Natural Language Processing and Information Extraction for Biomedicine

Guergana Savova, Ph.D.
Assistant Professor, CHIP and HMS

Jiaping Zheng, MS
CHIP and HMS

Overview

- Background
- cTAKES: overview
- cTAKES: type system
- cTAKES: coding example

Definitions

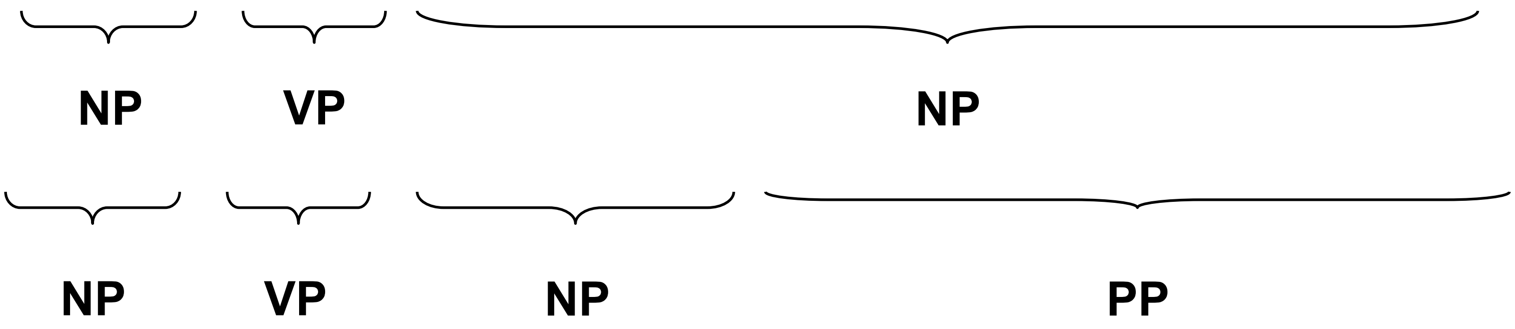
- **Information Extraction (IE)**
  - Extracting existing facts from unstructured or loosely structured text into a structured form
- **Information Retrieval (IR)**
  - Finding documents relevant to a user query
- **Named Entity Recognition (NER)**
  - Discovery of groups of textual mentions that belong to certain semantic class
- **Natural Language Processing (NLP)**
  - Computational methods for text processing based on linguistically sound principles
  - Clinical NLP – NLP for the clinical narrative
  - Biomedical NLP – NLP for the clinical narrative and biomedical literature

Problem Space

- **Structured information**
  - Relational databases
  - Easy to extract information from them
- **Semi-structured information**
  - Loosely formatted XML, CSV tables
  - Not challenging to extract information
- **Unstructured information**
  - Scholarly literature, clinical notes, research reports, webpages
  - Majority of information is unstructured!!
  - Real challenge to extract the information

Natural Language Processing

I	saw	the	man	with	the	telescope.
w1	w2	w3	w4	w5	w6	w7
pronoun	verb	article	noun	prep	article	noun



NP VP NP

NP VP NP PP

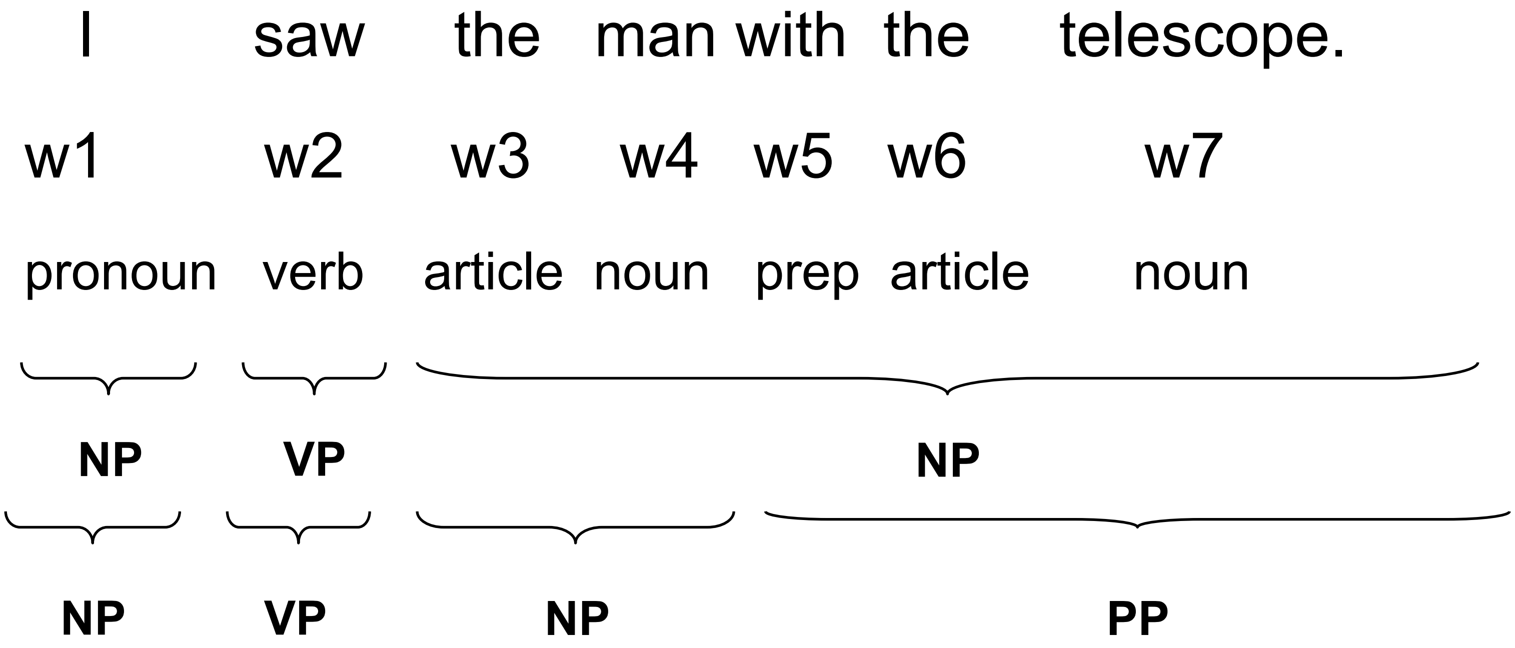
Courtesy Wendy Chapman

Natural Language Processing: Methods

Rule-based

Machine-learning/statistical

Hybrid



I saw the man with the telescope.

w1 w2 w3 w4 w5 w6 w7

pronoun verb article noun prep article noun

NP **VP** **NP**

NP **VP** **NP** **PP**

Why NLP? Why not Google?

- **From Google to language understanding**
  - Negation (and any other similar context)

    The patient denies headache, earache, sore throat, fever, rash, hallucinations, stomachache, cough and any pneumonia-related symptoms
  - Inverted syntax

    Colon, ascending and descending, biopsy
  - Relation discovery

    Tamoxifen is used in the treatment of breast cancer.
  - Morphologic variations

    runs, running, ran, run -> mapped to the same base form
  - Higher level discourse phenomena: synonyms, anaphora relations, temporal relations, document summarization

clinical Text Analysis and Knowledge Extraction System (cTAKES)

Overview

- **cTAKES**
  - Release 1.0 developed at Mayo (Savova and team)
  - Goal:
    - Phenotype extraction
    - Generic – to be used for a variety of retrievals and use cases
    - Expandable – at the information model level and methods
    - Modular
    - Cutting edge technologies – best methods combining existing practices and novel research with rapid technology transfer
    - Best software practices (80M+ notes)
- **Commitment to both R and D in R&D**

cTAKES Technical Details

- **Open source**
  - www.ohnlp.org
  - Downloads: Documentation and Downloads
  - Technical details: Publications
  - Java 1.5, Apache 2.0 license
- **Framework**
  - IBM's Unstructured Information Management Architecture (UIMA) open source framework, Apache project
- **Methods**
  - Natural Language Processing methods (NLP)
- **Application**
  - High-throughput system (80M+ notes; 80B+ tokens)

cTAKES: Components

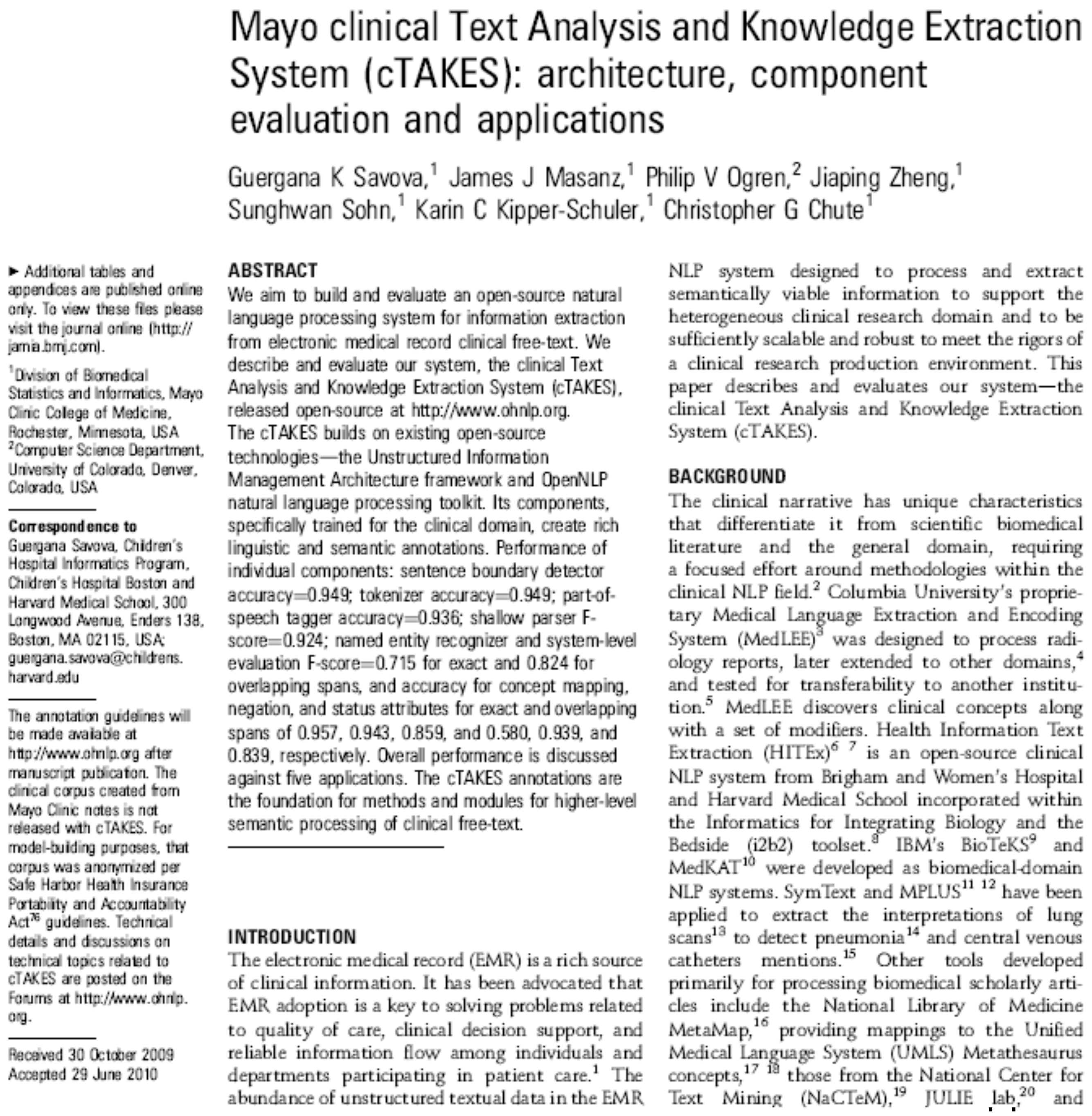
- **Clinical narrative as a sublanguage**
- **Core components**
  - Sentence boundary detection (OpenNLP technology)
  - Tokenization (rule-based)
  - Morphologic normalization (NLM’s LVG)
  - POS tagging (OpenNLP technology)
  - Shallow parsing (OpenNLP technology)
  - Named Entity Recognition
    - Dictionary mapping (lookup algorithm)
    - Machine learning (MAWUI)
    - types: diseases/disorders, signs/symptoms, anatomical sites, procedures, medications
  - Negation and context identification (NegEx)

Output Example: Drug Object

- **“Tamoxifen 20 mg po daily started on March 1, 2005**.”
  - **Drug**
    - Text: Tamoxifen
    - Associated code: C0351245
    - Strength: 20 mg
    - Start date: March 1, 2005
    - End date: null
    - Dosage: 1.0
    - Frequency: 1.0
    - Frequency unit: daily
    - Duration: null
    - Route: Enteral Oral
    - Form: null
    - Status: current
    - Change Status: no change
    - Certainty: null

Output Example: Disorder Object

- **“No evidence of cholangiocarcinoma.”**
  - **Disorder**
    - Text: cholangiocarcinoma
    - Associated code: SNOMED 70179006
    - Certainty: 1
    - Context: current
    - Relatedness to patient: true
    - Status: negated



Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications

Guergana K Savova,[1] James J Masanz,[1] Philip V Ogren,[2] Jiaping Zheng,[1] Sunghwan Sohn,[1] Karin C Kipper-Schuler,[1] Christopher G Chute[1]

► Additional tables and appendices are published online only. To view these files please visit the journal online (http://jamia.bmj.com).

[1]Division of Biomedical Statistics and Informatics, Mayo Clinic College of Medicine, Rochester, Minnesota, USA
[2]Computer Science Department, University of Colorado, Denver, Colorado, USA

Correspondence to
Guergana Savova, Children's Hospital Informatics Program, Children's Hospital Boston and Harvard Medical School, 300 Longwood Avenue, Enders 138, Boston, MA 02115, USA; guergana.savova@childrens.harvard.edu

The annotation guidelines will be made available at http://www.ohnlp.org after manuscript publication. The clinical corpus created from Mayo Clinic notes is not released with cTAKES. For model-building purposes, that corpus was anonymized per Safe Harbor Health Insurance Portability and Accountability Act[76] guidelines. Technical details and discussions on technical topics related to cTAKES are posted on the Forums at http://www.ohnlp.org.

Received 30 October 2009
Accepted 29 June 2010

ABSTRACT

We aim to build and evaluate an open-source natural language processing system for information extraction from electronic medical record clinical free-text. We describe and evaluate our system, the clinical Text Analysis and Knowledge Extraction System (cTAKES), released open-source at http://www.ohnlp.org. The cTAKES builds on existing open-source technologies—the Unstructured Information Management Architecture framework and OpenNLP natural language processing toolkit. Its components, specifically trained for the clinical domain, create rich linguistic and semantic annotations. Performance of individual components: sentence boundary detector accuracy=0.949; tokenizer accuracy=0.949; part-of-speech tagger accuracy=0.936; shallow parser F-score=0.924; named entity recognizer and system-level evaluation F-score=0.715 for exact and 0.824 for overlapping spans, and accuracy for concept mapping, negation, and status attributes for exact and overlapping spans of 0.957, 0.943, 0.859, and 0.580, 0.939, and 0.839, respectively. Overall performance is discussed against five applications. The cTAKES annotations are the foundation for methods and modules for higher-level semantic processing of clinical free-text.

INTRODUCTION

The electronic medical record (EMR) is a rich source of clinical information. It has been advocated that EMR adoption is a key to solving problems related to quality of care, clinical decision support, and reliable information flow among individuals and departments participating in patient care.[1] The abundance of unstructured textual data in the EMR NLP system designed to process and extract semantically viable information to support the heterogeneous clinical research domain and to be sufficiently scalable and robust to meet the rigors of a clinical research production environment. This paper describes and evaluates our system—the clinical Text Analysis and Knowledge Extraction System (cTAKES).

BACKGROUND

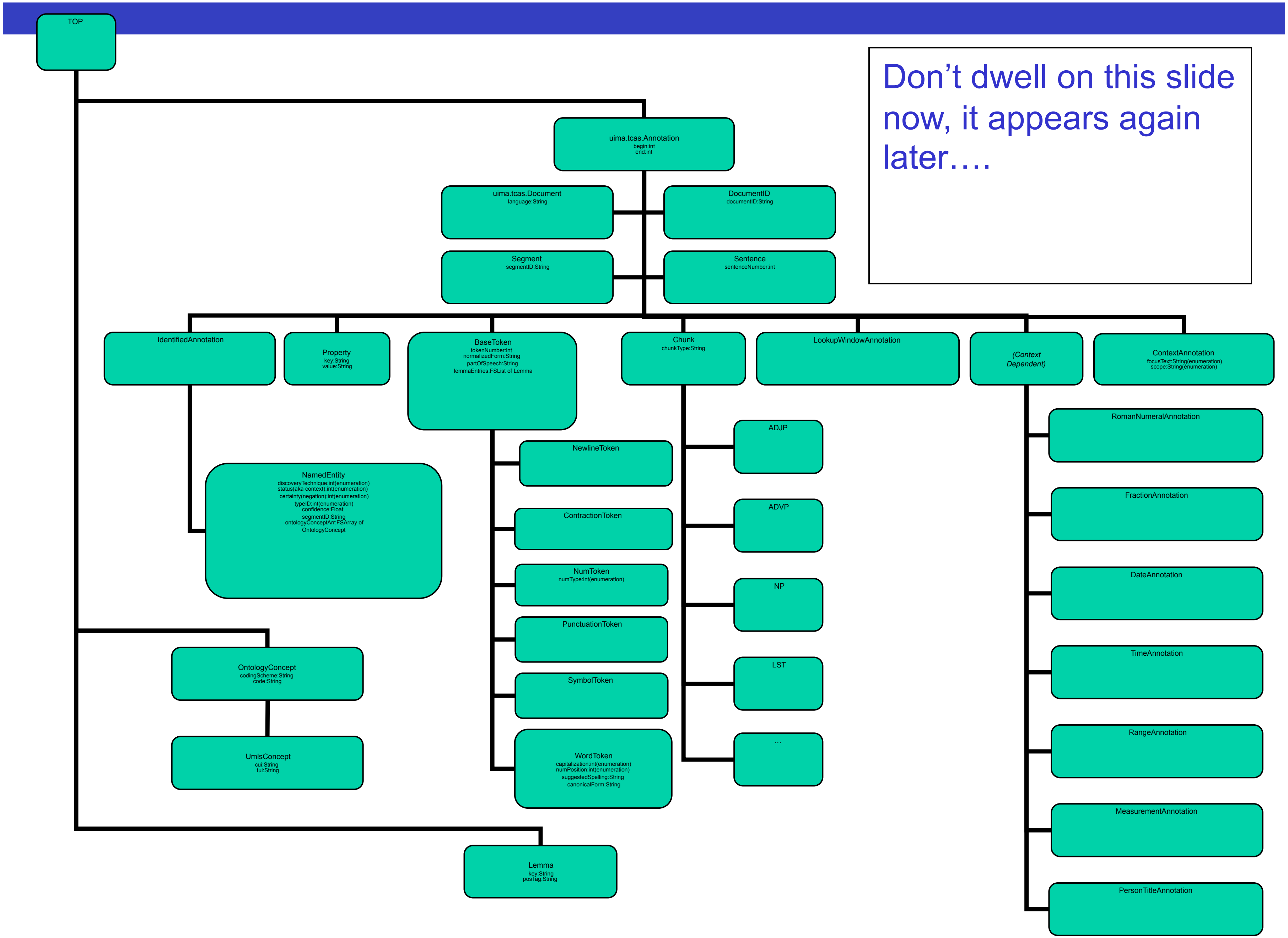
The clinical narrative has unique characteristics that differentiate it from scientific biomedical literature and the general domain, requiring a focused effort around methodologies within the clinical NLP field.[2] Columbia University's proprietary Medical Language Extraction and Encoding System (MedLEE)[3] was designed to process radiology reports, later extended to other domains,[4] and tested for transferability to another institution.[5] MedLEE discovers clinical concepts along with a set of modifiers. Health Information Text Extraction (HITEx)[6 7] is an open-source clinical NLP system from Brigham and Women's Hospital and Harvard Medical School incorporated within the Informatics for Integrating Biology and the Bedside (i2b2) toolset.[8] IBM's BioTeKS[9] and MedKAT[10] were developed as biomedical-domain NLP systems. SymText and MPLUS[11 12] have been applied to extract the interpretations of lung scans[13] to detect pneumonia[14] and central venous catheters mentions.[15] Other tools developed primarily for processing biomedical scholarly articles include the National Library of Medicine MetaMap,[16] providing mappings to the Unified Medical Language System (UMLS) Metathesaurus concepts,[17 18] those from the National Center for Text Mining (NaCTeM),[19] JULIE lab,[20] and

Mayo cTAKES: UIMA Type System

cTAKES 1.0.5

document version 1.0.0.1

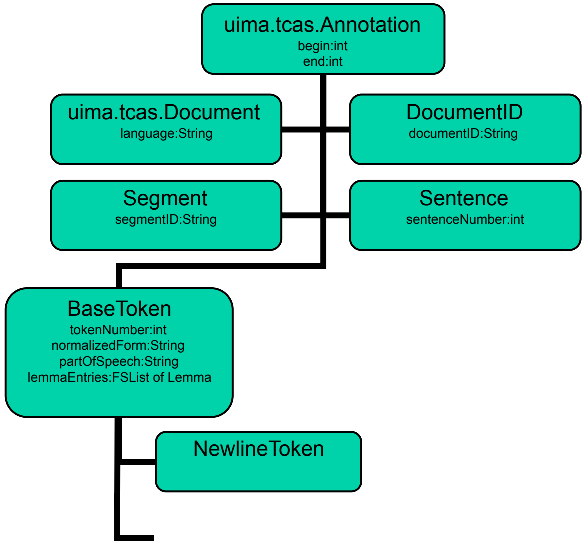
We start with a chart that shows the *inheritance* diagram of the UIMA Type System for cTAKES



Don’t dwell on this slide now, it appears again later….

Representing inheritance

Here's part of the larger diagram, to illustrate how inheritance is represented.

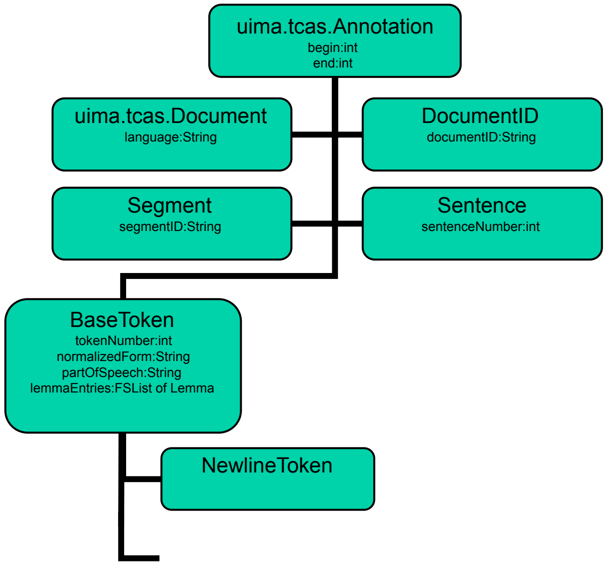


A parent has a line connecting to the bottom of its box. A child appears somewhere below its parent and is connected to its parent.

Document, DocumentID, Segment, Sentence and BaseToken are all children of uima.tcas.Annotation

NewlineToken is one of the children of BaseToken

Attributes accumulate:



uima.tcas.Annotation
begin:int
end:int

uima.tcas.Document
language:String

DocumentID
documentID:String

Segment
segmentID:String

Sentence
sentenceNumber:int

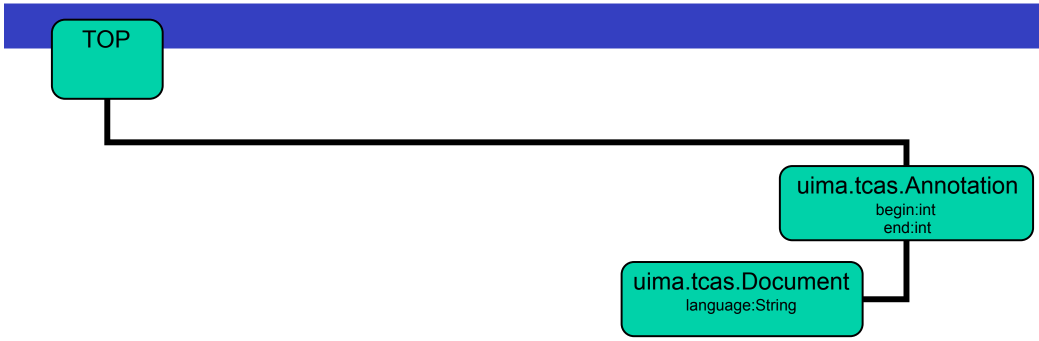
BaseToken
tokenNumber:int
normalizedForm:String
partOfSpeech:String
lemmaEntries:FSList of Lemma

NewlineToken

Children inherit their parent’s attributes, the attributes are not listed explicitly within the descendents

- All descendents of Annotation have begin and end attributes
- NewlineToken has the same attributes as BaseToken (which includes begin and end)

We will build up to the full diagram of the Type System for cTAKES, by starting with UIMA-provided types.



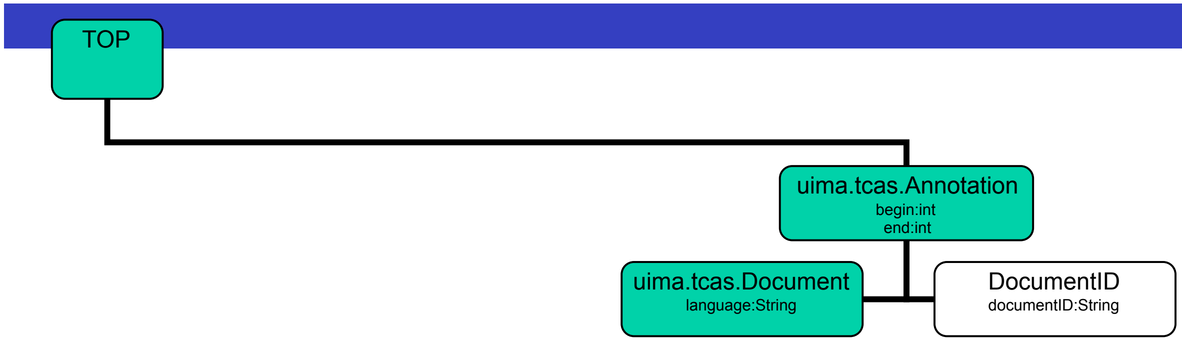
TOP

uima.tcas.Annotation
begin:int
end:int

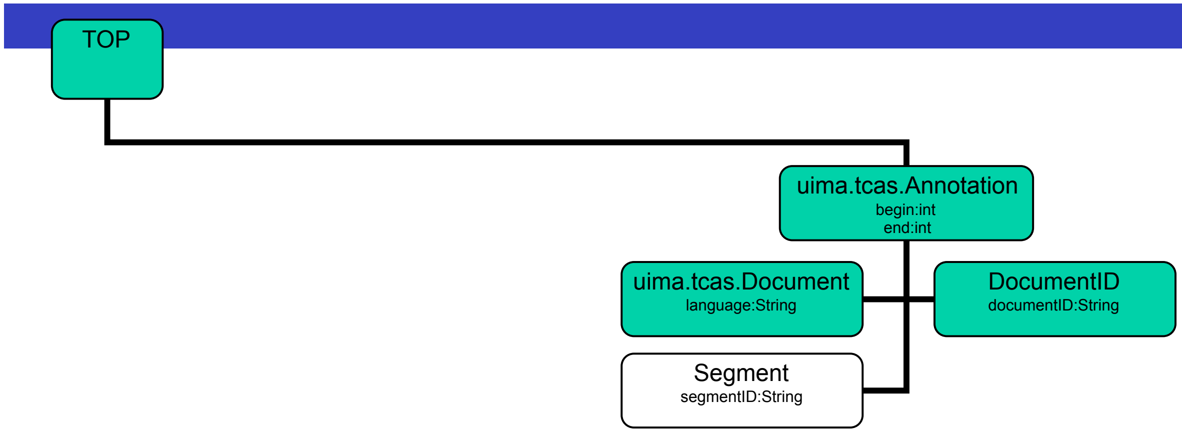
uima.tcas.Document
language:String

These types are provided by the UIMA framework.

Now we build up the cTAKES Type System diagram one step at a time, adding in the types in the order that annotations are generated by the Mayo cTAKES pipeline.

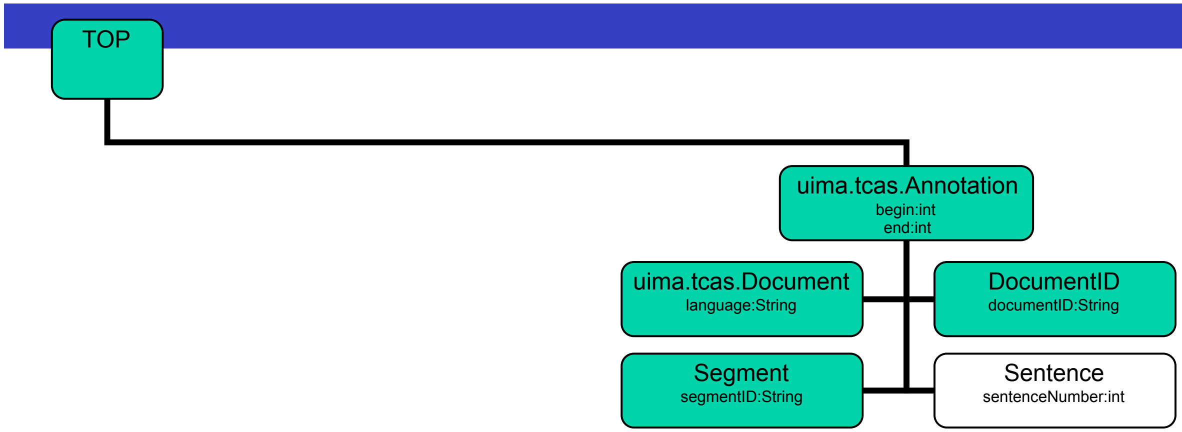


First store the name of the input file, or a handle for the document if it’s from a database

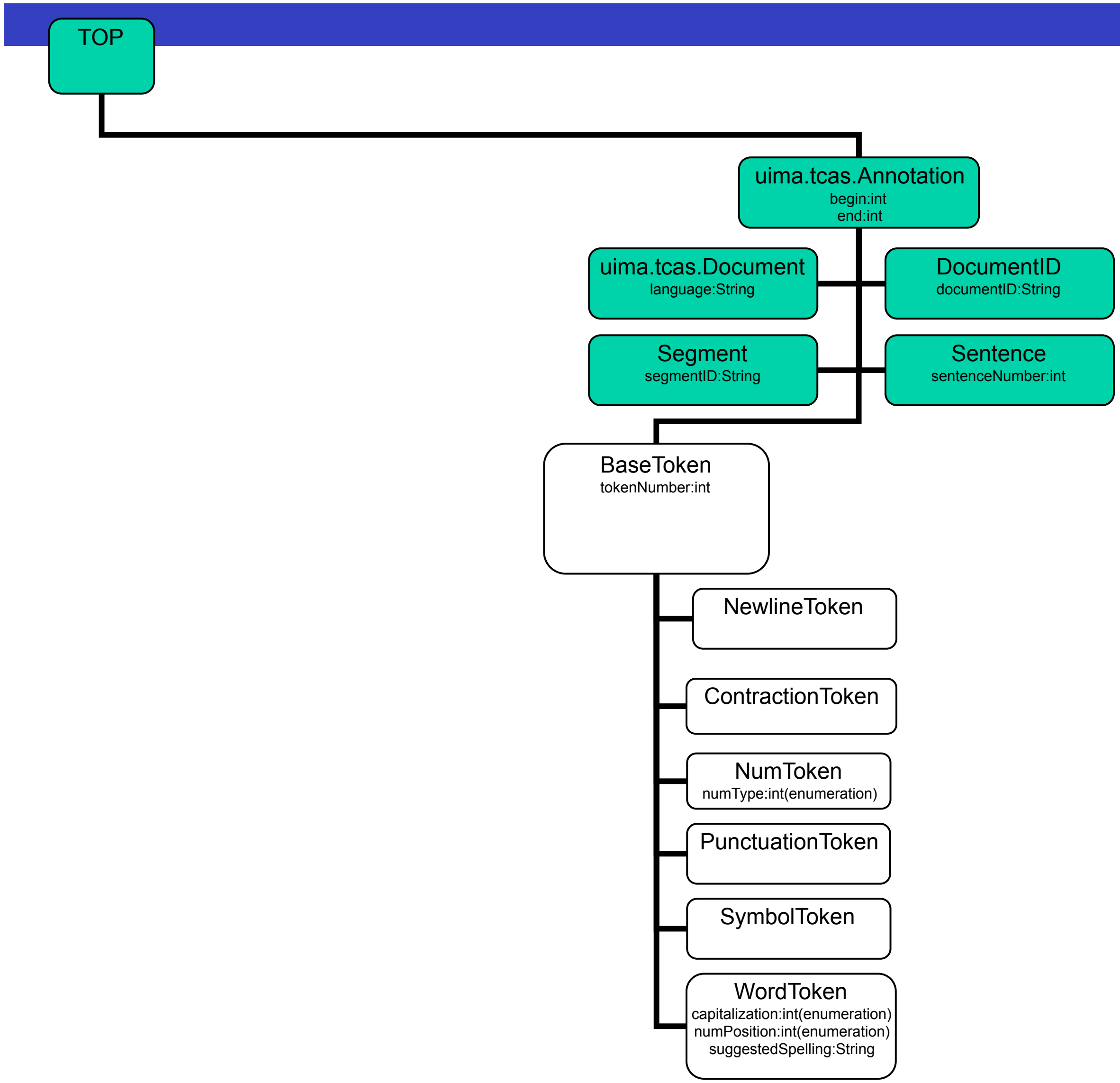


Add Segment annotations based on section tags in the CDA document.

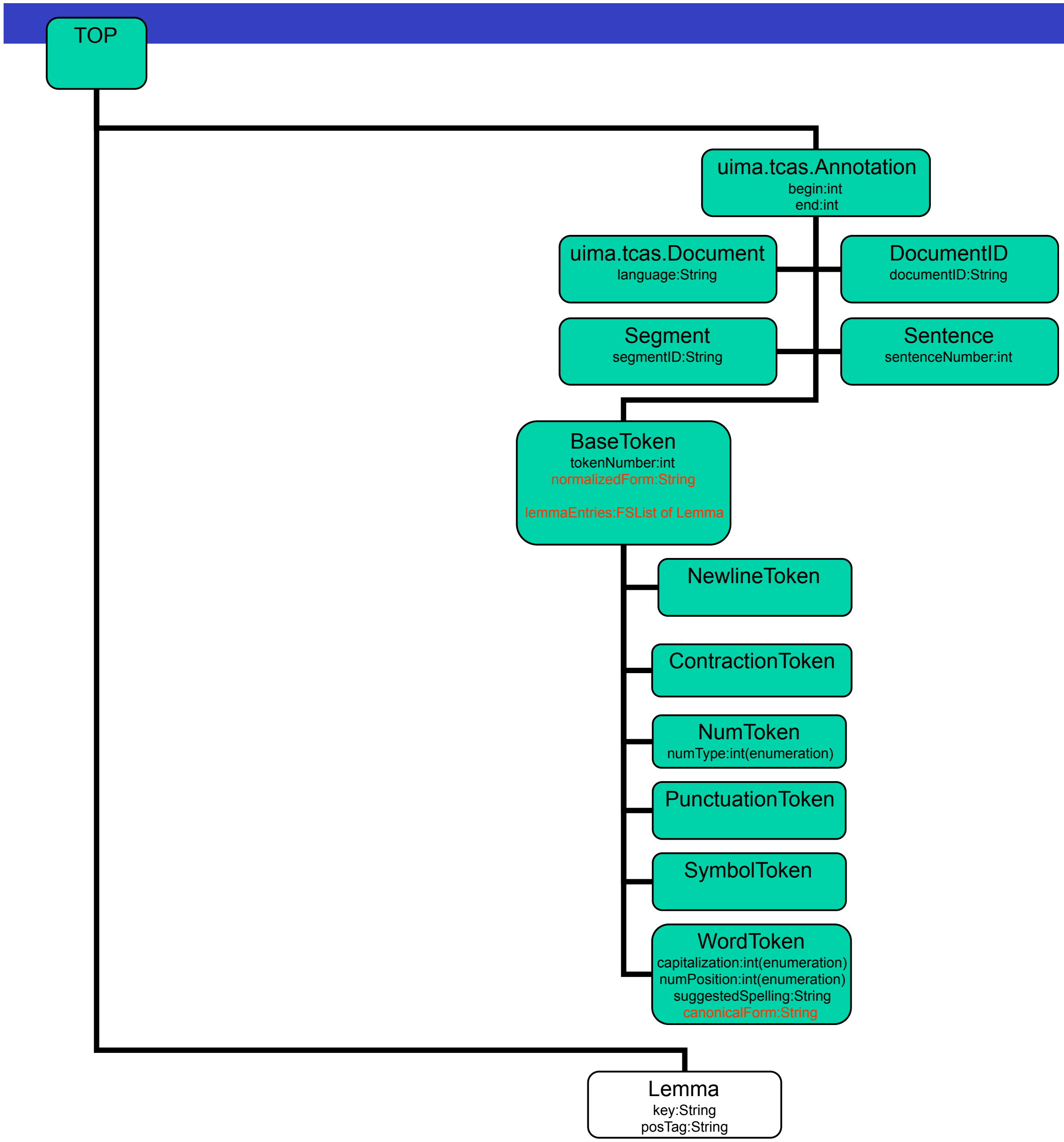
If not processing a CDA document, treat entire document as one section – create one Segment annotation.



SentenceDetector Annotator creates Sentence annotations

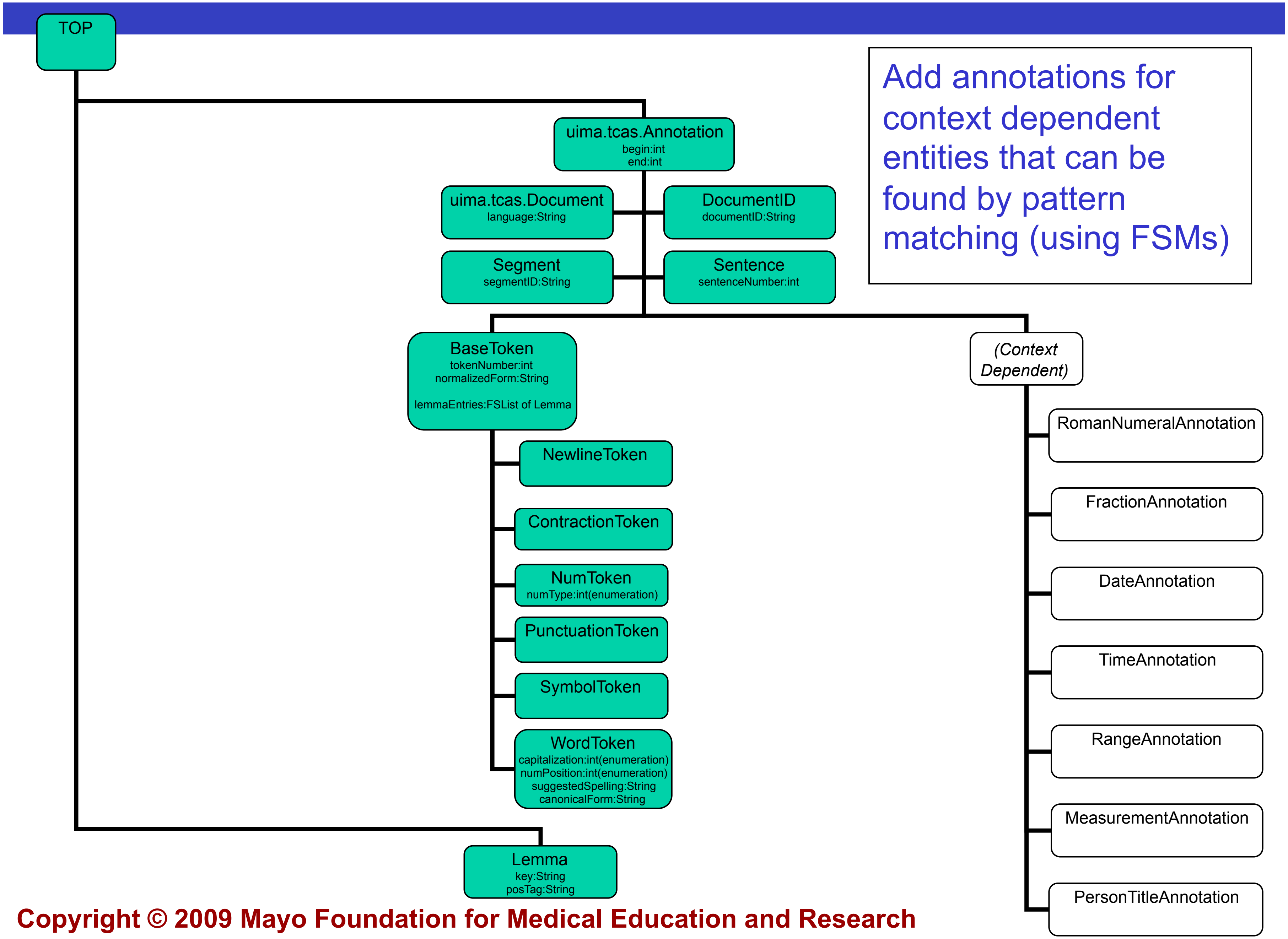


Each sentence is “broken” into tokens by the Tokenizer.



Add attributes to some Token annotations based on output of LVG, and add Lemma annotations.

Add annotations for context dependent entities that can be found by pattern matching (using FSMs)



TOP

uima.tcas.Annotation
begin:int
end:int

uima.tcas.Document
language:String

DocumentID
documentID:String

Segment
segmentID:String

Sentence
sentenceNumber:int

BaseToken
tokenNumber:int
normalizedForm:String
lemmaEntries:FSList of Lemma

NewlineToken

ContractionToken

NumToken
numType:int(enumeration)

PunctuationToken

SymbolToken

WordToken
capitalization:int(enumeration)
numPosition:int(enumeration)
suggestedSpelling:String
canonicalForm:String

Lemma
key:String
posTag:String

(Context Dependent)

RomanNumeralAnnotation

FractionAnnotation

DateAnnotation

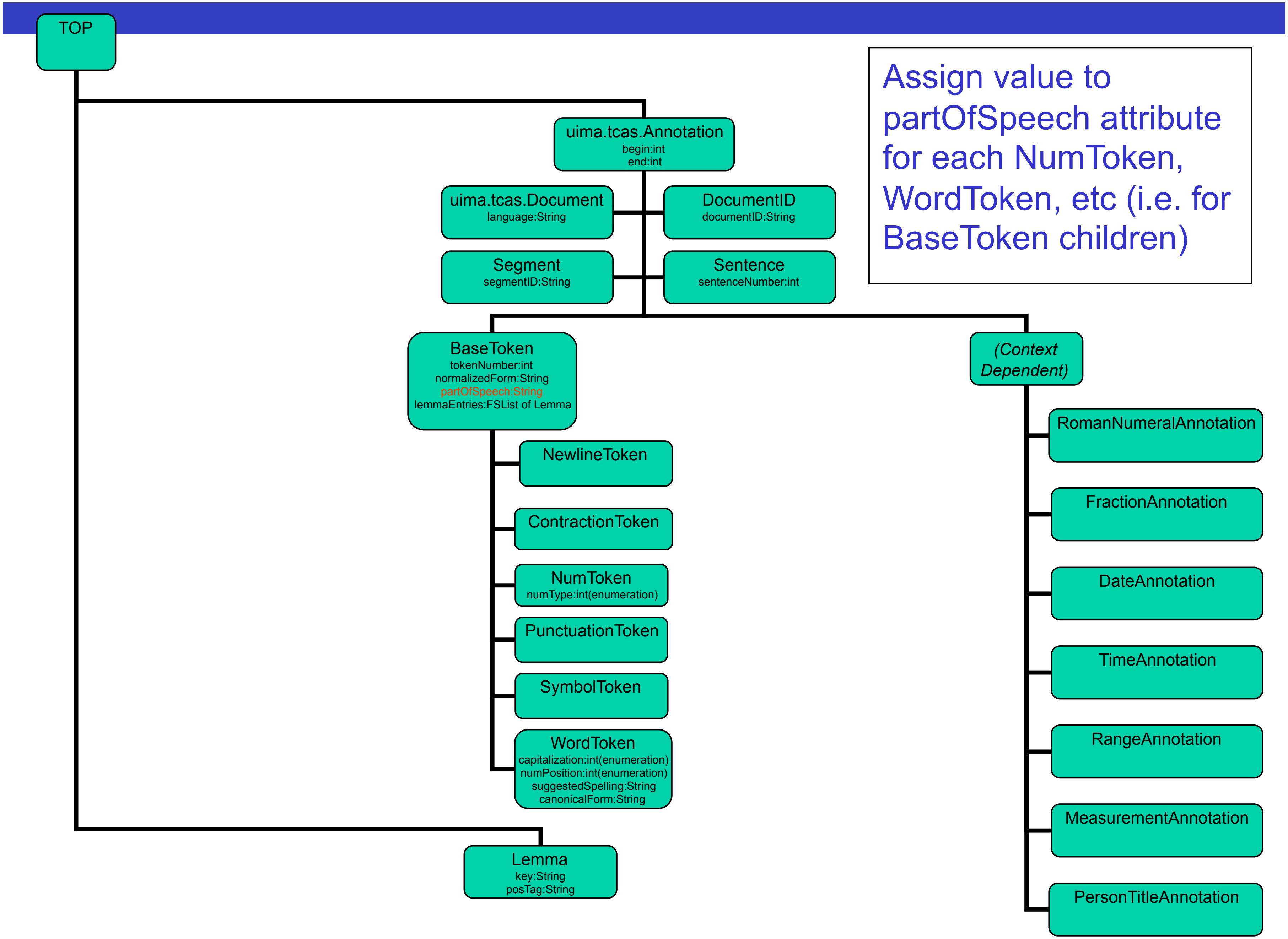
TimeAnnotation

RangeAnnotation

MeasurementAnnotation

PersonTitleAnnotation

Assign value to partOfSpeech attribute for each NumToken, WordToken, etc (i.e. for BaseToken children)



TOP

uima.tcas.Annotation
begin:int
end:int

uima.tcas.Document
language:String

DocumentID
documentID:String

Segment
segmentID:String

Sentence
sentenceNumber:int

BaseToken
tokenNumber:int
normalizedForm:String
partOfSpeech:String
lemmaEntries:FSList of Lemma

NewlineToken

ContractionToken

NumToken
numType:int(enumeration)

PunctuationToken

SymbolToken

WordToken
capitalization:int(enumeration)
numPosition:int(enumeration)
suggestedSpelling:String
canonicalForm:String

Lemma
key:String
posTag:String

(Context Dependent)

RomanNumeralAnnotation

FractionAnnotation

DateAnnotation

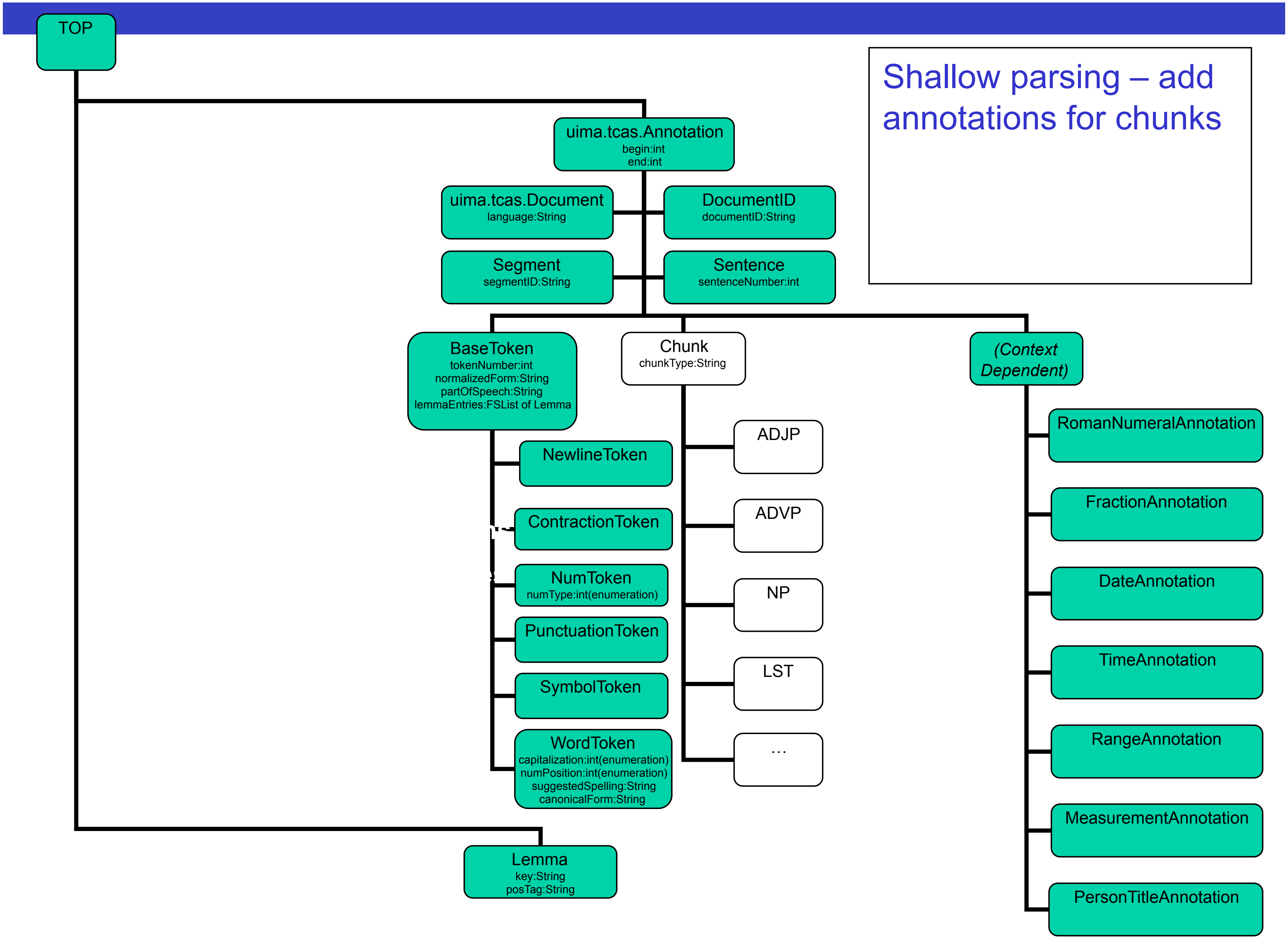
TimeAnnotation

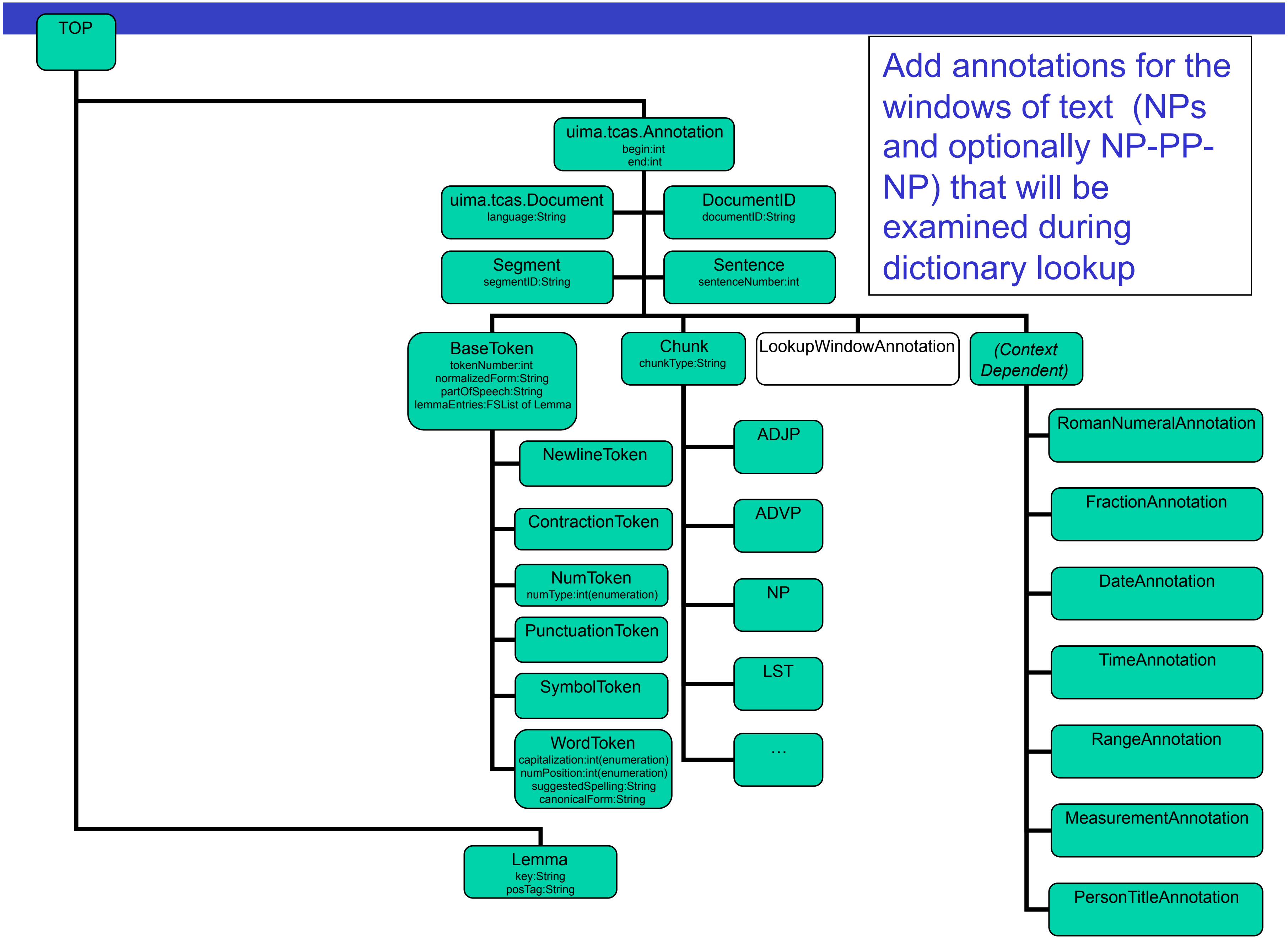
RangeAnnotation

MeasurementAnnotation

PersonTitleAnnotation

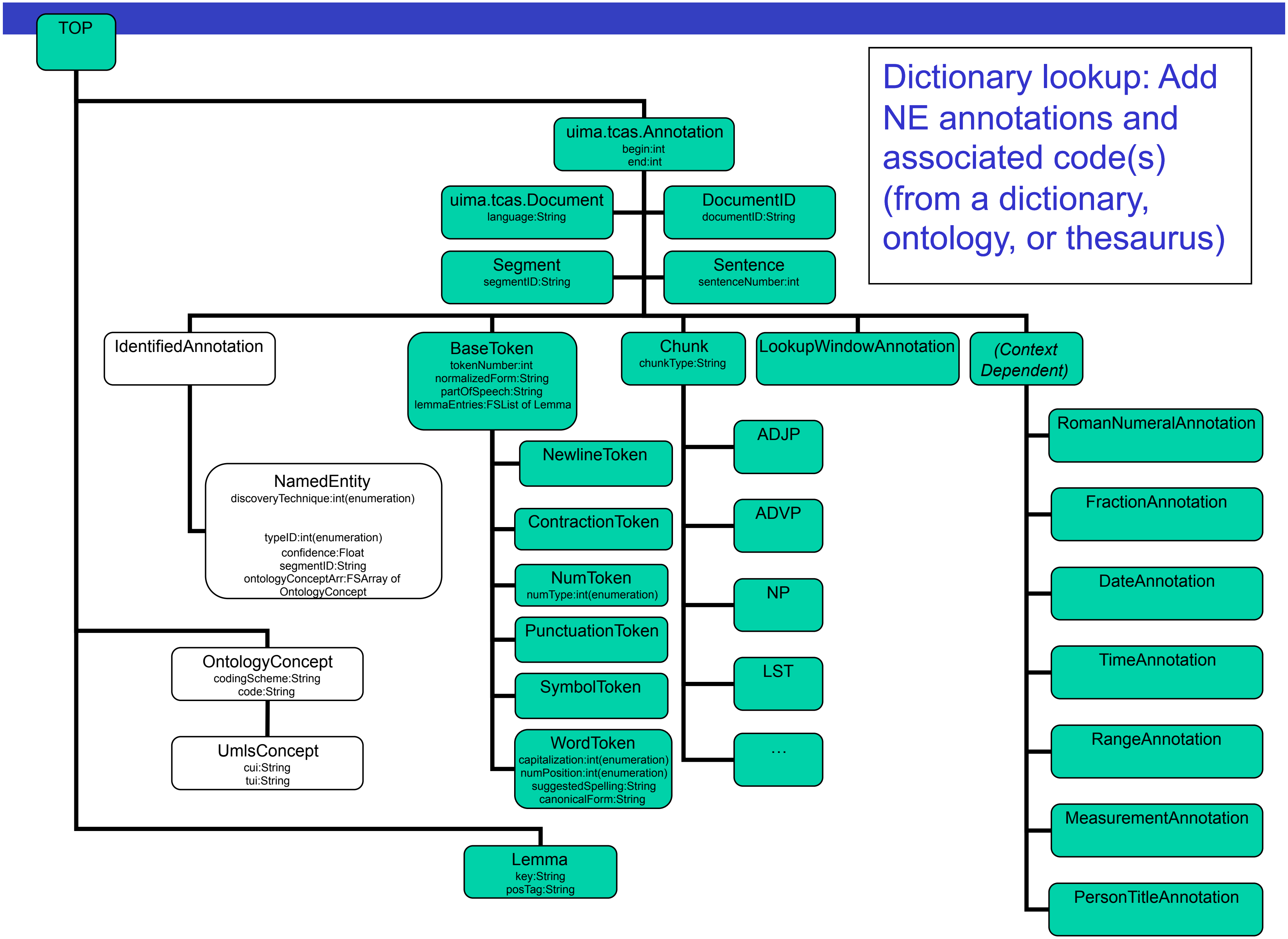
Shallow parsing – add annotations for chunks



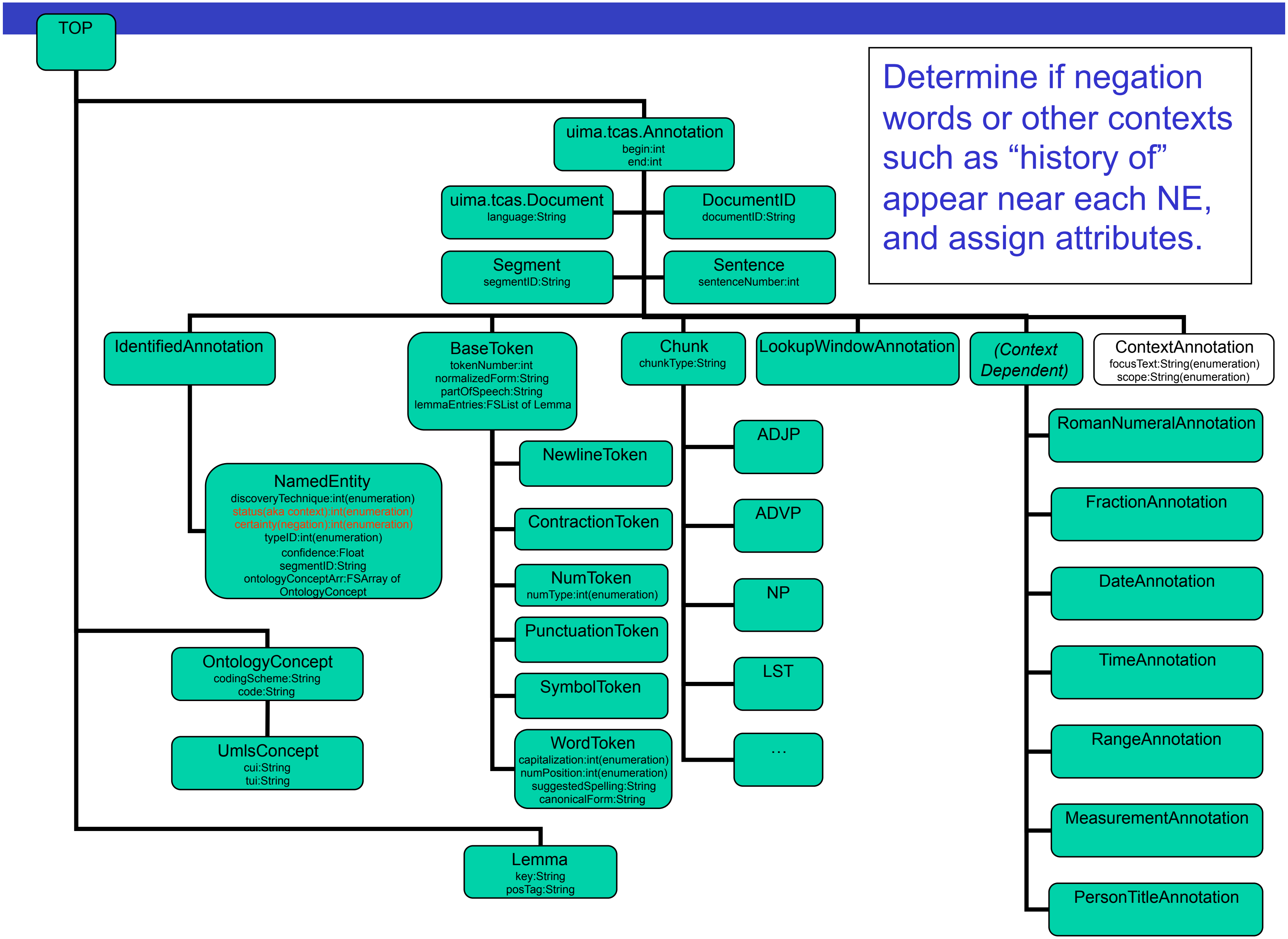


Add annotations for the windows of text (NPs and optionally NP-PP-NP) that will be examined during dictionary lookup

Dictionary lookup: Add NE annotations and associated code(s) (from a dictionary, ontology, or thesaurus)



Determine if negation words or other contexts such as “history of” appear near each NE, and assign attributes.



TOP

uima.tcas.Annotation
begin:int
end:int

uima.tcas.Document
language:String

DocumentID
documentID:String

Segment
segmentID:String

Sentence
sentenceNumber:int

IdentifiedAnnotation

NamedEntity
discoveryTechnique:int(enumeration)
status(aka context):int(enumeration)
certainty(negation):int(enumeration)
typeID:int(enumeration)
confidence:Float
segmentID:String
ontologyConceptArr:FSArray of OntologyConcept

BaseToken
tokenNumber:int
normalizedForm:String
partOfSpeech:String
lemmaEntries:FSList of Lemma

NewlineToken

ContractionToken

NumToken
numType:int(enumeration)

PunctuationToken

SymbolToken

WordToken
capitalization:int(enumeration)
numPosition:int(enumeration)
suggestedSpelling:String
canonicalForm:String

Chunk
chunkType:String

ADJP

ADVP

NP

LST

…

LookupWindowAnnotation

(Context Dependent)

RomanNumeralAnnotation

FractionAnnotation

DateAnnotation

TimeAnnotation

RangeAnnotation

MeasurementAnnotation

PersonTitleAnnotation

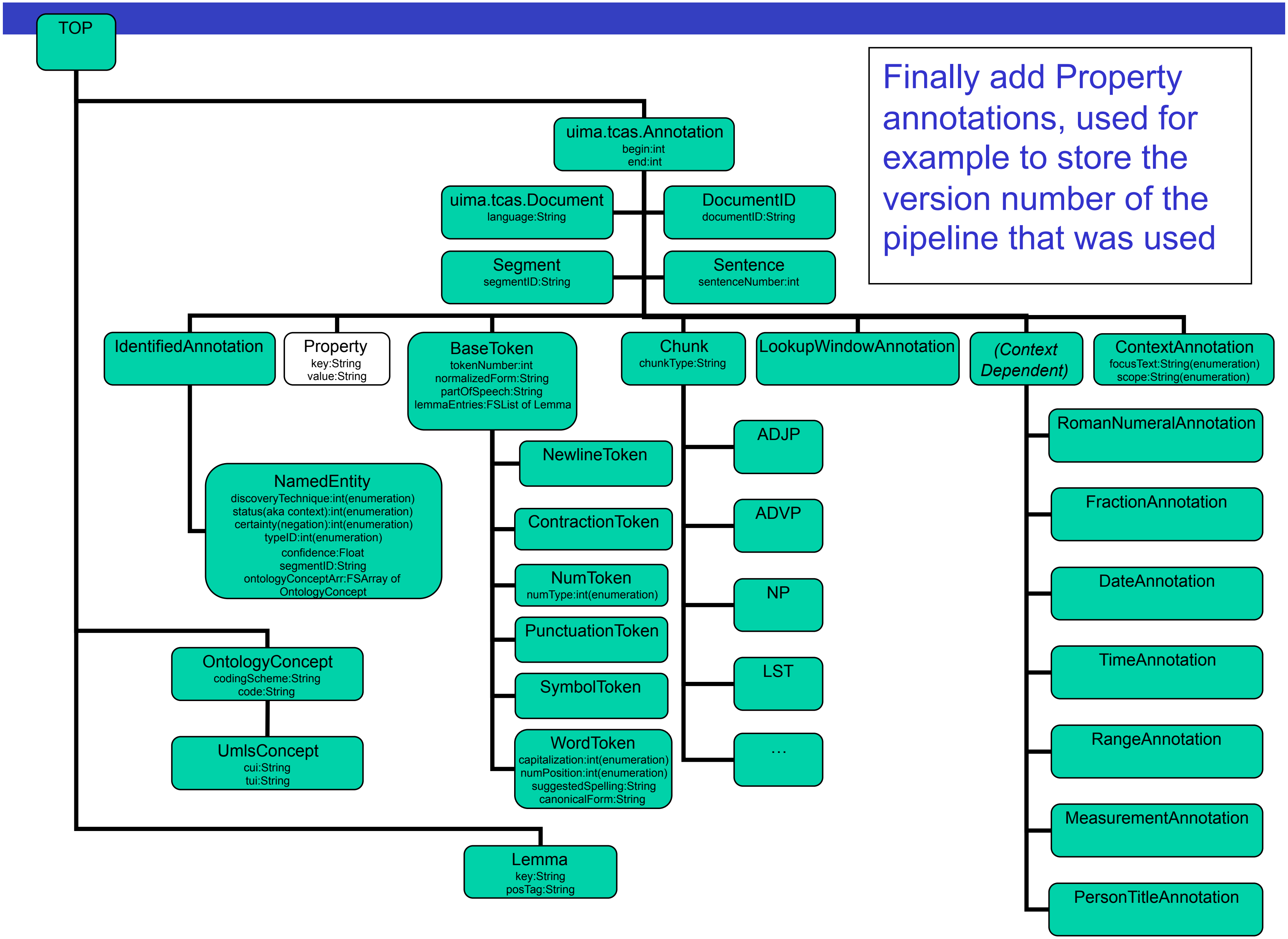
ContextAnnotation
focusText:String(enumeration)
scope:String(enumeration)

OntologyConcept
codingScheme:String
code:String

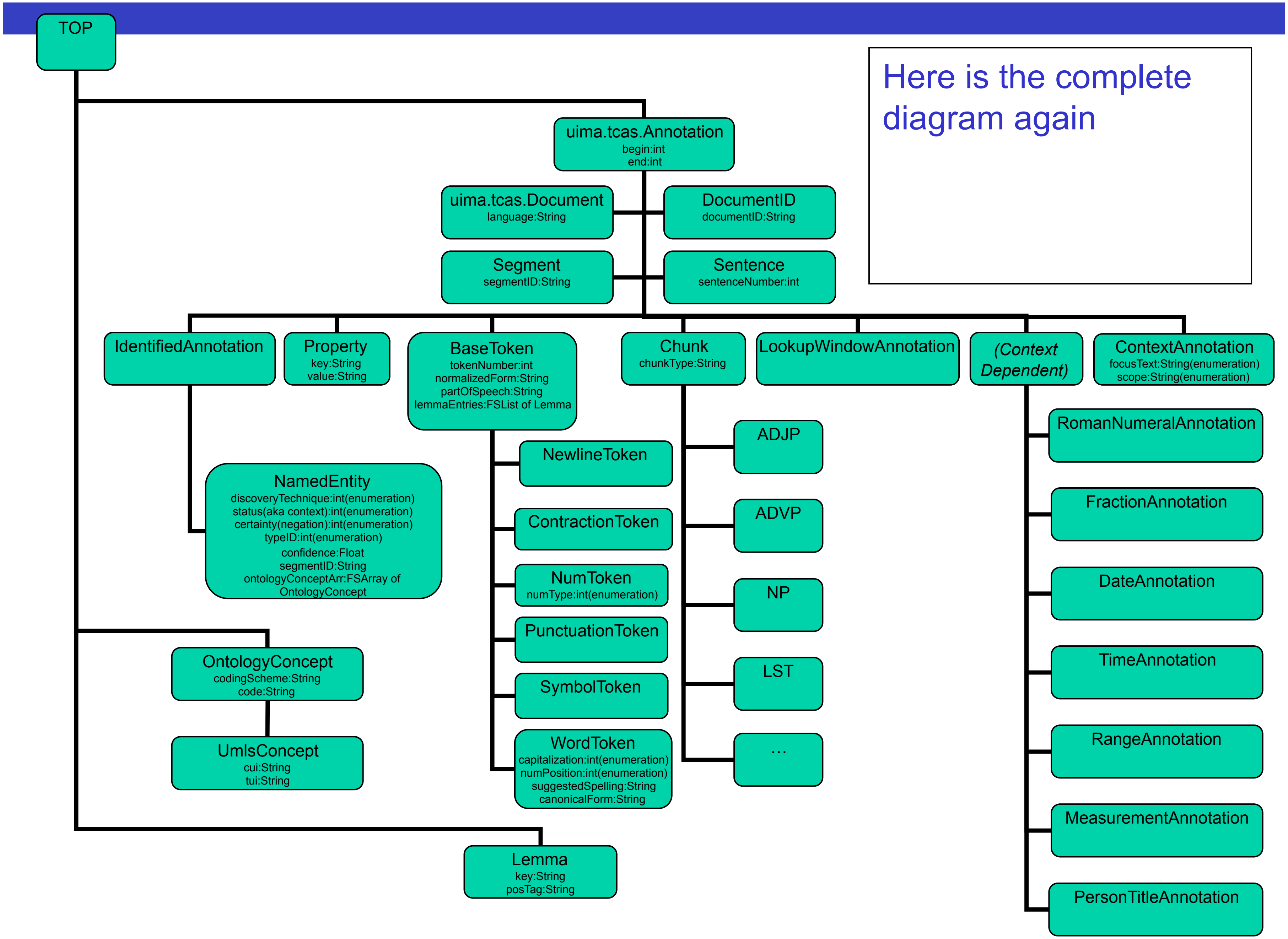
UmlsConcept
cui:String
tui:String

Lemma
key:String
posTag:String

Finally add Property annotations, used for example to store the version number of the pipeline that was used



Here is the complete diagram again



TOP

uima.tcas.Annotation
begin:int
end:int

uima.tcas.Document
language:String

DocumentID
documentID:String

Segment
segmentID:String

Sentence
sentenceNumber:int

IdentifiedAnnotation

Property
key:String
value:String

BaseToken
tokenNumber:int
normalizedForm:String
partOfSpeech:String
lemmaEntries:FSList of Lemma

Chunk
chunkType:String

LookupWindowAnnotation

(Context Dependent)

ContextAnnotation
focusText:String(enumeration)
scope:String(enumeration)

NamedEntity
discoveryTechnique:int(enumeration)
status(aka context):int(enumeration)
certainty(negation):int(enumeration)
typeID:int(enumeration)
confidence:Float
segmentID:String
ontologyConceptArr:FSArray of OntologyConcept

NewlineToken

ContractionToken

NumToken
numType:int(enumeration)

PunctuationToken

SymbolToken

WordToken
capitalization:int(enumeration)
numPosition:int(enumeration)
suggestedSpelling:String
canonicalForm:String

ADJP

ADVP

NP

LST

…

RomanNumeralAnnotation

FractionAnnotation

DateAnnotation

TimeAnnotation

RangeAnnotation

MeasurementAnnotation

PersonTitleAnnotation

OntologyConcept
codingScheme:String
code:String

UmlsConcept
cui:String
tui:String

Lemma
key:String
posTag:String

Questions