Thesaurus WriteUp

Why do I need a thesaurus?

A moldable thesaurus is seemingly very important for my project. I'm looking at building a model to shift the tone of a body of text while attempting to preserve semantics. Simply going off of intuition, we can figure that swapping a word for a synonym of that word may alter the tone. We would, then, like to aggregate some relative relationship between some set of tones and groups of synonymous words. Upon creating such a dataset, we may interchange a word with a synonym according to a difference in tone.

Why not use an existing API?

There are a few reasons to reconstruct our own thesaurus. First and foremost, we are interested in keeping additional data regarding tone that is not present within existing thesaurus APIs. It is far easier to manipulate data if it is all on hand, and it should save some server-side complexity in dealing with the decoupling of the thesaurus and the word-tone relationships. There is an issue with cost as well: APIs are rarely free past some established number of calls in some time interval, and I would like for this software to function with minimal cost–ideally the only cost is in hosting. Finally, this is a project in the realm of software engineering. While it is often better to rely on existing services–standing on the shoulders of those who came before you–it is also important to know how to create your own services.

How did we do it?

We created the thesaurus by web scraping, which is a large aspect of data-collection and thus is often a necessity in the sphere of machine learning. Of course, it is possible to compile a thesaurus using other means–surely one could buy a physical thesaurus and type up all the entries or even automate such menial tasks with computer vision–but we are interested in constructing a thesaurus as painlessly as possible. It is only one aspect of our project after all.

Web scraping often comes with a give and take. There are several existing online thesaurus services, so we did looked into a few of them and landed on John Watson's Big Huge Thesaurus. We began with trying thesaurus.com, but they have protections against traditional scraping. There are many of such protections including lazy-loading content, providing fake data, services like Captcha, even automatically altering the page's HTML. It seemed as if [thesaurus.com](www.thesaurus.com) had been randomizing their CSS classes and either lazy-loading content or providing fake data. While we could likely get around this with an automated browser like [Selenium](https://www.seleniumhq.org/), as the CSS classes are only a deterrent if scraping over a large time interval and the content would almost certainly exist within an automated browser session, we should respect that this site has practices in place to prevent scraping.

There may be protections against web scraping in place that we ought to honor, or there are often poorly laid out websites that would be a pain to use despite being open source, or the site may simply not provide all the information we would like. The latter is best exemplified by [moby](http://moby-thesaurus.org) which we may come back to if we decide that we care not about parts of speech. [John Watson's Big Huge Thesaurus](words.bighugelabs.com), on the other hand, seems to have all that we want: synonyms by part of speech, a clean interface, and permission for use given credit is provided.

Protections against web scraping from JonasCz