# Lexos

# Summary

Lexos is a suite of tools designed to facilitate the computational analysis of literary and historical texts. It offers an integrated workflow in which the pre-processing (“scrubbing”), analysis, and visualization steps can be accomplished in a single, web-based environment. Scrubbing features include handling punctuation, stop words, markup tags, and character consolidations, as well as document segmentation, culling, and n-gram tokenization. Analytical tools include basic document statistics, hierarchical and k-means cluster analysis, rolling window analysis, cosine similarity ranking, and z-score analysis. Visualizations include word and bubble clouds, comparative “multiclouds” (which can be used to analyze data from [MALLET](http://mallet.cs.umass.edu/)-produced topic models). Analytical tools produce line, PCA, Voronoi cell, and dendrogram graphs. Each of the tools has export functionality.

Lexos is aimed at entry-level users as well as advanced scholars using small to medium-sized text corpora. It places particular emphasis on the processing of ancient and non-standard languages, as well as non-Western languages that do not use the Roman alphabet.

# Release history

This repo reflects our Summer 2016 development work: Lexos v3.0b (currently unreleased).

Summer 2015: Our latest stable release (version 2.5) is available at <https://github.com/WheatonCS/Lexos/tree/masterSummer2015> and is running on our server at [lexos.wheatoncollege.edu](http://lexos.wheatoncollege.edu).

Summer 2014: Version 2.0 <https://zenodo.org/record/10956#.VXWcakZWJ-8>.

You can read more about our [Lexomics Research Group](http://lexomics.wheatoncollege.edu).

## System Architecture (in brief)

Lexos is written primarily in Python 2.7.11 (as distributed in [Anaconda](https://www.continuum.io/downloads) 4.1.1) using the [Flask](http://flask.pocoo.org/) microframework, based on Werkzeug and Jinja 2. The front end is designe using [jQuery](https://jquery.com/) and the [Bootstrap 3](http://getbootstrap.com/) framework, with a few functions derived from [jQuery UI](https://jqueryui.com/). We increasingly incorporate the wiz from [D3.js](http://d3js.org/) in our visualizations and the power in the [scikit-learn](http://scikit-learn.org/stable/) modules for text and statistical processing. The directions for setting up the development environment for testing (using localhost:5000) on your local machine are stored in the 0\_InstallGuides directory.

## Dependencies

flask, matplotlib, numpy, pip, scikit-learn, scipy, pip (gensim, ete2, natsort, chardet)

The PDF Viewer extension needs to be enabled in the Chrome browser on MacOS.

## License information

See the file LICENSE for information on the terms & conditions for usage and a DISCLAIMER OF ALL WARRANTIES.

## Citation information:

Kleinman, S., LeBlanc, M.D., Drout, M. and Zhang, C. (2016). Lexos. v3.0. https://github.com/WheatonCS/Lexos/.