NLTK from the article

David Arroyo Menéndez

March 5, 2019

Presentation

NLTK: The Natural Language Toolkit Edward Loper and Steven Bird Department of Computer and Information Science University of Pennsylvania, Philadelphia, PA 19104-6389, USA
URL: `https://arxiv.org/pdf/cs/0205028` Citations: 1952

Abstract

NLTK, the Natural Language Toolkit, is a suite of open source program modules, tutorials and problem sets, providing ready-to-use computational linguistics courseware. NLTK covers symbolic and statistical natural language processing, and is interfaced to annotated corpora. Students augment and replace existing components, learn structured programming by example, and manipulate sophisticated models from the outset.

Introduction

We need good tools written from scratch by the teachers.

Choice of Programming Language

- A shallow learning curve, so that novice programmers get immediate rewards for their efforts
- The language must support rapid prototyping and a short develop/test cycle; an obligatory compilation step is a serious detraction
- The code should be self-documenting, with a transparent syntax and semantics
- It should be easy to write structured programs, ideally object-oriented but without the burden associated with languages like C++.
- The language must have an easy-to-use graphics library to support the development of graphical user interfaces

Design Criteria

Requirements

- Easy to use
- Consistency
- Extensibility
- Documentation
- Simplicity
- Modularity

Non Requirements

- Comprehensiveness
- Efficiency
- Cleverness

Modules

- Parsing Modules
- Tagging Modules
- Finite State Automata
- Type Checking
- Visualization
- Text Clasification

Documentation

- Tutorials
- Reference Documentation
- Technical Reports

Uses of NLTK

- Assingments (Example: Chunk Parsing)
- Class demonstrations (Example: Chart Parsing Tool)
- Advanced Projects (Example: Probabilistic Parsing)

Evaluation

- A possitive experience for students and teachers
- A problem was find corpora.

Other approaches

- Linguistic Students
- Grammar Developers
- Other Researchers and Developers

Conclusions and Future Work