Entity-Level Sentiment & Cross-Communal Comparison

An LHS 712 Final Project Report by McCoy Doherty & Nicolas Ortega

Introduction

While we have both been exposed to a wide variety of interesting applications of modern computing in the course of our academic year in the informatics-masters program, among the most charming were the ideas of entity-recognition systems, covered in both this course and our prior Information Retrieval course, and the notion of “entity-level sentiment,” the idea of identifying sentiment in linguistic expression towards a given article, which we were first exposed to in a Medium article that broadly touched on the topic of sentiment analysis or opinion mining and how applicable it can be to wide variety of domains. The article mentions the concept of entity-level sentiment at a surface level, and we ventured to find a commercial API that could help us with our objective.

We finally settled with Google Cloud’s Natural Language API, which does require authentication and billing details. Pricing is organized by “units”. For our purpose, the first 5,000 were free and the remaining ~20,000 cost about $40 total. However, the value the API provides in terms of capturing granular entity sentiment scores, at least to the best of our research, is best-in-class in terms of usability and quality of results.

When debating sources for data, we started to realize the degree to which the social-sharing platform Reddit concentrated a massive variety of diverse, but uniquely-thematically-focused social web communities across topics that could relate to various aspects of health and wellness. We decided to build out an easily-adaptive tool that could be utilized to assist in exploration of differential entity-sentiment comparison between topically similar, adjacent, or potentially disjoint Reddit communities or “subreddits" and conduct early tests with a mix of similar-though-disjoint communities.

Clustering Communities

After reflecting on reports of increased recreational substance usage of non-prescribed substances in light of the Covid19 epidemic, as well as relapses of formerly-usage-burdened individuals, we decided to compare sentiments specifically across and within three pools or categorical clusters of subreddits in the table below, listing each sampled subreddit community, as well as their subscriber count at the time of sampling.

Health Condition Subreddits				
	r/anxiety	445k	r/depression	747k
	r/bipolar	132k	r/eatingdisorders	51.7k
Drug Moderation & Cessation Subreddits				
	r/leaves (Cannabis Cessation)	117k	r/petioles (Cannabis Moderation)	78.2k
	r/addiction (General Substance)	53k	r/redditorsInRecovery (General Substance)	41.8k
Drug Patronage Subreddits				
	r/cocaine	95.8k	r/DMT	204k
	r/drugs	765k	r/heroin	26.9k
	r/ketamine	50.6k	r/LSD	151k
	r/MDMA	151k	r/meth	42k
	r/microdosing	151k	r/opiates	119k
	r/psychonaut	339k	r/stims	85k
	r/trees (Cannabis Enthusiasts)	1.7M		

The Complete Pipeline: The Four Phases

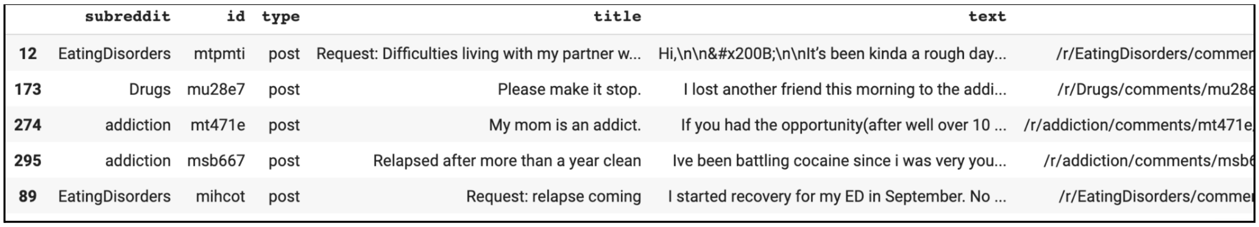
Step 1 :: Getting Data From Reddit

The first component of our project centers around the acquisition of data from Reddit. The platform affords users and researchers complimentary access to their data, but to achieve compliance with Terms of Service/Usage, said data must be acquired through their official API, which necessitates a Reddit account to acquire the requisite API credentials. After obtaining credentials, we opted to embrace an popular existing Python-compatible API wrapper, PRAW, to facilitate the gathering of our data.

Gathering data is primarily managed by the function:

```
        getter(PRAWRedditConnector, subreddits=[], n_posts=10)
```

This function takes in a list of subreddits, declares a Dataframe (object of the Pandas library), and fires API calls to catch the top n_posts from each reddit specified, as well as (generally) their comments, storing each post or comment as an entry, retaining the title, textual body/main content, a unique object identifier, an indication of whether or not a record represents a comment or original post, and a perma-link usable to inspect/view the resource in-browser. The PRAW library wrapper will return top N posts through a sorted scope that replicates those of the native web-platform's user interface, most typically sorted by “new”/”top”/”hot.”



	subreddit	id	type	title	text	
12	EatingDisorders	mtpmti	post	Request: Difficulties living with my partner w...	Hi,\n\n&#x200B;\n\nIt’s been kinda a rough day...	/r/EatingDisorders/commen
173	Drugs	mu28e7	post	Please make it stop.	I lost another friend this morning to the addi...	/r/Drugs/comments/mu28e
274	addiction	mt471e	post	My mom is an addict.	If you had the opportunity(after well over 10 ...	/r/addiction/comments/mt471e
295	addiction	msb667	post	Relapsed after more than a year clean	Ive been battling cocaine since i was very you...	/r/addiction/comments/msb6
89	EatingDisorders	mihcot	post	Request: relapse coming	I started recovery for my ED in September. No ...	/r/EatingDisorders/commen

Fig1: Example of harvested data from multiple sources, “End of Step 1”

Caveats & Oddities:

While the existing code for acquiring Reddit data is quite well-functioning, there are a few ways it can concretely be improved, a theoretical aspect that needs further investigation, and a potential bias to be aware of. The certainly-doable improvements involve the gathering of sub-comments, those which are responses-to-responses of the original post, as well as the reconfiguring an existing try-except block that manages “See More” instances. See-More instances are datatype objects that occur in the Reddit PRAW library to represent the “See More” button that would appear in a user’s view when so many posts or sub-comments exist that the website chooses to abridge/truncate the comment-tree to avoid burying initial comments in cases of excessive sub-comments. Effectively, this means that, over a certain threshold, not *all* comments of a post are being gathered; the documentation provides a suggested remedy to be adopted in the future, namely their convenience method to perform a breadth-first traversal of the comment forest to find comments at all existing levels, effectively retrieving the replies to a single comment thread of arbitrary depth.

The caveats of the Reddit data acquisition process that are perhaps more robustly difficult to manage are the inherent bias that could arguably occur from how the library handles processing, normatively by acquiring “new” or “hot” (currently popular) posts, the latter of which could create bias if sampling only the most popular posts, which could center around specific entities and tones towards said entities. Mixing in “new” posts can help counteract this potential for bias. While “new” content could bias towards current events, and sampling back enough in time to target specific dates is still an object of question, date-specific collection is also a tentative improvement in the plans.

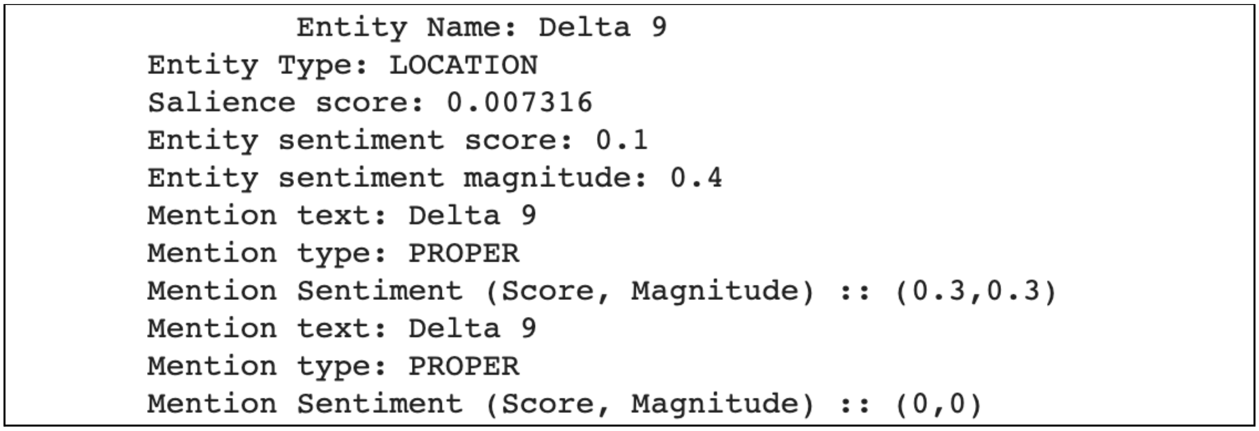
Step 2 :: From Document to Sentiment

The dataframe produced at the end of the first primary component is passed on into our second primary function:

```
    sample_analyze_entity_sentiment(document)
```

This function takes a given string of body-text from a post or comment and passes it along to the Google Cloud NLP Entity-Sentiment Analysis API, which returns a custom response datatype that includes a list of entity-recognized objects. For each entity identified, the object will generally attempt to include the entity’s name, salience (importance to focus of document), sentiment, and subdivided sentiment at the per-mention level, each mention having a respective instance-sentiment and instance-magnitude.

Our current structure establishes a dictionary-based JSON cache that, for each row (post/comment) of the provided dataframe, checks if the cache contains a response object “keyed” under the row/item’s unique identifier and, if not listed, runs the API call to Google’s cloud and saves the keyed response object to the cache. A more reader-friendly example of output for features of an extracted entity can be found below in Figure 2; it is worth noting that sentiment is measured between -1 and +1 representing negative/positive sentiment, and there’s no clear cutoff as to what ‘strong’ or ‘weak’ positivity/negativity scores are, so the threshold is domain dependent and somewhat arbitrary. The API returns the sentiment score, which refers to either the positive/negative prevalent emotion present within the document, and magnitude reflects how much emotional content is present within the whole document. Unlike score, magnitude is not normalized and is proportional to the length of the document.



```
                Entity Name: Delta 9
        Entity Type: LOCATION
        Salience score: 0.007316
        Entity sentiment score: 0.1
        Entity sentiment magnitude: 0.4
        Mention text: Delta 9
        Mention type: PROPER
        Mention Sentiment (Score, Magnitude) :: (0.3,0.3)
        Mention text: Delta 9
        Mention type: PROPER
        Mention Sentiment (Score, Magnitude) :: (0,0)
```

Fig2: Example of Entity Extracted From Post

Caveates & Oddities:

It is worth noting that, albeit chosen out of personal curiosity, we came to realize that Google's cloud API services, as commercial services, are fundamentally closed-source in nature, meaning much of the back-end // underlying methodology behind observed performance and outcomes have to be considered "black-box" and raise questions about explainability while destabilizing reproducibility. Theoretically, this component of the pipeline may potentially warrant replacement with alternative frameworks for extracting entities and sentiments.

It is also worth noting that we discovered instances where the API was not returning any entities from the API call to the sentiment analyzer; this is a point where incorporating original document text and resource URL in our previous data structure became very beneficial. We were able to identify multiple causal cases for a no-entity return:

- Original Posts that were Photo/Media-Only
- Posts/Comments with No True Entities
    - i.e. "nice" // "wow" // "lol"
- Posts/Comments with Antecedents
    - Uses articles like [it, this, that] to refer to image/video/media content or, alternatively, a subject established in a prior post/comment
- Malformed Grammar // Typo Incidents (common in social media contexts)
- General failure of API to catch existing entities

Given more time, we would have liked to have also conducted a more formal annotation study of the results observed to better understand and attempt to quantitatively benchmark the performance of the Google Cloud NLP API at the individual tasks of entity extraction and sentiment analysis.

Step 3 :: Entity Reduction // Name-Variant Grouping

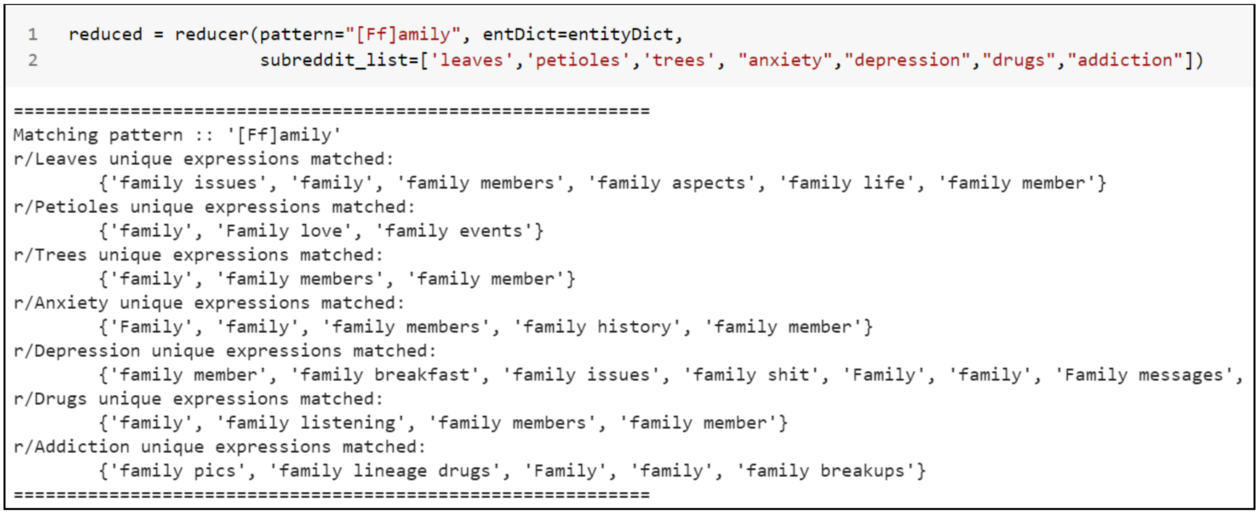
Having acquired the Reddit data and generated sentiment scores through the Google entity-sentiment analysis API, the next component of this project (located in the Part 2 ipynb notebook) focuses on handling a particular issue consequent of the API's structure: variation.

Whereas entity identification and extraction mechanisms such as those we have had prior experience with creating, most commonly with the SpaCy library, have better capacity to, in training custom entities, extract entities by genre of label, when working with our chosen Google API, there wasn't such an easy way to cluster these automatically-generated, generically-categorized (mostly always of type "other") labels.

This matters because the API could differentiate entities simply by capturing differentiation in capitalization, hyphenation between terms, mistakes of spelling/grammar, and the potentially-inappropriate capture of a prepending or trailing term. While any of these differentiations could indicate substantial differentiation of a term, this is situationally-relative and may need to be a judgement call by a researcher. Rather than establish an inflexible method for contracting entity variants into groups, our initial decision has been to focus this process into a function called:

```
reducer(pattern, entDict, subreddit_list, doesNotContain)
```

which takes in, along with a list of target subreddits and a sorted-by-reddit-then-by-entities dictionary, a regular-expression parameter called "pattern" which allows for specification of what sort of expressions should be factored into the aggregated summary of sentiment. This function reports each instance of a matched expression to the specified pattern per subreddit of specified interest and reports them to output so potentially misaligned matches can be dealt with.



```python
reduced = reducer(pattern="[Ff]amily", entDict=entityDict,
                  subreddit_list=['leaves','petioles','trees', "anxiety","depression","drugs","addiction"])
```

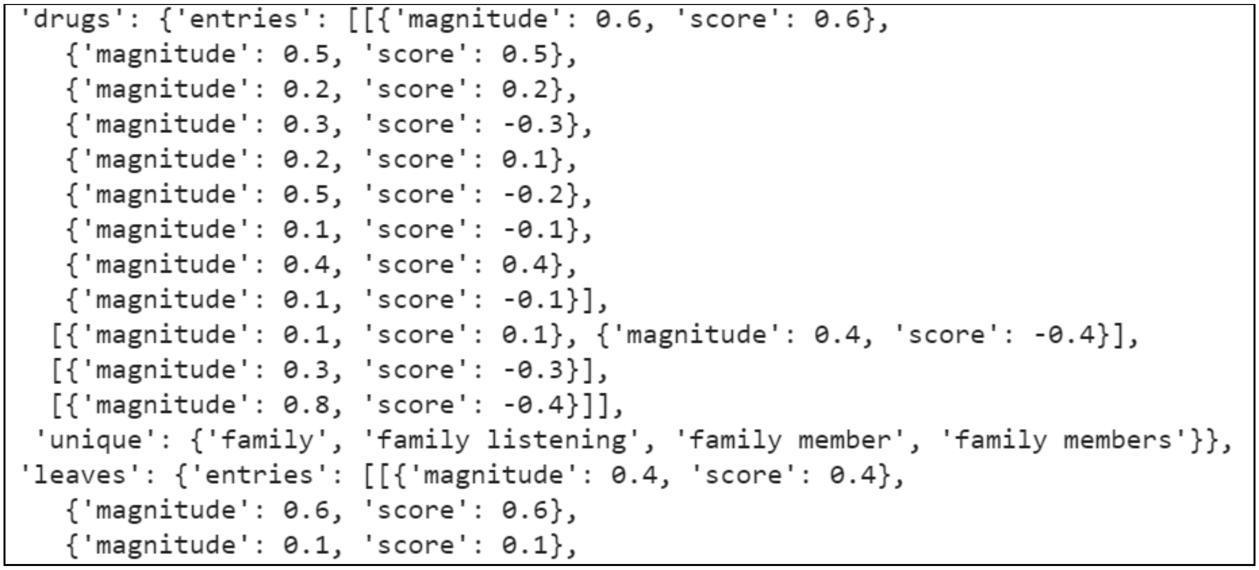
```
============================================================
Matching pattern :: '[Ff]amily'
r/Leaves unique expressions matched:
        {'family issues', 'family', 'family members', 'family aspects', 'family life', 'family member'}
r/Petioles unique expressions matched:
        {'family', 'Family love', 'family events'}
r/Trees unique expressions matched:
        {'family', 'family members', 'family member'}
r/Anxiety unique expressions matched:
        {'Family', 'family', 'family members', 'family history', 'family member'}
r/Depression unique expressions matched:
        {'family member', 'family breakfast', 'family issues', 'family shit', 'Family', 'family', 'Family messages',
r/Drugs unique expressions matched:
        {'family', 'family listening', 'family members', 'family member'}
r/Addiction unique expressions matched:
        {'family pics', 'family lineage drugs', 'Family', 'family', 'family breakups'}
============================================================
```

Fig3: Example of Reducer() User Feedback

In Figure 3 above we illustrate a practical example: consider if a researcher would like to identify tone towards aspects of the notion of “family” reflected in speech across an assortment of subreddits; we can see that for the pattern “[Ff]amily” to capture the term with and without initial capitalization, each subreddit is finding a different set of entity terms matching or including that term. The function reducer, in this case, is also returning a dictionary with:

- keys being the names of each subreddit specified
- [subreddit] [“unique”] being a set-object containing each unique expression caught by the user-provided pattern and collected for aggregation
- [subreddit] [“entries”] being a list of entity-level sentiments extracted from posts sampled in that given subreddit

In the case that an undesirable entity is incorporated into the collection, such as if one wanted “family doctor” removed from the aggregation of some results, this could either be excluded by adding a negative lookahead specification to the regular-expression pattern passed into the function or, alternatively, an exclusionary substring filter could be added to the list-parameter doesNotContain.



```
'drugs': {'entries': [[{'magnitude': 0.6, 'score': 0.6},
   {'magnitude': 0.5, 'score': 0.5},
   {'magnitude': 0.2, 'score': 0.2},
   {'magnitude': 0.3, 'score': -0.3},
   {'magnitude': 0.2, 'score': 0.1},
   {'magnitude': 0.5, 'score': -0.2},
   {'magnitude': 0.1, 'score': -0.1},
   {'magnitude': 0.4, 'score': 0.4},
   {'magnitude': 0.1, 'score': -0.1}],
  [{'magnitude': 0.1, 'score': 0.1}, {'magnitude': 0.4, 'score': -0.4}],
  [{'magnitude': 0.3, 'score': -0.3}],
  [{'magnitude': 0.8, 'score': -0.4}]],
 'unique': {'family', 'family listening', 'family member', 'family members'}},
'leaves': {'entries': [[{'magnitude': 0.4, 'score': 0.4},
   {'magnitude': 0.6, 'score': 0.6},
   {'magnitude': 0.1, 'score': 0.1},
```

Fig4: Example of Reducer() Output Data Structure -- Unique Expressions and Pooled Values of r/Drugs, as well as r/Leaves (truncated)

Future Development:

In the future we would like to experiment with adding additional parameters to allow other options for tidying-up the entities by name, such as allowing support for stemmers to collapse terms together, as well as a parameterized offering to let users elect to include terms within a specified threshold of Levenshtein's Edit Distance into the aggregation averaging step.

Step 4 :: Sentiment Aggregative Average

Once the data has been coerced together such that each entity-sentiment-within-post is sorted together by subreddit, we pass off the resulting dictionary to our function:

```
        AverageSentiments(reduced_dict)
```

to iterate through the dictionary and get the average of sentiment scores. It is worth noting that the function has a call to "handleScore(instance)" function, which is something we implemented to effectively reimpose lost zero-values. The API articulates sentiment in two halves, magnitude, which represents intensity, and score, which represents the positivity/negativity component of sentiment and our primary focus at this time. At the time of implementing this component of the project we had begun to realize exceptions being thrown by instances where magnitude was apparent, but score was not listed.

This is a situation where much time was lost trying to peek into the more computational / structural side of the “black box,” as we noted that, in the original API response objects for such documents, a document.sentiment.score was accessible, but upon access returned the value 0. We then realized that, in converting the API response object to a “protobuf” to then use another Google-custom method to convert that object into a more conventionally-serialized dictionary object so we could write to cache, these zeros were getting lost in the process. This is why we had to effectively check for sentiments with only their magnitudes accessible to “re-impose” the zeros and ensure they factored properly into the averaging.

The Corpus Composition

At the time of writing, our corpus includes 24,174 records in total, each record representing a post or comment from an online Reddit community, including both its original document text and its sentiment analysis stored from the API. We chose twenty-one subreddits to sample data from, ensuring at least 250 original posts per subreddit. Our full testing cache is described in the table below:

Subreddit	Original Posts	Comments	Entity Mentions	Entity Mentions With Sentiment	
r/anxiety	747	1,162	26,093	17,233	66.04%
r/depression	748	1,363	27,247	19,333	70.96%
r/bipolar	497	1,529	22,308	13,868	62.17%
r/eatingdisorders	512	333	20,635	12,895	62.49%
r/addiction	504	407	16,487	10,843	65.76%
r/leaves	557	393	15,385	10,793	70.15%
r/petioles	507	792	16,468	10,762	65.35%
r/redditsinrecovery	250	1,099	17,879	11,508	64.37%
r/drugs	252	1,207	12,029	7,367	61.24%
r/stims	250	525	5,977	3,734	62.47%
r/psychonaut	250	805	16,122	9,443	58.57%

r/microdosing	250	593	9,037	4,853	53.70%
r/lsd	251	846	6,380	3,811	59.73%
r/dmt	250	502	5,696	3,132	55.02%
r/opiates	250	617	11,269	6,891	61.15%
r/cocaine	250	294	2,205	1,229	55.74%
r/mdma	250	516	6,663	3,974	59.64%
r/ketamine	250	353	3,397	1,943	57.19%
r/heroin	251	524	4,811	2,873	59.72%
r/meth	250	548	5,279	2,840	53.80%
r/trees	260	2,180	8,763	4,814	54.94%

Example Analysis Cases & Outcomes

Average Entity-Level Sentiment By Subreddit:

To better establish the relativity of entity-level sentiment in these diverse and different Reddit topical communities, we thought it important to also report what the average entity-level sentiment is for each subreddit community. We can see in the table below that, overall, with the exception of the four underlined averaged sentiments, average sentiment scores are always within 0.09 of neutral score, which is based at 0.0. These exceptions are, fittingly enough, negative bias in self-explanatory r/depression, and positive in the cases of r/LSD, r/Petioles (which focuses on moderating cannabis usage), and r/Microdosing, possibly due to emphasis on positive lifestyle changes or realizations from members of those communities.

Subreddit	Avg. Sentiment	Entity Instances	**Subreddit**	Avg. Sentiment	Entity Instances
r/anxiety	-0.035043	17,233	r/drugs	+0.049192	7,367
r/bipolar	+0.015128	13,868	r/stims	+0.040279	3,734
r/depression	-0.110397	19,333	r/lsd	+0.098478	3,811
r/eating	-0.049903	12,895	r/dmt	+0.081226	3,132

disorders					
r/addiction	-0.032483	10,843	r/opiates	-0.014541	6,891
r/leaves	+0.04527	10,793	r/cocaine	+0.053377	1,229
r/petioles	+0.096673	10,762	r/mdma	+0.032486	3,974
r/redditors in recovery	+0.040945	11,508	r/ketamine	+0.080134	1,943
r/psychonaut	+0.08349	9,443	r/heroin	+0.010755	2,873
r/microdosing	+0.121822	4,853	r/meth	+0.000282	2,840
r/trees	+0.07983	4,814			

Case 1: Sentiment Towards Sleep

One realm of focus we figured would have a reasonable amount of cross-community representation within our corpus of sampled data is the topic of sleep; by aggregating entities matching the regular-expression pattern of “[Ss]leep” we were able to get the entity-level sentiment pooled for sleep and closely-adjacent tangential mentions of sleep, including representations such as “sleep apnea” or “sleeping difficulties.” These averaged sentiment and the instance-count aggregated in the computations, per subreddit, are captured in the table below. We can see that, of reported subreddits, tone towards sleep is, in all mental-health focused subreddits except for r/bipolar, negative. We can also see that, for the addiction-and-moderation-themed subreddits, sentiment is generally negative-but-not-strong, falling within 0.10 of neutrality. The strongest positive sentiment is reflected in r/trees, which is a subreddit community for cannabis enthusiasts, allowing us to speculate this is attributed to the sleep-inducing effects of the substance.

Subreddit	Average Sentiment	Entity Instances
r/anxiety	-0.171233	72
r/bipolar	+0.052239	67
r/depression	-0.146809	47

r/eatingdisorders	-0.171429	14
r/addiction	-0.08	25
r/leaves	-0.031481	54
r/petioles	+0.091525	59
r/redditorsinrecovery	-0.068571	35
r/drugs	+0.151852	27
r/stims	-0.061538	13
r/trees	+0.2375	16
r/meth	+0.005556	18

Case 2: Sentiment Towards Therapy & Therapists

A natural direction we wanted to explore in this project was investigating the sentiment scores towards therapy and therapists in different subreddits. Our original analysis included 'doctor' entities, but the entity is not as frequent as we would have desired. We believe, however, that therapy is a good proxy to investigate how people are reaching out and getting help, and more importantly, how they perceive this. As performed in the previous example, below are the average scores for each subreddit regarding therapy. We can see a slightly negative sentiment for bipolar and depression, and slightly positive for both anxiety and eating disorders. Anxiety and depression are often coupled, so seeing a slight inverse relationship between the two is somewhat surprising. Both subreddits contained similar therapy-based entity variations, but depression views therapy with a more negative sentiment.

Depression and bipolar, however, are the only subreddits with a negative average sentiment for therapists. Addiction cessation communities all show positive scores, from slight to quite high at 0.26 (r/redditorsinrecovery). This result is consistent with the subreddit's mission as it presents itself as a positive community dedicated to providing resources and support to those in recovery. Additionally, drug-centric communities like microdosing, psychonaut, and ketamine also see therapy in a positive light. This is presumably because of the increased use of psychoactive substances such as ketamine in new depression treatments. We hope that the trend for an increased sentiment score for therapy will continue across communities.

Subreddit	Average Sentiment	Entity Instances
r/anxiety	+0.020472	127
r/bipolar	-0.005263	57
r/depression	-0.095385	130
r/eatingdisorders	+0.080808	99
r/addiction	+0.030435	23
r/leaves	+0.031579	19
r/petioles	+0.127586	29
r/redditorsinrecovery	+0.261905	42
r/microdosing	+0.212	25
r/psychonaut	+0.127778	18
r/ketamine	+0.116667	6

Case 3: Sentiment Towards Cannabis in Enthusiast, Moderation, & Cessation Groups

When building our initial analysis corpus, we noticed an interesting triad in the sample of subreddits that we had selected, in that r/trees represents a pro-cannabis community of enthusiasts, r/leaves represents a community of individuals who intend permanent cannabis cessation, and r/petioles seems to assert itself as a bit of a cultural mid-point, for individuals seeking to moderate their relationship with the substance but not strike for permanent cessation. We were particularly curious of whether or not mentions of cannabis would see stratification of sentiment across these subreddits, as well as what sentiment would look like in more mental-health-oriented groups.

As reflected in the table below, average tone in cannabis discussions for these subreddits was, generally, neither steeply positive nor steeply negative, however, in a relative sense, it seems that r/depression and r/eatingdisorders represented the strongest sentiment average, both leaning substantially into the negative tone. We can speculate that, inn the case of r/eatingdisorders, this could be linked to cannabis' role in inducing hyperphagia in users which, in those with eating disorders, could trigger binge-eating behaviors and consequent regrets.

It is also worth noting that, albeit that those two subreddits had the strongest averaged sentiments, they also represented relatively fewer data points of relevant mentions to cannabis in their overall contents than drug-and-addiction-cluster subreddits, logically. We believe that these averaged scores could change substantially if sample size for the respective subreddits is increased substantially enough to catch more (albethey rare) mentions of cannabis topics.

Regarding our original curious intention of comparing averages across the trees-petioles-leaves spectrum, we can see that the sentiment of r/leaves is net-negative, which makes sense given the generally lamentful / repentful nature community members feel towards cannabis usage, evident in their group focus being cessation. We can also see that r/trees, the enthusiast group, has a positive sentiment, similar in intensity to that of r/leaves' negative sentiment, and that r/petioles, the moderation-seeking group, even has a more positive averaged sentiment than that represented in r/trees.

Our current speculation as to the nature of r/petioles presenting with the stronger positive sentiment towards cannabis is that there are a few potential biases that could be occurring here in the nature of these communities and/or specificity of their language patterns:

1) Members of r/petioles specifically are cannabis users with such fond proclivity towards the substance that they felt the need to moderate in the first place
2) Members of r/petioles may be more socially celebrant of their perceived positive lifestyle changes and moderation of cannabis consumption
3) Members of the r/trees group are enthusiasts who may concentrate more conversation and potential sentiment into specific terminology, rather than umbrella terms used to aggregate the samples for sentiment
    a) Possibly less about "weed" or "pot" or "cannabis
    b) Possibly more talk about "joints" or "bong rips" / other tangential entities

Subreddit	Avg.Sentiment	Entity Instances	Subreddit	Avg.Sentiment	Entity Instances
r/anxiety	-0.054545	11	r/addiction	-0.084615	65
r/bipolar	-0.045455	11	r/leaves	-0.032603	411
r/depression	-0.138462	13	r/petioles	+0.046469	439
r/eating disorders	-0.133333	3	r/redditors in recovery	-0.061538	65

r/trees	+0.024627	134			

Case 4: Sentiment Towards Feelings

As a very broad, catch-all term, we wanted to see both the variety of terms captured when referring to feeling as an entity as well as the sentiment score for the specific entity within the subreddit. We observed that there was a variety of captured entities that can be categorized into physical and emotional feelings. These, in a sense, could be thought of as symptoms, but not looking for a strictly medical context. As seen in the table below, most subreddits (except for petioles, trees, drugs) exhibit a negative sentiment towards feelings. Depression exhibits the lowest sentiment score towards feelings, which makes sense since often text in the subreddit reflects the negative, depressed feelings of the users.

As a corollary, we see petioles and redditorsinrecovery exhibit positive sentiment scores for feelings, which further validates the performance of the Natural Language API. In the drugs subreddits, two interesting results include negative sentiment scores for stims and mdma. This is interesting, since mdma and stimulants in general are often associated with overwhelming positive feelings. However, we suspect the results below are more accurate since the users post after their experiences, often during a lull the day after. In this sense, positive feelings might be the first association while on the substance, but the more true, clear feelings are reflected after the experience. We can see the positive scores in both drugs and trees, meaning that the distinction in terms of feelings for every drug can be further refined to study how specific drugs are discussed and perceived in online communities. A study delving deeper into drug subreddits by performing opinion mining with more details in terms of categorizing the type of feeling that is exhibited (i.e. physical, emotional) could be performed in order to understand how granularly we can categorize our target entities, possibly using machine learning methods to train a custom entity recognizer. Feelings is a broad topic, but we think that a feature to track feelings sentiment in a particular subreddit could be a substantial benefit to moderators inside the Reddit community, and in turn for the users and members of the community.

Subreddit	Average Sentiment	Entity Instances
r/anxiety	-0.094737	133
r/bipolar	-0.118421	38
r/depression	-0.164655	116
r/eatingdisorders	-0.079167	72
r/addiction	-0.046667	45
r/leaves	-0.045588	68
r/petioles	+0.011429	70
r/redditorsinrecovery	+0.097222	36
r/drugs	+0.075676	37
r/stims	-0.1875	8
r/trees	+0.2	9
r/mdma	-0.042857	14

Continued Expansion & Future Direction

As established throughout this paper, we've developed a four-component working pipeline to support the collection and sentiment-analysis of Reddit data, however we find ourselves with incremental improvements to be made along the way. In recapitulation, we aspire to add more parameters to the guiding functions of each pipeline component to support for

- Collection of comments further down the tree structure
- Collection of comments hidden behind Reddit's "See More" buttons
- Collection of data from more specific time periods
- Alternative approaches for wrangling variations in extracted entities for aggregation
    - Offer parameter-thresholded Levenshtein Edit Distance collapse of variants

It goes without saying that continued expansion of the overall dataset by continually collecting new posts, comments, and data from new online reddit communities of interest, naturally represents expanded worth of the project.

Last but not least, due to the modular nature of our design, we have two larger endeavors in mind for the continued development of the project, not so thoroughly discussed in the paper thus-far.

Major Update #1: Swappable Entity Extractor & Sentiment Analyzer

Due to the blackbox nature of Google's commercial APIs for entity and sentiment processing, we believe that it is important to, at the very least, capitalize on the more modular four-component design of the project to allow users to "socket-in" alternatives to the for-pay Google API used in this paper. Due to our authors' mutual experience working with the library SpaCy for entity extraction and sentiment analysis, we believe it would be more beneficial to support usage of SpaCy's free, open-source entity and sentiment analysis affordances.

SpaCy affords not only robust pre-trained models for extracting entities in English contexts, foreign-language contexts, and multi-linguistic contexts, but it also affords users the ability to, with sufficient annotated data, train custom entity recognizers that additionally tag specific classifications of entities. Impressed with SpaCy's performance in our previous work in extraction of certain custom categories of entities from highly-variable job application postings, we believe it would make an ideal interchangeable option for those who may feel uncomfortable with the low-control, low-explainability, cost-required Google NLP API.

We also think it would be worth taking the time to annotate and compare performance of the standard pre-trained SpaCy entity-recognition model to the performance exhibited by the Google NLP API, which, having everything retained in cache, it would be relatively easy to compare on a 1-to-1 basis.

Major Update #2: Web Application / A Non-Programmer's Interface

While we personally both come from academic backgrounds heavily focused around computation and the art of computer programming, throughout this course we have been regularly presented with both programming-oriented approaches to NLP tasks and potential non-programming alternatives for these tasks. Between this reminder that there are often researchers who may wish to perform natural-language tasks like those performed in our project and our personal experiences also learning about modern web development in our other studies this year, we have realized that a natural next step for this project could be creating a web application to allow users to utilize the tooling we've written with minimal or no programming required.

Rebuilding part of our codebase to facilitate usage as a web application seems doable so long as users go through the non-programming steps of acquiring the requisite API keys for Reddit and for Google Cloud (unless using an alternative such as those mentioned in the prior section). Building this application would help decrease a technical skill barrier to accessing this type of analysis, to the benefit of researchers and simply-curious-individuals alike.

Theoretically, with the right end-user license agreements / disclaimers and database infrastructure, such a web application could also save researchers money by saving analyzed documents to a large database, similar to how we cache our results, allowing for future researchers to used cached data instead of firing the API call to analyze a document that already has a stored analysis. Establishing a crowd-sourced "runoff database" like this could also allow users who are unable to acquire API keys or pay for using the Google Cloud NLP API to experiment with entity-level sentiment at no cost at all.

References

Sentiment Analysis: A Fascinating Problem by Nitin Kumar Kain
https://medium.com/@AI_with_Kain/sentiment-analysis-a-fascinating-problem-cf7f9773147

PRAW: The Python Reddit API Wrapper
Documentation Home :: https://praw.readthedocs.io/en/latest/
On Capturing All Comments :: https://praw.readthedocs.io/en/latest/tutorials/comments.html

Google Cloud :: Natural Language Processing
Documentation Home :: https://cloud.google.com/natural-language
Pricing Schema :: https://cloud.google.com/natural-language/pricing

Interpretation of Sentiment Analysis Values ::
https://cloud.google.com/natural-language/docs/basics#interpreting_sentiment_analysis_values

Team Contribution Summary

Component	Primary Participants
Project Conceptualization	Mutually // McCoy & Nico
Searching for Target Reddit Contenders (Searching Reddit, finding interesting contenders, reporting nature of community and subscriber count)	Mutually // McCoy & Nico
Setting up Reddit ● API Credentials ● Code to make API calls	McCoy
Setting up Google API ● API Credentials ● Billing ● Code to make API calls	Nico
Setting up Caching && Integrating to API Call System **	McCoy
** mini-rant :: This was much more difficult and time consuming than imaginable because of Google using custom datatypes, dir() and __dict__() issues with custom datatype serialization, and documentation not helping much with this probably so people'd make more API calls not realizing two obscure methods could be conjoined to make this doable… [This took basically a full day to figure out the small-looking nuances to make it work]	
Setting up Cache Comparison ● Reorganize Data :: Pool Entities Within Subreddit ● Write Averaging Logic	Mutually // McCoy & Nico
Investigate missing values ++ “defaulting” ++ nuances of serializing Google’s weird data structure to a normal one	Mutually // McCoy & Nico

Extracted-Entity Reduction	About 75-25 Nico-McCoy
Brainstorming “Experiments” (Terms for comparative lookup)	Mutually // McCoy & Nico
Code Refactoring ● Focus on “cleaning up” our implementations ● Focus on wrapping things up into functions ○ Adding meaningful parameters for control	About 75-25 McCoy-Nico
Report Writeup	Mutually // McCoy & Nico