Patrick BURNS

Multiplex Lemmatization with the Classical Language Toolkit

The Classical Language Toolkit is an open-source Python framework supporting natural language processing for historical languages, including Latin. CLTK aims to offer components for all stages of the NLP pipeline and one of its most active areas of development is lemmatization. This presentation will review three ways in which a "multiplex" strategy is used for lemmatization in the Classical Language Toolkit: backoff tagging, ensemble tagging, and the development of wrappers for external components. Backoff lemmatization is currently available for Latin and Greek in the CLTK; ensemble lemmatization and wrapper development are areas of current development.

1. Backoff tagging allows CLTK users to conceive of a lemmatizer not as a single tagger but rather as a customizable suite of sub-lemmatizers, based on the SequentialBackoffTagger in the Natural Language Toolkit (http://bit.ly/nltk-sequential). SequentialBackoffTagger allows the user to "chain taggers together so that if one tagger doesn't know how to tag a word, it can pass the word on to the next backoff tagger." (Perkins, J. *Python 3 Text Processing with NLTK 3 Cookbook*, p. 92.) This kind of composite method allows users to mix and match different lexicon-based, rules-based, and probabilistic taggers in service of efficient, accurate lemmatization, helping in particular with the disambiguation of high- frequency though ambiguous forms (e.g. *est* > *esse* as opposed to *est* > *edere*) as well as the lemmatization of out-of-vocabulary words (*strugitant* > **strugito*, cf. anonymous Latin translation of Carroll's *Jabberwocky*).

2. Ensemble tagging will similarly allow CLTK users to build composite methods for lemmatization. Here, borrowing from methods used in machine learning (e.g. http://bit.ly/sklearn-ensemble), we would adduce a number of taggers, assign probabilities to each tagger's outputs, and use a voting "classifier" to choose (or rank) likely lemmas. This method would offer similar benefits to backoff tagging, but whereas the backoff method cuts short the chain of taggers as soon as it finds a match, the ensemble tagger arrives at a final decision based on all possible outputs from each of the sub-taggers. Accordingly, ensemble lemmatization is less efficient, though it should boost accuracy.

3. Lastly, I have begun developing wrappers—that is, programming interfaces that allow code from one domain or language to be used inside another—for the CLTK in order to support external lemmatizers. Lemmatization is (and has been) an active area of research in digital Latin philology and this research has produced excellent standalone programs and packages. The CLTK does not position itself as a competitor to these projects, but rather aspires to be a collaborator. For this reason, I have begun to develop Python wrappers for TreeTagger, Whitaker's Words, Morpheus, LemLat, Latmor, and Collatinus among others that either incorporate web services or capture command line output so that lemmatizer results can be easily incorporated into the CLTK

backoff and ensemble methods, so giving our users maximum flexibility in the construction of their digital philological pipeline. The challenge here (and likely one of dominant topics of conversation at this workshop) is the standardization of inputs and, especially, outputs. A linked-data approach to Latin lexical data would greatly assist with this kind of framework interoperability.