Introduction to the CLTK

June 8, 2016

Kyle P. Johnson, PhD
kyle@kyle-p-johnson.com
http://cltk.org

Linking the Big Ancient Mediterranean
University of Iowa, June 6-8, 2016
#BAM2016
The CLTK’s goals …

- **Low**: Good datasets for NLP of ancient languages (Egyptian hieroglyphs, Ancient Greek, Latin, Hebrew, Sanskrit, Tibetan, Classical Chinese, etc.)
The CLTK’s goals ...

- **Low**: Good datasets for NLP of ancient languages (Egyptian hieroglyphs, Ancient Greek, Latin, Hebrew, Sanskrit, Tibetan, Classical Chinese, etc.)
- **Medium**: Quantified Classics
The CLTK’s goals …

- **Low**: Good datasets for NLP of ancient languages (Egyptian hieroglyphs, Ancient Greek, Latin, Hebrew, Sanskrit, Tibetan, Classical Chinese, etc.)
- **Medium**: Quantified Classics
- **High**: Framework for an integrated study of the ancient world
... for a connected ancient world
By the numbers

- Began 2014
- 1,523 commits
- 24 contributors
- 27 watchers, 103 stars, 80 forks
- 39 people, 18 teams
- 24 releases (with DOI for every release)
- 81% code coverage
- Supports POSIX OS (and partially Windows)
- 2 students, Google Summer of Code
  - Patrick Burns, PhD (ISAW)
  - Suhaib Khan (Netaji Subhas Institute of Technology, Delhi, India); mentored by Luke Hollis of Archimedes Digital)
Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language
Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language
- Git: Software for distributed software development
Some basic terms

- **Python**: A programming language known for its easy-to-read syntax and general friendliness
- **NLP**: Natural language processing
- **Package or “library”**: Collection of software for a particular set of tasks
- **NLTK**: A prominent NLP package for the Python language
- **Git**: Software for distributed software development
- **GitHub**: A website which makes Git easy
Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language
- Git: Software for distributed software development
- GitHub: A website which makes Git easy
- Jupyter (formerly IPython): “Scientific notebooks”, an easy way to share code
CLTK Archive (by Luke Hollis)

Explore open classical literature

Search...

Read classical works in Greek, Latin, Chinese, Coptic, and Pali
And research metadata on your favorite texts.

Learn More
Get Started

BROWSE POPULAR AUTHORS, POETS, AND HISTORIANS

poemata
catullus

washingtonii_vita
glass

bellum_catilinae
sailust

jugurtha
sailust

agricola	
tacitus

annales	
tacitus
Cui dono lepidum novum libellum
Cui dono lepidum novum libellum
arido modo pumice expolitum?
Corneli, tibi; namque tu solebas
Cui dono lepidum novum libellum
Cui dono lepidum novum libellum
arido modo pumice expolitum?
Corneli, tibi; namque tu solebas
catullus

poemata

no lepidum novum libellum

nodo pumice expolitum?

...
CLTK API

```json
{"language": "latin", "authors": ["plato", "tacitus", "virdi", "catullus", "sallust"]}
```
Design principles

Disintermediation

- Independent of academic bureaucracy
- Software direct into researchers' hands
Design principles

Disintermediation
- Independent of academic bureaucracy
- Software direct into researchers’ hands

Decentralization
- Distributed by Git
- Not “pet project” of one person – of many!
Design principles

- **Disintermediation**
  - Independent of academic bureaucracy
  - Software direct into researchers’ hands

- **Decentralization**
  - Distributed by Git
  - Not “pet project” of one person – of many!

- **Transparency**
  - Public development on GitHub
  - Public, readable code
<table>
<thead>
<tr>
<th>Design principles</th>
<th>Details</th>
</tr>
</thead>
</table>
| **Disintermediation** | • Independent of academic bureaucracy  
• Software direct into researchers’ hands |
| **Decentralization** | • Distributed by Git  
• Not “pet project” of one person – of many! |
| **Transparency** | • Public development on GitHub  
• Public, readable code |
| **Standardization** | • Scientific reproducibility  
• Good basic texts, but editable |
Design principles

**Extensibility**
- Accepting of any proven NLP algorithms
- 100% NLP coverage of all ancient langs
Design principles

- **Extensibility**
  - Accepting of any proven NLP algorithms
  - 100% NLP coverage of all ancient langs

- **Multi-disciplinary**
  - Academic depts, CS, faith traditions
  - Intersection of industry & academe
## Design principles

<table>
<thead>
<tr>
<th>Design Principle</th>
<th>Details</th>
</tr>
</thead>
</table>
| **Extensibility** | ● Accepting of any proven NLP algorithms  
                   ● 100% NLP coverage of all ancient langs |
| **Multi-disciplinary** | ● Academic depts, CS, faith traditions  
                           ● Intersection of industry & academe |
| **Mutual benefit** | ● Full public record of all commits  
                             ● Researchers develop own work |
Design principles

- **Extensibility**
  - Accepting of any proven NLP algorithms
  - 100% NLP coverage of all ancient langs

- **Multi-disciplinary**
  - Academic depts, CS, faith traditions
  - Intersection of industry & academe

- **Mutual benefit**
  - Full public record of all commits
  - Researchers develop own work

- **Inclusion**
  - Collaborative, encouraging
  - Free, easy communication
Design principles

Free & open source

- Fork, modify, merge … whatever!
- MIT licence (OK for commercial use)
Scientific method and communication

1. Make observations
2. Think of interesting questions
3. Formulate hypotheses
4. Develop testable predictions
5. Gather data to test predictions
6. Develop general theories
(repeat)
Scientific method and communication

1. Make observations
2. Think of interesting questions
3. Formulate hypotheses
4. Develop testable predictions
5. Gather data to test predictions
6. Develop general theories
(repeat)

1. Peer review
2. Documentation
3. Reproducibility
   ○ Archiving
   ○ Data sharing
Scientific method and communication

1. Make observations
2. Think of interesting questions
3. Formulate hypotheses
4. Develop testable predictions
5. Gather data to test predictions
6. Develop general theories
(repeat)

1. Peer review
2. Documentation
3. Reproducibility
   - Archiving
   - Data sharing
Scientific method and communication

1. Make observations
2. Think of interesting questions
3. Formulate hypotheses
4. Develop testable predictions
5. Gather data to test predictions
6. Develop general theories
(repeat)

Data sets should be:
- Versioned
- Author-attributed
- Auditable
- Editable
- Easily obtained

1. Peer review
2. Documentation
3. Reproducibility
   - Archiving
   - Data sharing
Technical organization: Repositories
Technical organization: Repositories
Technical organization: Core vs. Data

- CLTK Core software
  - Led by programmers
  - Coordinates data processing
  - Downloads and installs data repositories
Technical organization: Core vs. Data

- **CLTK Core software**
  - Led by programmers
  - Coordinates data processing
  - Downloads and installs data repositories

- **Linguistic data repositories**
  - Led by language experts
  - Plaintext corpora
  - Trained models (for machine learning)
  - Dictionaries, word lists
  - Tagged texts (for part-of-speech, dependency grammar)
Technical organization: Core vs. Data

● CLTK Core software
  ○ Led by programmers
  ○ Coordinates data processing
  ○ Downloads and installs data repositories

● Linguistic data repositories
  ○ Led by language experts
  ○ Plaintext corpora
  ○ Trained models (for machine learning)
  ○ Dictionaries, word lists
  ○ Tagged texts (for part-of-speech, dependency grammar)

● CLTK Archive and API
  ○ Reading environment, with NLP and research extras
  ○ Totally led by Luke Hollis
Personnel organization: People and Teams

- CLTK organization on GitHub
  - 39 People
  - 18 Groups
    - Languages, plus Core, API, website,
  - Admins (a few, mostly housekeeping)
  - Unaffiliated Contributors
<table>
<thead>
<tr>
<th>Username</th>
<th>Name</th>
<th>Teams</th>
<th>Access</th>
<th>Role</th>
<th>Manage access</th>
</tr>
</thead>
<tbody>
<tr>
<td>ancatmara</td>
<td>Oksana Dereza</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>andreasgrv</td>
<td>Andreas Grivas</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>blockspiezer</td>
<td>Brett Lockspeiser</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>christophermorse</td>
<td>Christopher Morse</td>
<td>4</td>
<td>Public</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>coderbhpendra</td>
<td>Bhupendra Singh Chauhan</td>
<td>2</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>diyclassics</td>
<td>Patrick J. Burns</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>evgeniiaraz</td>
<td>Evgenia</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>ferthalangur</td>
<td>Rob Jenson</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>for15pounds</td>
<td>Valerie</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>gree-gorey</td>
<td>Grigory Ignatyev</td>
<td>1</td>
<td>Private</td>
<td>Member</td>
<td>Manage access</td>
</tr>
<tr>
<td>Language</td>
<td>Members</td>
<td>Repositories</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>3</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coptic</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core</td>
<td>21</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Docker</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Greek</td>
<td>5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hebrew</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hindi</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Irish</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latin</td>
<td>5</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(Really quick) quickstart

- Make virtualenv and download core
  - `$ pyvenv venv`
  - `$ source venv/bin/activate`
  - `$ pip install cltk`
- Download and import corpora
  - `$ python`
  - `>>> from cltk.corpus.utils.importer import CorpusImporter`
  - `>>> ci = CorpusImporter('greek')`
  - `>>> ci.list_corpora`
    - `['greek_software_tlgu', 'greek_text_perseus', 'phi7', 'tlg',
    'greek_proper_names_cltk', 'greek_models_cltk', 'greek_treebank_perseus',
    'greek_lexica_perseus', 'greek_training_set_sentence_cltk',
    'greek_word2vec_cltk']`
  - `>>> ci.import_corpus('greek_text_perseus')`
Setup for PHI and TLG corpora

- PHI5, PHI7, and TLG_E
  - Not downloaded, but imported from local files
  - >>> ci.import_corpus('tlg', '~/Documents/corpora/TLG_E/')
  - Makes copy of corpus at ~/cltk_data/originals

- Convert TLG from Beta Code into Unicode
  - >>> from cltk.corpus.greek.tlgu import TLGU
  - >>> t = TLGU()
  - >>> t.convert_corpus(corpus='tlg')
  - >>> t.convert_corpus(corpus='phi5')
  - Makes copy of corpus in ~/cltk_data/greek/text/tlg or
    ~/cltk_data/latin/text/phi5
NLP for all languages

- Concordance
- Information retrieval
  - Plain and regex searching
  - Robust boolean search on the way
- n-gram: 'Ut primum nocte discussa sol'
  - bigrams: ('ut', 'primum'), ('primum', 'nocte'), ('nocte', 'discussa'), ('discussa', 'sol')
  - trigrams: ('ut', 'primum', 'nocte'), ('primum', 'nocte', 'discussa'), ('nocte', 'discussa', 'sol')
  - 5-grams: ('ut', 'primum', 'nocte', 'discussa', 'sol')
- Word frequencies
  - simple count for a word
  - complete reports for a text
- Word tokenization (via NLTK)
NLP for Greek and Latin

- Text normalization
  - \( j \rightarrow i, \, v \rightarrow u \) (Latin)
  - Beta Code conversion (for legacy Greek texts)
  - TLG and PHI corpus specific (remove formatting)
  - Unicode normalization

- Sentence tokenizer
- Lemmatizer
- Stemmer (Latin)
- Word tokenizer, for enclitics (Latin)
- Stopword filtering
NLP for Greek and Latin (cont'd.)

- Named Entity Recognition (NER)
- Part-of-speech (POS) tagger
  - From Perseus/Alphaeus treebank
  - Great work remaining to be done, convert their codes to others (Brill, PROIEL, etc)
- Dependency grammar tagger  # In progress!
- Prosody scanner
- Syllabifier (Greek)
- TLG and PHI5 indices
  - File to author, genre to authors, date to authors, gender to authors, etc.
- Word2Vec
Beyond Greek and Latin

- ~60 repos at https://github.com/cltk
- Chinese, Coptic, Pali, Tibetan, Middle English, Telugu, Classical Hindi, Sanskrit, Hebrew, Aramaic
  - 2.5 GB (!) of Chinese Buddhist texts
  - Coptic texts (via Coptic Scriptorium)
  - Pali Tipitaka
  - Tibetan POS tagged texts and a lexicon
  - Parallel corpora – ready for statistical machine translation (hint, hint)
  - Corpus of ~50 million Hebrew words, ~20 million Aramaic (via Sefaria)
  - Entirety of Perseus/Open Philology
Citation

- Developed by **many talented contributors**!
- BibTex
  ```
  @Misc{johnson2014,
  author = {Kyle P. Johnson et al.},
  title = {CLTK: The Classical Language Toolkit},
  howpublished = \{\url{https://github.com/cltk/cltk}\},
  note = \{\{DOI\} 10.5281/zenodo.<current_release_id>\},
  year = {2014--2016},
  }
  ```
- Chicago author-date
  ```
  ```

*Note: Current DOI release id available at: [https://github.com/cltk/cltk](https://github.com/cltk/cltk)*
Resources

- **This and other lectures**
- **Core software:** [https://github.com/cltk/cltk](https://github.com/cltk/cltk)
  - Bug tracking: [https://github.com/cltk/cltk/issues](https://github.com/cltk/cltk/issues)
    - Beginners’ issues labeled *easy*
- **Project repositories:** [https://github.com/cltk](https://github.com/cltk)
- **Docs:** [http://docs.cltk.org](http://docs.cltk.org)
- **Python + Command line basics**
  - Intro to the command line: [http://blog.teamtreehouse.com/introduction-to-the-mac-os-x-command-line](http://blog.teamtreehouse.com/introduction-to-the-mac-os-x-command-line)
  - Python installation: [https://www.python.org/downloads](https://www.python.org/downloads) (choose 3.5)
  - Good self-paced Python lessons: [http://learnpythonthehardway.org](http://learnpythonthehardway.org)
Contribute & contact

- Classical Language Toolkit
  - Home: http://cltk.org
  - Docs: http://docs.cltk.org/en/latest
  - Source: https://github.com/cltk/cltk
  - Corpora: https://github.com/cltk
  - Import module: https://github.com/cltk/cltk/blob/master/cltk/corpus/utils/importer.py

- Contribute
  - Issue tracking: https://github.com/cltk/cltk/issues
  - Other questions: kyle@kyle-p-johnson.com
Sources

- **Images**

- **Git**
  - GitPython: [https://github.com/gitpython-developers/GitPython](https://github.com/gitpython-developers/GitPython)
  - [https://en.wikipedia.org/wiki/Git_(software)](https://en.wikipedia.org/wiki/Git_(software))

- **Science**