CogCompNLP: Your Swiss Army Knife for NLP
Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev Ratino, Guanheng Luo, Quang Do, Chen-Tse Tsai Subho 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 Annotation**.
- 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.

**CogComp-NLP**

**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

**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.

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

---

**Acknowledgements**
Partly based on research sponsored by:
- DARPA under agreement number HR0011-15-2-0008.
- Allen Institute for Artificial Intelligence (allenai.org)
- Google
- NSF grant BCS-1348522; and by NSF grant R01-HD054448.