CogCompNLP: Your Swiss Army Knife for NLP

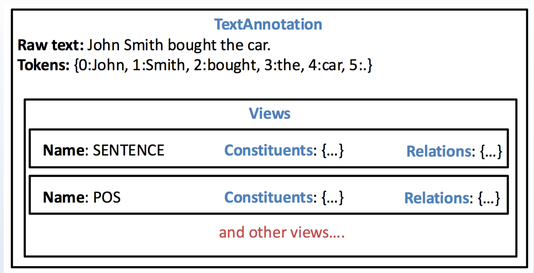
Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev Ratinov, Guanheng Luo, Quang Do, Chen-Tse Tsai Subhro Roy, Stephen Mayhew, Zhili Feng, John Wieting, Xiaodong Yu, Yangqiu Song, Shashank Gupta, Shyam Upadhyay, Naveen Arivazhagan, Qiang Ning, Shaoshi Ling, Dan Roth

Motivation

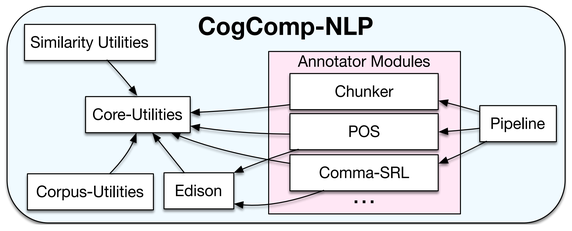
- Supporting Natural Language Understanding applications requires preprocessing text at multiple, syntactic and semantic, levels.
- We call each preprocessing level –from Tokenization to POS tagging, to Semantic Role Labeling, etc–a *Text Annotator*.
- The process of managing and aggregating annotations is labor-intensive and error prone, requiring significant engineering.
- It is essential to building software frameworks for easy access to a wide range of NLP annotators and for straightforward use.

Basic Data-Structures

- *A Text-Annotation* contains the raw source text with its tokenization and other annotation layers
- A *View* is a data structure which contains an annotation structure of a text.
- An *Annotator* is a class which produces a *View* given a text, and potentially some other *Views*.



Components



Core-Utilities

Fundamental data-structures and operators; hence many of the other modules depend on it:
- SQL-like operations on Text-Annotation
- Experiment utilities & statistical significance
- String pattern-matching algorithms
- Utilities for reading and writing annotations.

Pipeline

Provides a simple interface to access Annotator components either individually or as a group.

Edison

A feature extraction framework that extract features to be used by machine learning algorithms. It enables users to define feature extraction functions that take as input the *Views* and *Constituents* created by Annotators.

Corpus-Readers

The corpus reader module includes NLP corpus readers that populate Text-Annotation objects. A few important datasets supported:
- Treebank Shallow Parse
- PennTreebank Constituency Parse
- ACE 2004/2005

Ontonotes 5.0

Similarity Utilities

For calculating semantic similarity between words (e.g. Word2Vec, ESA, etc), phrases, and entities.

Java code: https://github.com/CogComp/cogcomp-nlp

Python code: https://github.com/CogComp/cogcomp-nlpy

Java

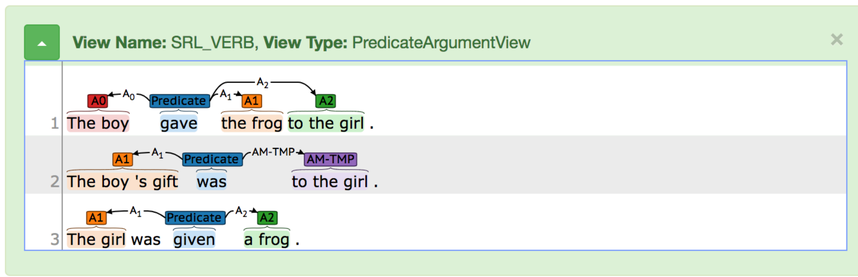
```
// 'ta' is a partially annotated text
TextAnnotation ta = ...
AnnotatorService pipeline = PipelineFactory.buildPipeline(ViewNames.POS,
                                                          ViewNames.NER_CONLL);
TextAnnotation augTa = pipeline.annotateTextAnnotation(ta);
System.out.println(augTa.getView(ViewNames.POS).getConstituents());
// (NNP Pierre) (NNP Vinken) (, ,) (CD 61) (NNS years)...
```

Python

```
from ccg_nlpy import remote_pipeline
pipeline = remote_pipeline.RemotePipeline()
text = "Hello, how are you. I am doing fine"
ta = pipeline.doc(text)
print(ta.get_pos)
# (UH Hello) (, ,) (WRB how) (VBP are) ...
```

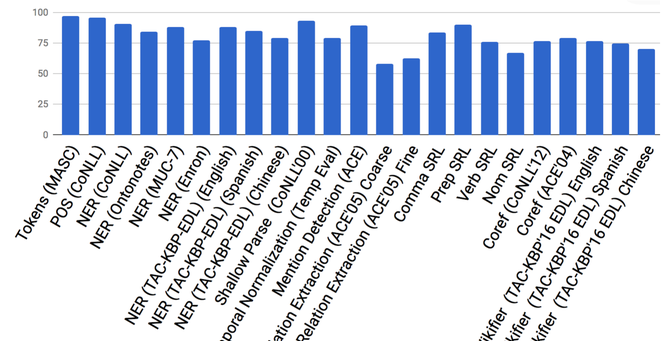
		Sentence splitting	Tokenizing	Lemmatizing	Part of Speech tagging	Shallow Parsing	NER (4 labels)	Extended NER (15-20 labels)	Constituency Parsing	Dependency Parsing	Quantity Normalization	Verb-sense Classification	Temporal Normalization	Mention Detection	Comma SRL	Preposition SRL	Propbank (verb) SRL	Nombank (nominal) SRL	Coreference resolution	Relation Extraction	Sentiment	OpenIE	Wikifier
	CogCompNLP (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Java	CoreNLP	✓	✓	✓	✓		✓	✓	✓	✓			✓	✓				✓	✓	✓	✓	✓	
Java	OpenNLP	✓	✓	✓	✓	✓	✓		✓									✓					
Python	SpaCy	✓	✓	✓	✓	✓	✓	✓	✓	✓													
Python	NLTK	✓	✓	✓	✓	✓	✓																
Python	TextBlob	✓	✓	✓	✓	✓															✓		

Link to demos: http://nlp.cogcomp.org

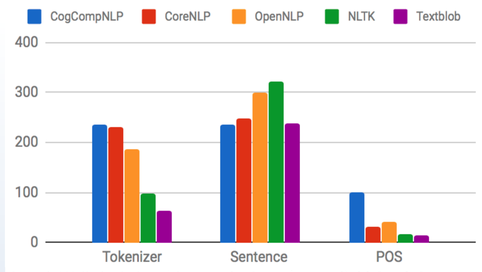


Quantitative Evaluation

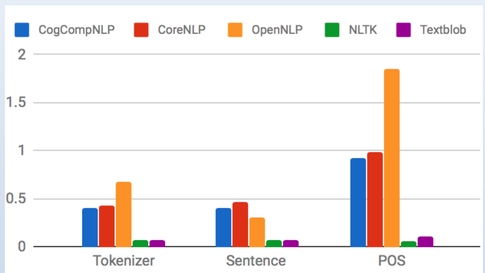
A qualitative assessment of the major components show that they have state-of-the-art quality or very close to the best existing results.



Speed and memory comparison between major NLP pipelines:



Speed: Wall clock time (thousand tokens / second); **higher is better**.



Memory (GB); **lower is better**.

Acknowledgements

Partly based on research sponsored by
- DARPA under agreement number FA8750-13-2- 0008.
- Contract HR0011-15-2-0025 with the US Defense Advanced Research Projects Agency (DARPA).
- Allen Institute for Artificial Intelligence (allenai.org)
- Google
- NSF grant BCS-1348522; and by NIH grant R01-HD054448.