Sentiment Dynamics on Reddit: A Deep Dive into Subgroup Perceptions of the Israel-Palestine Conflict using Advanced NLP Techniques

Greg Forkutza
Student ID: 400277514

24 October, 2023

Contents

1 Introduction **2**

2 Methods **2**

3 Results **4**

4 Project Timeline **5**

5 Supplementary Material **6**

References **13**

1 Introduction

The primary objective of this project aims to provide a unique intersection of manifold learning and natural language processing (NLP) techniques to discern the sentiment of various Reddit subgroups towards specific current world events. Reddit, a popular social media platform, hosts a diverse range of users, often segregated into subreddits that cater to specific interests or demographics. For this analysis, we will focus on the recent Israel vs. Palestine conflict, aiming to understand how different sub populations perceive the situation given the evolving nature of social media's role in shaping public opinion on global events. The data set is sourced from the Reddit API, which provides comprehensive details, including comments, upvotes/downvotes, author information, and timestamps.

Reddit, as a platform, has previously been analyzed for various sociological and informational trends. While certain studies focus on sentiment during global events, others analyze patterns in user behaviors or community dynamics. Our study will bridge the gap between broad sentiment analysis and niche subgroup evaluations.

We aim to develop an integrated pipeline that captures, processes, analyzes and visualizes thematic content and semantic stricture from varied subreddits. The end product is an application that will allow the us to compare and contrast how subgroups perceive information as it changes over time. Understanding these nuances is essential for effective public discourse; for communities to communicate effectively and collaboratively address issues, we must grasp the depths of their sentiments."

2 Methods

Approach: Two primary approaches will be utilized:

1. This approach leverages established methodologies in the realm of text analysis. `Word2vec` captures semantic relationships between words based on their contextual usage Alammar [2023] and `BERT` (Bidirectional Encoder Representation from Transformers) Bratt [2023], a transformer-based model pre trained on vast amounts of data.

2. This approach harnesses state-of-the-art Large Language Models (LLM) such as Open AI's GPT-3.5 Turbo offer a deep and nuanced understanding of the text.

The advantage of the LLM approach lies in recognizing subtle contexts, idiomatic expressions and the intricate play of natural language that gets missed by traditional tools. The disadvantage of this approach is the token based cost of GPT-3.5 Turbo. The advantage of `BERT` and `Word2Vec` is that they are cost efficient in terms of computational cost in terms of speed and money. Being unsupervised models, they eliminate the

need for labeled data. Furthermore they do not require prompt engineering to tune the model response as with LLM's.

Data Collection, Monitoring, Processing and Storage:

We will employ Docker to facilitate the setup and management of key services like Apache Kafka and Zookeeper. We will use the R packages `httr` and `jsonlite` to fetch and process data from the reddit API CRAN [2023a]. This data is transformed into a JSON format. Next we will integrate Kafka with R using the `rkafka` package Apache Software Foundation [2023]. This will enable our Reddit data collection script to feed the acquired data into a Kafka topic. This serves as a buffer and staging area for the flow of data. Kafdrop is a UI tool we will deploy for monitoring and debugging purposes. When the data enters our local machine, messages will be consumed from the Kafka Topic in R for pre-processing and cleaning using the `tidyr` and `dyplyr` packages. The processed data will be stored in a database using `RSQlite` for easy retrieval for downstream tasks, given the transient and rapidly changing nature of sentiment analysis CRAN [2023b].

Scope: Beyond basic sentiment scores there are many nuanced approaches we can take with regards to the analysis. Optional possibilities include (time dependent):

- **Focusing on one Sub population Dimension:** Delve deep into the nuances of sentiment between obvious contrasting subgroups, e.g, the "liberals vs conservative" spectrum.
- **Event Specificity:** Focus on a single event regarding the Isreal-Palestine conflict to allow for a more granular analysis of sentiment shifts.
- **Sentiment Dynamics:** Examples questions include; how quickly does sentiment change, is there a noticeable delay in reaction to news events?
- **Influence Analysis:** Which comments get the most traction and shape the narrative within a sub-group.
- **Topic Modelling First:** Identify main topics discussed in relation to event before sentiment analysis in order to contextualize it based on specific themes.
- **Bias Detection:** What are the inherent biases in discussions: myths, misinformation, misconception being propagated within subgroups.

Feature Engineering:

-Features will be derived from comment length, upvotes/downvotes ratio, author history, subreddit associations, temporal patterns etc… This is still be explored and depends on decisions made regarding scope.

Embedding and Analysis:

1. **Word2Vec Analysis:**
    - Create embedding of the cleaned Reddit comments. Explore word similarities and compute word vectors for sentiment analysis

2. **BERT:**
    - The `text` package in R provides a BERT interface to embed the Reddit comments. We can either train a classifier model for sentiment analysis on a labeled subset of the Reddit data or use pre-trained models. We can also use BERT embedding for topic modelling. This has the effect of potentially improving on predictions methods like `Word2vec` by capturing deeper semantic meaning.
3. **GPT-3.5 Turbo:**
    - Send processed comments to GPT 3.5 for sentiment evaluation to get sentiment scores. This will involve engineering the prompts to get the desired sentiment output. We will need to develop a scoring system to aggregate and quantify the sentiment scores.

Visualization:

- Given the high dimensional nature of the word embedding, we can employ Universal Manifold Approximation and Projection(UMAP) to reduce dimensionality and create visual representations of the sentiment and topic analysis McInnes, L, Healy, J [2023]. Such methods are indispensable when visualizing and interpreting high-dimensional data. We will construct a interactive shiny dashboard to showcase sentiment distributions, word embedding visualizations, word similarities and other interesting insights. Incorporate options for users to select subreddits, data ranges and particular topics to update the displayed analysis.

Interepretability, Validation, Evaluation and Comparison:

- For GPT-3.5 Turbo's sentiment analysis we can manually review a subset of comments and their sentiment scores to gauge accuracy. For Word2Vec's embedding, we can check if semantically similar words are close in the embedding space. Subset some very small part of the database, split into a training set and test set and manually label these. Evaluate the models using metrics like accuracy, F1-score, precision, recall and ROC_AUC. Create side by side visual comparisons of the visualizations. Examine overfitting.

Documentation

Since we are focusing on the future reproducability and scalability of this project we need to keep the processes, decisions and code base well documented using GIT for version control.

3 Results

Currently, I've acquired and cleaned comments from the Reddit API and am in the process of training the Word2vec model. The manual labeling process has proven challenging. I'm considerations using LLMs for this task. I'm enthusiastic about using GPT-3.5 Turbo.

4 Project Timeline

Have the proof of concept completed by Nov 10th. This means choosing a single scoping project and completing the visualizations and dashboard. Then work on analysis and prepare slides for the 20th of November. This leaves three weeks to write the paper and add additional scopes if time permits.

5 Supplementary Material

I have included an analysis of one comment thread from one comment post for display purposes.

```
# Define API credentials
client_id <- "q1vvgIe6E8n7rgfcovg7KQ"
client_secret <- "MFkw2VP8S76lRJ7nLA_D2Msp7RLcpA"

reddit_endpoints <- oauth_endpoint(
  authorize = "https://www.reddit.com/api/v1/authorize",
  access = "https://www.reddit.com/api/v1/access_token"
)



# Authenticate
my_app <- oauth_app("reddit", key = client_id, secret = client_secret)

reddit_token <- oauth2.0_token(
  reddit_endpoints,
  my_app,
  scope = c("read"),
  use_basic_auth = TRUE,
  config_init = user_agent("Mac:UMAP:v1.0 (by /u/Forkman3939)")
)

# Define the URL for the post's comments
url <- "https://oauth.reddit.com/r/trans/comments/151dseo"



# Fetch the comments using the GET request
response <- GET(url, add_headers(Authorization = paste("Bearer",
                                 reddit_token$credentials$access_token)))

#Extract the content from the response
content_response <- content(response)
```

```r
# Extract post info from the content
post_list <- content_response[[1]]

# Extract the comment info from the content
comments_list <- content_response[[2]]

# Define function to recursively extract features from nested list structure
extract_comments <- function(comments, parent_id = "ROOT") {
  results <- data.frame(
    ups = integer(),
    downs = integer(),
    comment_id = character(),
    author = character(),
    parent_id = character(),
    subreddit = character(),
    link_id = character(),
    score = integer(),
    is_submitter = logical(),
    body = character(),
    depth = integer(),
    controversiality = integer(),
    created_utc = integer(),
    stringsAsFactors = FALSE
  )

  for (i in 1:length(comments$data$children)) {
    comment <- comments$data$children[[i]]$data
    results <- rbind(results, data.frame(
      ups = comment$ups,
      downs = comment$downs,
      comment_id = comment$id,
      author = comment$author,
      parent_id = parent_id,
      subreddit = comment$subreddit_name_prefixed,
      link_id = comment$link_id,
      score = comment$score,
```

```
      is_submitter = comment$is_submitter,
      body = comment$body,
      depth = comment$depth,
      controversiality = comment$controversiality,
      created_utc = comment$created_utc,
      stringsAsFactors = FALSE
    ))

    # Recursively extract replies, but only if 'replies' is a list
    if (!is.null(comment$replies) && is.list(comment$replies)
        && !is.null(comment$replies$data)) {
      results <- rbind(results, extract_comments(comment$replies, comment$id))
    }
  }

  return(results)
}

df <- extract_comments(comments_list)
# saveRDS(df, file = "trans.RDS")
```

```
df <- readRDS("trans.RDS")
```

```
# Word2Vec Exploration

   # Tokenizing, replacing special characters and lowercaseing
     df_clean <- df %>%
        mutate(clean_body = gsub("[^[:alnum:] ]", "", body)) %>%
        mutate(clean_body = tolower(clean_body)) %>%
        mutate(tokens = tokenize_words(clean_body))

   # Remove stop Words
     stop_words <- stopwords("en")

     df_clean$tokens <- lapply(df_clean$tokens, function(words) {
       words[!words %in% stop_words]
     })
```

```r
  # Lemmatization
    df_clean$lemmatized_tokens <- lapply(df_clean$tokens, lemmatize_words)



#  Convert token lists back to sentences for word2vec training
df_clean$sentences <- sapply(df_clean$lemmatized_tokens, paste, collapse = " ")
```

```r
print(head(df_clean))
```

```
##    ups downs comment_id           author parent_id subreddit    link_id score
## 1 1070     0    js80zll        [deleted]      ROOT   r/trans t3_151dseo  1070
## 2  468     0    js8bwgp yinzgahndahntahn   js80zll   r/trans t3_151dseo   468
## 3  258     0    js8ivf4        DrShanks7   js8bwgp   r/trans t3_151dseo   258
## 4   60     0    js9s9t5       stormwind3   js8ivf4   r/trans t3_151dseo    60
## 5   33     0    jsa6p9n          Klaziks   js9s9t5   r/trans t3_151dseo    33
## 6   -5     0    jsa5n68        [deleted]   js9s9t5   r/trans t3_151dseo    -5
##   is_submitter
## 1        FALSE
## 2        FALSE
## 3        FALSE
## 4        FALSE
## 5        FALSE
## 6        FALSE
##                                                                                                       body
## 1                                                                                                [removed]
## 2     u/spez is a notorious transphobe. That’s why he lets "he gets us" target LGBTQIA+ subreddits.
## 3 That dude is very hated in every sub I'm in, lol. The dnd subs are upset with him atm. I love it.
## 4                                                              He's reddit's CEO, if you didn't know
## 5                                                                         Thank you, I had no idea XD
## 6                                                                                                [removed]
##   depth controversiality created_utc
## 1     0                0  1689533115
## 2     1                0  1689537595
## 3     2                0  1689540435
## 4     3                0  1689561335
## 5     4                0  1689569458
## 6     4                0  1689568789
```

```
##                                                                                clean_body
## 1                                                                                 removed
## 2       uspez is a notorious transphobe thats why he lets he gets us target lgbtqia subreddits
## 3 that dude is very hated in every sub im in lol the dnd subs are upset with him atm i love it
## 4                                                                hes reddits ceo if you didnt know
## 5                                                                       thank you i had no idea xd
## 6                                                                                 removed
##                                                                                  tokens
## 1                                                                                removed
## 2 uspez, notorious, transphobe, thats, lets, gets, us, target, lgbtqia, subreddits
## 3                    dude, hated, every, sub, im, lol, dnd, subs, upset, atm, love
## 4                                                  hes, reddits, ceo, didnt, know
## 5                                                                 thank, idea, xd
## 6                                                                          removed
##                                                                    lemmatized_tokens
## 1                                                                             remove
## 2 uspez, notorious, transphobe, thats, let, get, us, target, lgbtqia, subreddits
## 3                    dude, hate, every, sub, im, lol, dnd, sub, upset, atm, love
## 4                                                 hes, reddits, ceo, didnt, know
## 5                                                               thank, idea, xd
## 6                                                                       remove
##                                                                  sentences
## 1                                                                   remove
## 2 uspez notorious transphobe thats let get us target lgbtqia subreddits
## 3                     dude hate every sub im lol dnd sub upset atm love
## 4                                          hes reddits ceo didnt know
## 5                                                 thank idea xd
## 6                                                         remove
```

```r
#  Train the Word2Vec model
paths <- tempfile()
writeLines(df_clean$sentences, paths)
model <- word2vec(paths, dim = 100, window = 5, iter = 10, min_count = 5)



# Sentiment Analysis
# I need to manually label comments or get a LLM like GPT to label them for me.
```

```r
# Calculate average word vectors for each comment
df_clean$avg_vector <- lapply(df_clean$lemmatized_tokens, function(tokens) {
  sapply(tokens, function(token) {
    if (token %in% names(model$vocabulary)) {
      as.vector(get_vector(model, token))
    } else {
      rep(0, 100)
    }
  }) %>%
  Reduce('+', .) / length(tokens)
})


# Convert list column to matrix for model training
avg_vector_matrix <- do.call(rbind, df_clean$avg_vector)

#  Model Training


library(caret)
set.seed(123)
splitIndex <- createDataPartition(df_clean$sentiment, p = .70, list = FALSE,
                                  times = 1)
train_data <- avg_vector_matrix[splitIndex,]
test_data  <- avg_vector_matrix[-splitIndex,]
train_labels <- df_clean$sentiment[splitIndex]
test_labels  <- df_clean$sentiment[-splitIndex]

model_lr <- glm(sentiment ~ ., data = as.data.frame(train_data),
                family = "binomial")
predictions <- predict(model_lr, newdata = as.data.frame(test_data),
                       type = "response")

# Model Evaluation
# Convert predictions to binary class labels
```

```
predicted_labels <- ifelse(predictions > 0.5, 1, 0)
confusionMatrix(as.factor(predicted_labels), as.factor(test_labels))
```

References

Jay Alammar. The illustrated word2vec, 2023. URL http://jalammar.github.io/illustrated-word2vec/. Visualizing machine learning one concept at a time.

Apache Software Foundation. *Apache Kafka Documentation*, 2023. URL https://kafka.apache.org/documentation/.

Jonathan Bratt. Rbert: Implementation of bert in r. https://github.com/jonathanbratt/RBERT, 2023. Apache-2.0 license.

CRAN. *httr: Tools for Working with URLs and HTTP*, 2023a. URL https://cran.r-project.org/web/packages/httr/httr.pdf. R package version 1.4.2.

CRAN. *RSQLite: An Interface to the SQLite Database System*, 2023b. URL https://cran.r-project.org/web/packages/RSQLite/vignettes/RSQLite.html. R package vignette.

McInnes, L, Healy, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, 2023. URL https://umap-learn.readthedocs.io/en/latest/.