**Project: NLP Analysis of four different subreddits**

Project commencement date: 11 June 2021

RESEARCH AIM 🡪 PROJECT DESIGN/METHOD 🡪 PROJECT STEPS EXECUTED 🡪 PROJECT RESULT

Scrape text from four different subreddits (you’ve got the code for some of it already):

1. /r/productivity
2. /r/antiwork
3. /r/skeptic
4. /r/psychic

**Thinking about my research aim**

* To think about my research aim, I have to ask myself the question: what do I want the OUTPUT or RESULT of this project to be?
* I know what I want the INPUT to be: plain text data from four subreddits: /r/productivity, /r/antiwork, /r/skeptic and /r/psychic. I chose these four subreddits for their representation of opposing sides on two different spectrums:

PRODUCTIVITY (capitalist convention) 🡨🡪 ANTIWORK (contrary to capitalist convention)

PSYCHIC (conforms to magical thinking) 🡨🡪 SKEPTIC (challenges magical thinking)

* OUTPUT: Do I want some pages of written insights from the analysis of the reddit language data? – YES.
  + Do I want any additional output to this page of written insights?
    - Depends. I think obtaining the written insights will at least be a good minimum result.
* METHOD: Need to research what methods to use. This will heavily depend on the research questions I choose to investigate as well.

**POTENTIAL RESEARCH GOALS FOR THIS PROJECT:**

1. **Word frequency distribution analysis – what are the most frequently occurring words (open-class) that occur in each of these four subreddits? Do these word choices indicate any patterns in interaction in this subreddit? Collocation analysis – what kinds of words tend to occur together in this subreddit?**
2. **Sentiment analysis – overall, what sort of sentiments are conveyed from some redditors to others in the subreddits (could be particularly interesting for**
3. **Comment generator – could my analysis of word frequency in the above subreddits lead to a script that could randomly generate a comment for each of these subreddits? It would not be a response to any particular question, just a general comment that a user would be likely to write. What would it look like? Method: n-grams**

**STEPS LOG**

Modules imported: praw, nltk (tokenize, FreqDist)

1. Find the Python program you got already with the reddit bot and scraper…
   1. DONE – Found the sentiment\_analysis.py program from last year.
2. Access reddit using personal credentials
3. Initialise variables to the four subreddits (Antiwork, Productivity, Psychic, Skeptic)
4. Find top 100 rated comments on subreddits (not sure if I’ve picked out top rated or just a random sample of comments here – perhaps make a distinction between top-rated [influential] comments and random comments?)
5. Create corpus training data for each subreddit
6. Tokenize corpus training data so that the words become meaningful tokens
7. Analyse frequency distribution of tokenized corpus data (filter to only include words of more than length three – more likely to be meaningful, open-class words as opposed to closed-class grammatical words)
   1. Plot these distributions visually in a graph (top 50 most frequent words)

IDEAS FOR WORD FREQUENCY DISTRIBUTION ANALYSIS SECTION

* Sort most frequent words by word class? Noun, adjective, verb, adverb, intensifiers, negations, pronouns…

INTERESTING NLTK FUNCTIONS

* Tokenize
* FreqDist

ConditionalFreqDist – The ConditionalFreqDist class and ConditionalProbDistI interface are used to encode conditional distributions. if X and Y are two jointly distributed random variables, the conditional distribution of Y given X is the probability distribution of Y when X is known to be a certain value. For instance, we might want to know the probability that a person prefers a certain sport GIVEN that they are male. We might want to know the probability that the word “cat” is generated given that the word “the” is the word that has just occurred in a sentence.

Collocations - *Finding collocations requires first calculating the frequencies of words and their appearance in the context of other words. Often the collection of words will then requiring filtering to only retain useful content terms. Each ngram of words may then be scored according to some association measure, in order to determine the relative likelihood of each ngram being a collocation.*

* + The BigramCollocationFinder and TrigramCollocationFinder classes provide these functionalities, dependent on being provided a function which scores a ngram given appropriate frequency counts. A number of standard association measures are provided in bigram\_measures and trigram\_measures.

**TO DO 11 August 2021**

1. Gauge most common meaningful words by subreddit (Antiwork, Productivity, Psychic, Skeptic)
   1. Find nltk package that recognises POS?
2. Sort most common meaningful words by syntactic class (noun, verb, adverb, adjective, pronoun)
3. Sort by random sample of comments vs top-rated comments/posts (judging most reinforced language)

|  |
| --- |
| POS tag list in NLTK:  CC coordinating conjunction  CD cardinal digit  DT determiner  EX existential there (like: "there is" ... think of it like "there exists")  FW foreign word  IN preposition/subordinating conjunction  JJ adjective 'big'  JJR adjective, comparative 'bigger'  JJS adjective, superlative 'biggest'  LS list marker 1)  MD modal could, will  NN noun, singular 'desk'  NNS noun plural 'desks'  NNP proper noun, singular 'Harrison'  NNPS proper noun, plural 'Americans'  PDT predeterminer 'all the kids'  POS possessive ending parent\'s  PRP personal pronoun I, he, she  PRP$ possessive pronoun my, his, hers  RB adverb very, silently,  RBR adverb, comparative better  RBS adverb, superlative best  RP particle give up  TO to go 'to' the store.  UH interjection errrrrrrrm  VB verb, base form take  VBD verb, past tense took  VBG verb, gerund/present participle taking  VBN verb, past participle taken  VBP verb, sing. present, non-3d take  VBZ verb, 3rd person sing. present takes  WDT wh-determiner which  WP wh-pronoun who, what  WP$ possessive wh-pronoun whose  WRB wh-abverb where, when |

def process\_content():

try:

for i in tokenized[:5]:

words = nltk.word\_tokenize(i)

tagged = nltk.pos\_tag(words)

print(tagged)

except Exception as e:

print(str(e))

process\_content()

Chart, line chart

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generated

What can I see from the graphs above?

1. /r/antiwork analysis
2. /r/psychic analysis
3. /r/skeptic analysis
4. /r/productivity analysis

**Monday 16 August 2021**

* To do:
  + Get comments (n=50) from a sample of 50 submissions per subreddit
  + get TOP comments (n=10) from a sample of 50 submissions per subreddit.

**Monday 11 October 2021**

* To do:
  + Look at word frequency analysis tables – any words need to be filtered out? E.g. ‘Discord’, all words generated by comment bots on each post of a subreddit
  + Filter out words generated by bots, to make sure all word data are from comments written by humans

**Tuesday 9 November 2021**

* Feeling set on a certain idea: generating bots for different subreddits based on n-grams
  + Next step: find a way to initialise counting n-grams (bigrams, trigrams) for the corpus data of the 4 subreddits

**Sunday 2 January 2022**

* Re-started this project
* Set up proper repository in GitHub
* Decided to re-frame my goals into much easier tasks:
  + Find top 20 nouns from all 4 subreddits
  + Find top 20 verbs from all 4 subreddits
  + Find top 20 adjectives from all 4 subreddits
  + Collate all this data into comparison tables and perhaps write up some qualitative insight into this