NLP Project Proposal
Vanna Willerton and Greg Theos

The purpose of the project will be to create a pun generator. The proposed project is relevant as it ties into multiple topics discussed in class (including semantic relatedness and similarity), as well as some things which were not covered in class but represent important linguistic information (including phonetics). Below is an outline of the proposed project, including a description of the generator, relevant existing tools to be used, and possible methods.

The Pun Generator

Input: A user is prompted for two different inputs. 1) a topic of discussion; and 2) a phrase/sentence to modify (max number of words or characters to be decided). The topic will be used to find a number of related and similar words with which to modify the sentence.
Output: The input phrase modified with pun(s) wherever possible.
Example:
Input Topic: Ocean
Input Sentence: *Oh well, it's time for bed*. (or *I see what you did there*)
Possible output: *Oh whale, it's tide for bed.* (or *I sea water you did there*)

Necessary Tools

1) A model of Semantics for measuring relatedness and similarity of words. We will used existing pre-trained *word vector evaluation tools* discussed in class (http://wordvectors.org/suite.php).
2) English to IPA transliteration tools. There are many existing open source python programs designed for this task, including *epitran* (https://pypi.org/project/epitran/0.4)
3) A way of measuring how much a string is edited by modifying input words with related and similar words. We can use existing measures such as *Levenshtein distance*.
4) A phonotactics scorer for ensuring the edited strings make words which sound like valid English. This can be done using existing python scorers such as *python-BLICK* (https://pypi.org/project/python-BLICK/ implemented by someone from McGill's very own Linguistics Department).

Proposed Methods/How it Works

1) The topic word is run through a word vector evaluation tool to get the top X similar and related words. (X to be decided)
2) Input sentence, topic word, and related words are all converted to IPA; all words are lemmatized so we don't add to distance by comparing nouns to verbs or inflected verbs to uninflected forms. For example, we would want *bark/parking* to be considered a close match.

3) Each of the related words are compared to each word of the input phrase using Levenshtein distance. Distances over some point (TBD) are rejected, those that fall under continue to the next step.
4) Remaining possible modifications are put through the BLICK phonotactics scorer. Scores worse than some point (TBD) are rejected.
5) Output results. Note that the cutoff strategy means that in some cases there will be no viable modifications. For simplicity we are allowing a null output. Also, lacking a competition step, multiple modifications for one output are also possible. We see this as a feature.