">It</dependent>
  </dep>
  <dep type="cop">
    <governor idx="5">university</governor>
    <dependent idx="2">is</dependent>
  </dep>
  <dep type="det">
    <governor idx="5">university</governor>
    <dependent idx="3">a</dependent>
  </dep>
  <dep type="amod">
    <governor idx="5">university</governor>
    <dependent idx="4">great</dependent>
  </dep>
```

```
      <dep type="punct">
        <governor idx="5">university</governor>
        <dependent idx="6">.</dependent>
      </dep>
    </dependencies>
    <dependencies type="enhanced-plus-plus-dependencies">
      <dep type="root">
        <governor idx="0">ROOT</governor>
        <dependent idx="5">university</dependent>
      </dep>
      <dep type="nsubj">
        <governor idx="5">university</governor>
        <dependent idx="1">It</dependent>
      </dep>
      <dep type="cop">
        <governor idx="5">university</governor>
        <dependent idx="2">is</dependent>
      </dep>
      <dep type="det">
        <governor idx="5">university</governor>
        <dependent idx="3">a</dependent>
      </dep>
      <dep type="amod">
        <governor idx="5">university</governor>
        <dependent idx="4">great</dependent>
      </dep>
      <dep type="punct">
        <governor idx="5">university</governor>
        <dependent idx="6">.</dependent>
      </dep>
    </dependencies>
  </sentence>
</sentences>
<coreference>
  <coreference>
    <mention representative="true">
      <sentence>1</sentence>
      <start>1</start>
      <end>4</end>
      <head>3</head>
      <text>Queen Mary University</text>
    </mention>
    <mention>
      <sentence>2</sentence>
```

```
          <start>1</start>
          <end>2</end>
          <head>1</head>
          <text>It</text>
        </mention>
        <mention>
          <sentence>2</sentence>
          <start>3</start>
          <end>6</end>
          <head>5</head>
          <text>a great university</text>
        </mention>
      </coreference>
    </coreference>
  </document>
</root>
```

All information is represented as elements, where attributes characterise metadata. Dependency tree is written as one string, despite XML and parse trees both have embedded structure.

Task 8

Run the package with the CONLL output format:

```
java -cp "*" -Xmx8g edu.stanford.nlp.pipeline.StanfordCoreNLP -annotators
tokenize,ssplit,pos,lemma,ner,parse,dcoref -file input.txt -outputFormat conll
```

```
1         Queen     Queen     NNP       ORGANIZATION                3         compound
2         Mary      Mary      NNP       ORGANIZATION                3         compound
3         University          University          NNP       ORGANIZATION                5         nsubj
4         is        be        VBZ       O         5         cop
5         located             located             JJ        O         0         ROOT
6         in        in        IN        O         7         case
7         London              London              NNP       CITY      5         nmod
8         .         .         .         O         5         punct

1         It        it        PRP       O         5         nsubj
2         is        be        VBZ       O         5         cop
3         a         a         DT        O         5         det
4         great     great     JJ        O         5         amod
5         university          university          NN        O         0         ROOT
6         .         .         .         O         5         punct
```

Task 9

Run the package with the CONLL output format from the properties file:

# Content of the properties file:

```
annotators = tokenize, ssplit, pos, lemma, ner, parse, dcoref
outputFormat = conll
file = input.txt
```

# Run from config file:

```
java -cp "*" -Xmx8g edu.stanford.nlp.pipeline.StanfordCoreNLP -file input.txt -props StanfordCoreNLP.properties
```

Task 10

What does it mean to say that the deterministic coreference resolver is based on a multi-pass sieve?

Multi-pass sieve coreference resolution stands for an anaphora resolution, which applies tiers of deterministic coreference models (”sieves”) one at a time from highest to lowest precision, where each model builds on the previous model’s cluster output. The aim is to achieve both high precision and high recall.

Task 11

Interesting examples of coreference that the system can process correctly:

1) “If practice makes perfect and perfect needs practice, I'm perfectly practiced and practically perfect.” Does not mix adjective and noun with the same spelling.
2) “I wish to wish the wish you wish to wish, but if you wish the wish the witch wishes, I won't wish the wish you wish to wish.” Correctly identified noun and verb having the same spelling.

Interesting examples of coreference that the system can not process correctly:

1) “Lana was robbed by Moana and she was arrested by policemen.” Both verbs refer to Lana, despite Moana was arrested, according to the common sence.
2) “A ship is female and it takes a lot of paint to keep her good-looking.” No reference between the “ship” and “she”.

Task 12

Downloaded directory with all reference resolvers from website:
http://grepcode.com/snapshot/repo1.maven.org/maven2/edu.stanford.nlp/stanford-corenlp/3.2.0/

New configuration file:

```
annotators = tokenize, ssplit, pos, lemma, ner, parse, dcoref
outputFormat = conll
file = input.txt
dcoref.animate = /dcoref/animate.unigrams.txt
dcoref.inanimate = /dcoref/inanimate.unigrams.txt
dcoref.male = /dcoref/male.unigrams.txt
dcoref.neutral = /dcoref/neutral.unigrams.txt
dcoref.female = /dcoref/female.unigrams.txt
```

NLP> Lana was robbed by Moana and she was arrested by policemen.

```
Sentence #1 (12 tokens):
Lana was robbed by Moana and she was arrested by policemen.

Tokens:
[Text=Lana CharacterOffsetBegin=0 CharacterOffsetEnd=4 PartOfSpeech=NNP Lemma=Lana
NamedEntityTag=PERSON]
[Text=was CharacterOffsetBegin=5 CharacterOffsetEnd=8 PartOfSpeech=VBD Lemma=be
NamedEntityTag=O]
[Text=robbed CharacterOffsetBegin=9 CharacterOffsetEnd=15 PartOfSpeech=VBN Lemma=rob
NamedEntityTag=O]
[Text=by CharacterOffsetBegin=16 CharacterOffsetEnd=18 PartOfSpeech=IN Lemma=by
NamedEntityTag=O]
[Text=Moana CharacterOffsetBegin=19 CharacterOffsetEnd=24 PartOfSpeech=NNP
Lemma=Moana NamedEntityTag=PERSON]
[Text=and CharacterOffsetBegin=25 CharacterOffsetEnd=28 PartOfSpeech=CC Lemma=and
NamedEntityTag=O]
[Text=she CharacterOffsetBegin=29 CharacterOffsetEnd=32 PartOfSpeech=PRP Lemma=she
NamedEntityTag=O]
[Text=was CharacterOffsetBegin=33 CharacterOffsetEnd=36 PartOfSpeech=VBD Lemma=be
NamedEntityTag=O]
[Text=arrested CharacterOffsetBegin=37 CharacterOffsetEnd=45 PartOfSpeech=VBN
Lemma=arrest NamedEntityTag=O]
[Text=by CharacterOffsetBegin=46 CharacterOffsetEnd=48 PartOfSpeech=IN Lemma=by
NamedEntityTag=O]
[Text=policemen CharacterOffsetBegin=49 CharacterOffsetEnd=58 PartOfSpeech=NNS
Lemma=policeman NamedEntityTag=O]
[Text=. CharacterOffsetBegin=58 CharacterOffsetEnd=59 PartOfSpeech=. Lemma=.
NamedEntityTag=O]

Dependency Parse (enhanced plus plus dependencies):
root(ROOT-0, robbed-3)
nsubjpass(robbed-3, Lana-1)
```

```
auxpass(robbed-3, was-2)
case(Moana-5, by-4)
nmod:agent(robbed-3, Moana-5)
cc(robbed-3, and-6)
nsubjpass(arrested-9, she-7)
auxpass(arrested-9, was-8)
conj:and(robbed-3, arrested-9)
case(policemen-11, by-10)
nmod:agent(arrested-9, policemen-11)
punct(robbed-3, .-12)

Extracted the following NER entity mentions:
Lana   PERSON
Moana  PERSON
she    PERSON

Coreference set:
        (1,7,[7,8]) -> (1,1,[1,2]), that is: "she" -> "Lana"

NLP> A ship is female and it takes a lot of paint to keep her good-looking.
Sentence #1 (16 tokens):
A ship is female and it takes a lot of paint to keep her good-looking.

Tokens:
[Text=A CharacterOffsetBegin=0 CharacterOffsetEnd=1 PartOfSpeech=DT Lemma=a
NamedEntityTag=O]
[Text=ship CharacterOffsetBegin=2 CharacterOffsetEnd=6 PartOfSpeech=NN Lemma=ship
NamedEntityTag=O]
[Text=is CharacterOffsetBegin=7 CharacterOffsetEnd=9 PartOfSpeech=VBZ Lemma=be
NamedEntityTag=O]
[Text=female CharacterOffsetBegin=10 CharacterOffsetEnd=16 PartOfSpeech=JJ
Lemma=female NamedEntityTag=O]
[Text=and CharacterOffsetBegin=17 CharacterOffsetEnd=20 PartOfSpeech=CC Lemma=and
NamedEntityTag=O]
[Text=it CharacterOffsetBegin=21 CharacterOffsetEnd=23 PartOfSpeech=PRP Lemma=it
NamedEntityTag=O]
[Text=takes CharacterOffsetBegin=24 CharacterOffsetEnd=29 PartOfSpeech=VBZ Lemma=take
NamedEntityTag=O]
[Text=a CharacterOffsetBegin=30 CharacterOffsetEnd=31 PartOfSpeech=DT Lemma=a
NamedEntityTag=O]
[Text=lot CharacterOffsetBegin=32 CharacterOffsetEnd=35 PartOfSpeech=NN Lemma=lot
NamedEntityTag=O]
[Text=of CharacterOffsetBegin=36 CharacterOffsetEnd=38 PartOfSpeech=IN Lemma=of
NamedEntityTag=O]
```

```
[Text=paint CharacterOffsetBegin=39 CharacterOffsetEnd=44 PartOfSpeech=NN Lemma=paint
NamedEntityTag=O]
[Text=to CharacterOffsetBegin=45 CharacterOffsetEnd=47 PartOfSpeech=TO Lemma=to
NamedEntityTag=O]
[Text=keep CharacterOffsetBegin=48 CharacterOffsetEnd=52 PartOfSpeech=VB Lemma=keep
NamedEntityTag=O]
[Text=her CharacterOffsetBegin=53 CharacterOffsetEnd=56 PartOfSpeech=PRP$ Lemma=she
NamedEntityTag=O]
[Text=good-looking CharacterOffsetBegin=57 CharacterOffsetEnd=69 PartOfSpeech=JJ
Lemma=good-looking NamedEntityTag=O]
[Text=. CharacterOffsetBegin=69 CharacterOffsetEnd=70 PartOfSpeech=. Lemma=.
NamedEntityTag=O]

Dependency Parse (enhanced plus dependencies):
root(ROOT-0, female-4)
det(ship-2, A-1)
nsubj(female-4, ship-2)
cop(female-4, is-3)
cc(female-4, and-5)
nsubj(takes-7, it-6)
conj:and(female-4, takes-7)
det:qmod(paint-11, a-8)
mwe(a-8, lot-9)
mwe(a-8, of-10)
dobj(takes-7, paint-11)
mark(keep-13, to-12)
advcl:to(takes-7, keep-13)
nsubj(good-looking-15, her-14)
xcomp(keep-13, good-looking-15)
punct(female-4, .-16)

Extracted the following NER entity mentions:
her     PERSON

Coreference set:
        (1,6,[6,7]) -> (1,2,[1,3]), that is: "it" -> "A ship"
```

Result: Nothing changed. There is still no link between proper words because the knowledge of the subject area is needed.