The textbook I used for learning about NLP was [Speech and Language Processing by Jurafsky and Martin.](http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210) I have a physical copy and a PDF version of the book to use for reference. If you wanted, you could rent/buy your own copy, and I also hear that some sort of Buccaneer Cove or something like that has a PDF available for download.

Some of the features I think an NLP scientist might want to use in a program are listed below. Note that most of these already have libraries which exist to do them - no sense reinventing the wheel for something like lemmatization if our code can just reuse an existing algorithm:

* Regular expression checking
* File conversion: in from and out to any conceivable data source depending on the data structure - .doc, .pdf, raw text, JSON/XML, SQL database entries are the first ones that come to mind as common formats.
* Text Preprocessing
  + Stemming (combining words with the same root, i.e. studying, studied, and study all get converted to study), see also lemmatization, which does the same thing while inferring the part of speech (to avoid ambiguity like Cook as a name, cook as a job, and cook as a verb)
  + Part of speech tagging/Word sense disambiguation (figuring out what type of word each word in the text is)
  + Sentence/word segmentation (finding the breaks between words and sentences, especially when there’s ambiguity like if someone wrote Mr. Smith)
  + Pronunciation annotation (given words, annotate them with the standard pronunciation alphabet for analysis with speech algorithms)
* Automated summarization
  + Given a block of text, figure out the important bits and give a summary
* Sentence parsing
  + Given a sentence, diagram the sentence, building a “parse tree” showing how the words combine into phrases following a grammar
* Question answering
  + Parse a question (perhaps with contextual clues), and use some knowledge source to answer it
* Sentiment analysis
  + Given a block of text, examine it for words indicating emotion (perhaps on a positive-negative spectrum, or maybe a more detailed breakdown of emotion) and describe how strong the sentiment is
* Relationship extraction
  + Given a block of text, identify the entities in it and their relationships (Bob is Jill’s husband, Max is Bob’s dog, the corvette is Jill’s car, etc.)
* Speech recognition
  + Anything involving sounds and speech is really its own separate area that is quite different from standard text data, IMO. This would be a great thing for a different team to tackle in a year, I think.
* Topic Modeling
  + This is something I didn’t learn about but my professor added to the curriculum for this year’s class. She recommends two toolkits for working with that, GenSim and Mallet.

<http://en.wikipedia.org/wiki/Outline_of_natural_language_processing#Natural_language_processing_toolkits> This article is outstanding, and the linked section discusses existing toolsets for NLP data. In short, Java and Python are by far the best supported languages for these toolkits, although there is one toolkit in Ruby, one in Javascript/Node, and several in C++.

<http://www.nltk.org/book/>