EDISON: Feature Extraction for NLP

Mark Sammons¹, Christos Christodoulopoulos¹, Parisa Kordjamshidi¹, Daniel Khashabi¹, Vivek Srikumar¹, Paul Vijayakumar¹, Mazin Bakhani¹, Xinbo Wu¹, and Dan Roth¹
¹Cognitive Computation Group, University of Illinois
²School of Computing, University of Utah

What is Feature Extraction?
Natural Language Processing systems such as Part of Speech taggers, Shallow Parsers, and Named Entity Recognizers are typically built as components of Natural Language Processing (NLP) tools. They are often used to support decision-making processes in NLP tasks. Feature extractors are essential for NLP tasks, as they identify features that are useful for identifying and classifying tasks.

Feature Extraction Challenges
• Implementation is time-consuming.
• Replicating other people’s research is hard – their published descriptions of feature extractors may lack important details.
• There is much duplication of effort: many different versions of the same tools; and different NLP applications, many different implementations of feature extractors.
• Even given existing feature extraction code, it may be hard to find the extractor you need.

Project Goals
• Speed up development by centralizing feature extraction and making it easier for other researchers.
• Clarify individual project code and reduce the number of NLP tasks. Edison specifies a range of generic feature extractors that support Natural Language Processing (NLP), plus utilities such as corpus readers for a number of NLP tasks. Edison specializes feature extraction code using these data structures.

Use Feature Extractors Programmatically

LBJava and Saul
Edison’s feature extractor classes can be directly incorporated in Java applications using the CCG data structures, and in programmable learning environments.

LBJava is a self-contained extension to Java that supports machine learning. It provides a specific language that allows users to quickly develop machine learning algorithms. Saul is a new Learning-Based Programming framework in Scala that supports rapid development of machine learning applications that use Structured Prediction methodologies. It generalizes the capabilities of LBJava.

Using the Edison Feature Extraction Library

Example Task: Named Entity Recognition
Named Entity Recognition (NER) is a basic NLP task that identifies proper nouns and their types in English text. It is a useful component for other applications.

Use Feature File Outputs
SVMLight format
Edison provides support for the use of popular machine learning packages such as Weka, Mallet, and SVMlight by providing classes to write out feature extractor outputs in the SVMlight data format. Edison generates a lexicon mapping feature types to integer values, and SVMlight input files that use these integer values.

Use Feature Extractor Search Interface
Edison users need to know what feature extractors are available, and what they do. Every feature extractor is designed according to a standard to give some idea of what Views and features they generate. Each is documented with a clear description of its behavior, and each has a unit test that illustrates its use and specifies a representative output for that extractor.

Available Resources
Edison features
- Feature Extraction Code
- Feature Output Files
- Learning Framework Application
- Edison Search Interface

Find the source code at:

Copyright 2015 CCG Software