Feature Extractor Details

**Background**

* Originally looked into exclusively using NLTK to achieve this
* Thought this would be a relatively simple task to complete using its various tools (POS tagger, Parser, etc.)
* Gave accurate results on small test sets, but performance was TERRIBLE.
* Decided to simplify task and focus on objective features of reviews (so mostly nouns and noun phrases).
* Like discussed in class, the main parts of a sentence are typically Noun Phrases (NP) and Verb Phrases (VP). The NP’s usually make up what the sentence is about (topic), which is exactly what we want in this case.
* Now, since we want NP’s, we are only concerned with a few basic tag sequences (ex: “NN”, “NN+NN”, “JJ+NN”, etc.). These are the tag patterns that more than likely correspond to the features we are looking for.
* So instead of focusing on full parses of the sentence structure, we look for tag patterns that “most likely” resemble noun phrases we are looking for.
* This method is not quite as accurate and comprehensive as using just NLTK tools, but the significant increase in performance outweighs these drawbacks. Especially for a task like our project.

**Implementation**

* define a basic “CFG”(dict) that corresponds to the basic tag sequences that we are looking for.
* define a tagger that is built from NLTK tagger classes, trained on the brown corpus
  + This is a bigram tagger that uses Katz back-off with a unigram tagger
  + Much like how we simplified the tag sequences we were looking for, we can simplify some of the brown corpus tags. This is done by renaming many similar tags with the same tag (for ease of searching). We can do this because there are many tags that we are not concerned with (think of the O tag in BIO tagging examples)
* Finally, we split/tokenize the sentence, tag it using our custom NLTK tagger, and search for any tag sequences that match any sequences in our defined basic “CFG”

**Putting it all together**

* These features are then collected and the most common features per each business are added to the “attributes” list of that business (so potential attributes user can enter as input)