*http://stackoverflow.com/questions/17864466/flatten-a-list-of-strings-and-lists-of-strings-and-lists-in-python*

Code segment to flatten a list of lists.

*#http://codereview.stackexchange.com/questions/90692/removing-all-stopwords-from-a-list-of-words*

Code segment to remove stop-words from a list of words

<https://code.google.com/archive/p/word2vec/>

Link to different large corpuses (I’m using the Google News corpus)

<https://github.com/japerk/nltk-trainer>

Github Repo on how to train and evaluate using NLTK in python, including tokenization and tagging

<http://www.aclweb.org/anthology/W11-3709>

Paper on the sentiment analysis of words using intensity, longevity, and timing. Gives me a solid groundwork on how to think about analyzing the strength of the emotions in the stories.

<https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-3-more-fun-with-word-vectors>

Example on how to cluster words based on emotion (only using positive/negative emotion though). Taken from a Kaggle tutorial.

<http://www.aclweb.org/anthology/C10-1104>

Paper on how to analyze emotion in text. Mostly focuses on Chinese text, but gives a formula for how they value the emotions, as well as points me towards the Maximum Entropy Modeling Toolkit (<http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html>), which I’m going to start looking in to.