NLP Analysis of Google Reviews for Saudi Arabian Sites

Artefact Data Science Assessment

Vishesh Chugh

Table of Contents

Sentiment Analysis Methodology

Overview

```
print(df.shape)
df.head(5)
```

```
(10000, 7)
```

	id	content	date	language	tags	title	ratings
0	377380-203583770957	من الاماكن الهاديه الجميله الممتعه في التسوق ت...	2021-04-11T06:45:00+00:00	ara	[{'value': 'c07bdfc8hb0r13sa7agg', 'sentiment'...	Al Ahsa Mall by Arabian Centres	{'normalized': 100, 'raw': 5}
1	377380-203585579625	مساحة خضراء تتنفس فيها الهواء النقي المناظر	2021-04-11T06:45:00+00:00	ara	[{'value': 'c07bdncbb64t6si78ssg', 'sentiment'...	King Abdullah Park, Sea front	{'normalized': 100, 'raw': 5}
2	377380-203590496913	nice place	2021-04-11T06:45:00+00:00	eng	[{'value': 'c0rlhqgcu1i938rekca0', 'sentiment'...	Green Mountain Resort	{'normalized': 100, 'raw': 5}
3	377380-203589330972	❤جميل	2021-04-11T06:45:00+00:00	ara	[{'value': 'c9ga0skbb64rs4ni6s7g', 'sentiment'...	Waterfront Beach Royal Commission Yanbu	{'normalized': 80, 'raw': 4}
4	377380-203586632060	جمييييل	2021-04-11T06:45:00+00:00	ara	[{'value': 'c07bdncbb64t6si78ssg', 'sentiment'...	Dammam Corniche	{'normalized': 100, 'raw': 5}

- After **removing duplicates (11 rows),** we are left with **9989 reviews**, out of which **~24% is English** "content" or "title" & **76% is Arabic** "content" or "title".
- To proceed with sentiment analysis there are multiple ways one can take to do this, some of them that I could think of (in increasing order of complexity & sophistication):
  1. Use **"ratings"** column to generate sentiment (Higher rating -> positive sentiment; lower rating -> negative sentiment)
  2. **Translate Arabic text to English** and utilize various approaches such as lexicon-based sentiment analysis, pre-trained transformer models.
  3. **Treat Arabic & English text separately**, utilize **CAMeL-Lab/camel_tools** (very sophisticated library developed by NYU Abu Dhabi) to perform sentiment analysis on Arabic Text & lexicon-based/pre-trained for English

     Ref: https://github.com/CAMeL-Lab/camel_tools

Due to limited time, computational power & for ease of understanding the results & EDA. **I finalized approach 2 mentioned above**

Detailed Steps

- Utilized **GoogleTranslator API** to translate Arabic text to English.

```
from deep_translator import GoogleTranslator
```

```
name = 'جميل❤'
name_translated = GoogleTranslator('ar', 'en').translate(name)
print(name_translated)
```

```
Beautiful❤
```

- Used **VADER (Valence Aware Dictionary for Sentiment Reasoning)** model to generate sentiment (positive, negative, neutral) for the text.

  VADER is a **lexicon and rule-based sentiment analysis** tool that is specifically attuned to sentiments expressed in social media

  It is equipped with the following characteristics, which makes it easy to implement directly **without any pre-processing**:

  - typical negations (e.g., "not good")
  - use of contractions as negations (e.g., "wasn't very good")
  - conventional use of punctuation to signal increased sentiment intensity (e.g., "Good!!!")
  - conventional use of word-shape to signal emphasis (e.g., using ALL CAPS for words/phrases)
  - using degree modifiers to alter sentiment intensity (e.g., intensity boosters such as "very" and intensity dampeners such as "kind of")
  - understanding sentiment-laden initialisms and acronyms (for example: 'lol')

    Ref: github.com/cjhutto/vaderSentiment

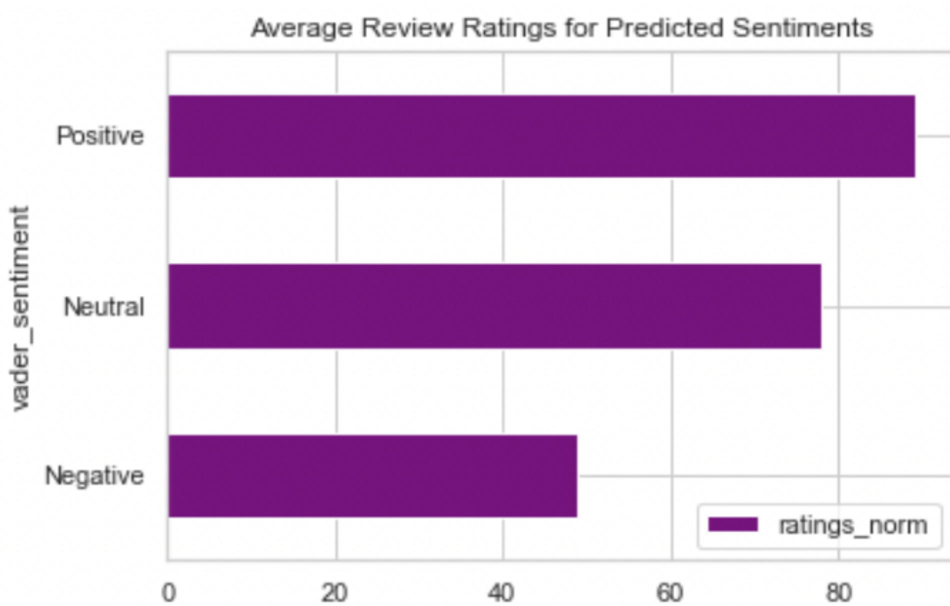
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
def sentiment_scores(sentence):
# Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()
    sentiment_dict = sid_obj.polarity_scores(sentence)
    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        return "Positive"
    elif sentiment_dict['compound'] <= - 0.05 :
        return "Negative"
    else :
        return "Neutral"
```

```
sentiment_scores('A green space where you can breathe fresh air.')
```

```
'Positive'
```

```
sentiment_scores('food was not so great')
```

```
'Negative'
```



Insights

Average ratings (out of 100) seem in-line with the predicted sentiment using VADER. **Positive sentiments have higher average rating** as compared to **Neutral & Negative sentiments.**

Text Preprocessing

We apply the below string cleaning steps after prediction of sentiment for deeper insights:

- Remove non-alphanumerics, URLs using regex
- Convert to lower-case
- Remove custom list of stop-words using nltk
- Tokenize the text & apply WordNetLemmatizer along with using POS tags as inputs

```
from nltk.corpus import wordnet
def get_wordnet_pos(treebank_tag):
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN
```

```
def clean_text(a):
    a_replaced = re.sub('[^A-Za-z0-9]+', ' ', a)
    a_replaced = re.sub(r'w+:/{2}[dw-]+(.[dw-]+)*(?:(?:/[^s/]*))*', '', a_replaced)
    return a_replaced
```

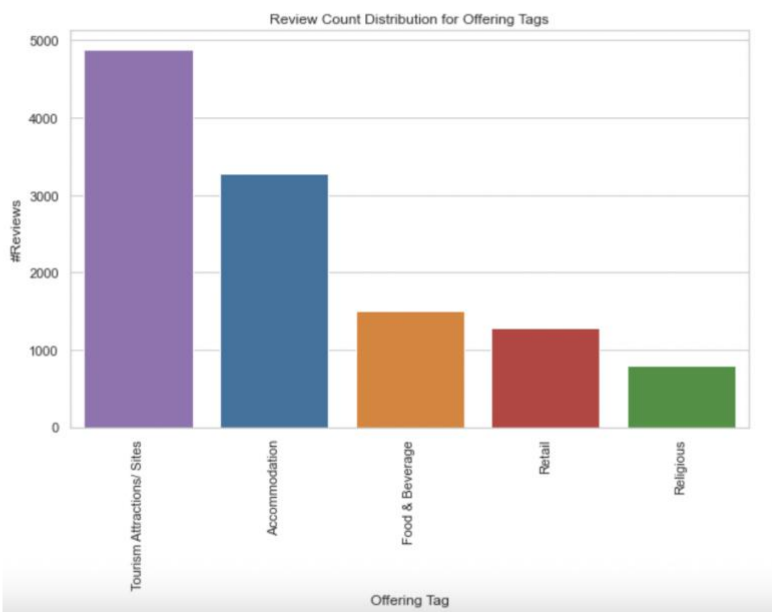
```
w_tokenizer = tokenize.WhitespaceTokenizer()
lemmatizer = WordNetLemmatizer()
def lemmatize_text(text):
    return [lemmatizer.lemmatize(w,get_wordnet_pos(dict(nltk.pos_tag([w]))[w]))
            for w in w_tokenizer.tokenize(clean_text(text.lower()))
            if w not in stopwords.words('english') + ["google","translated"]]

df_nlp['review_lemmas'] = df_nlp['content_en'].apply(lambda x : lemmatize_text(x))
```

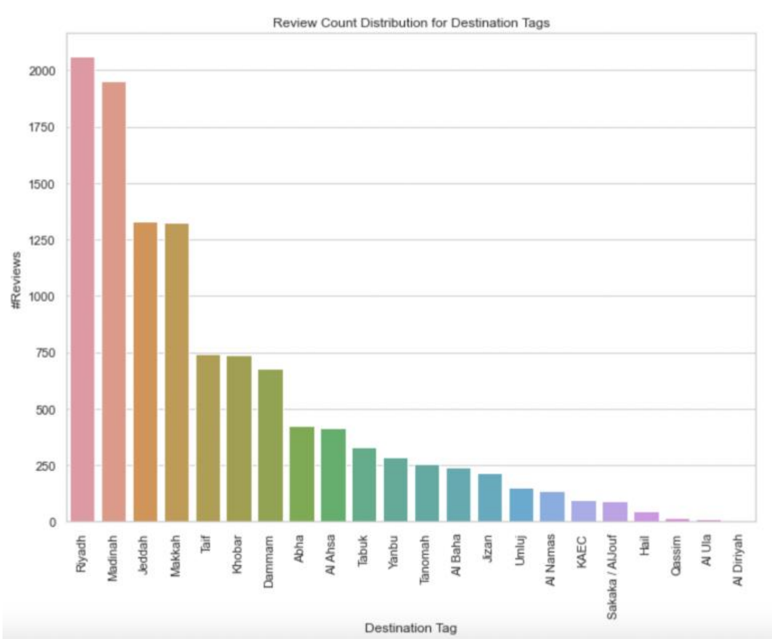
```
lemmatize_text('I am running, so exciting, loving it')
```

```
['run', 'excite', 'love']
```

Findings

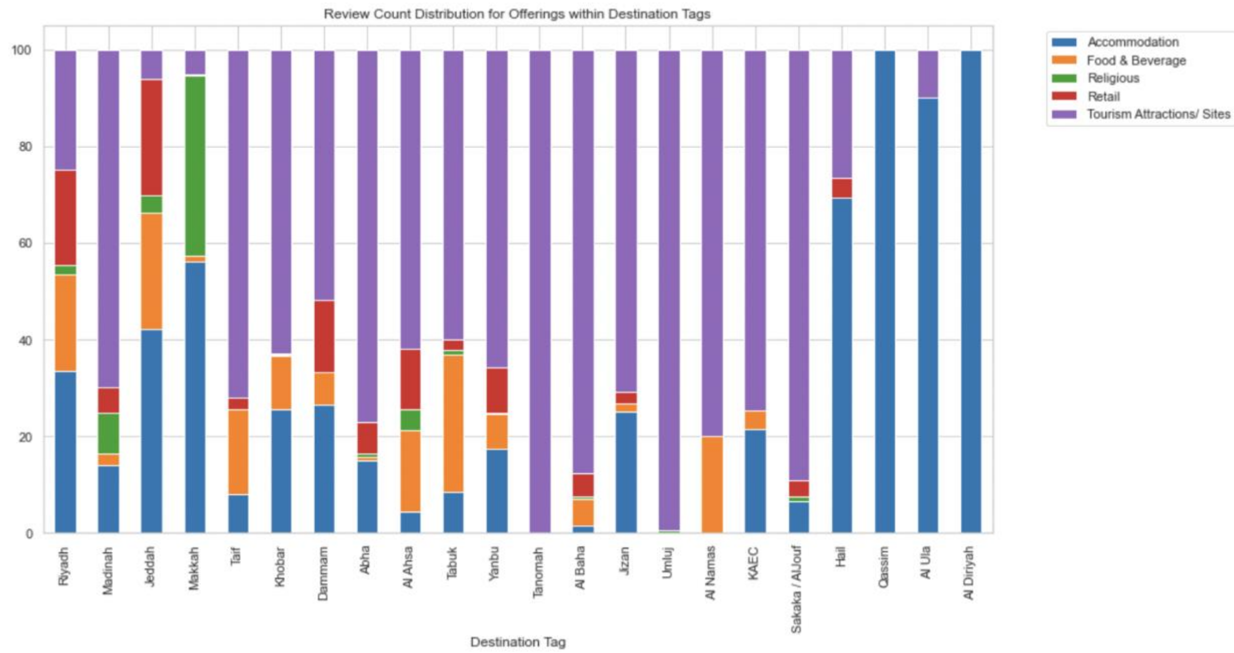
EDA & Insights





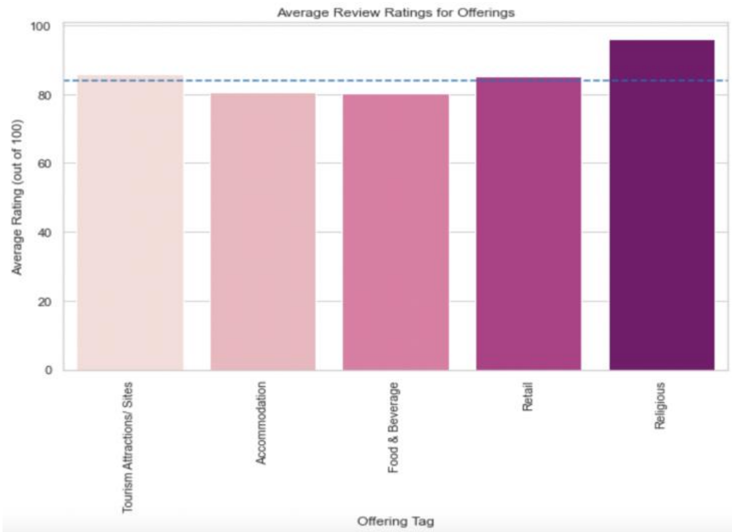
Insights

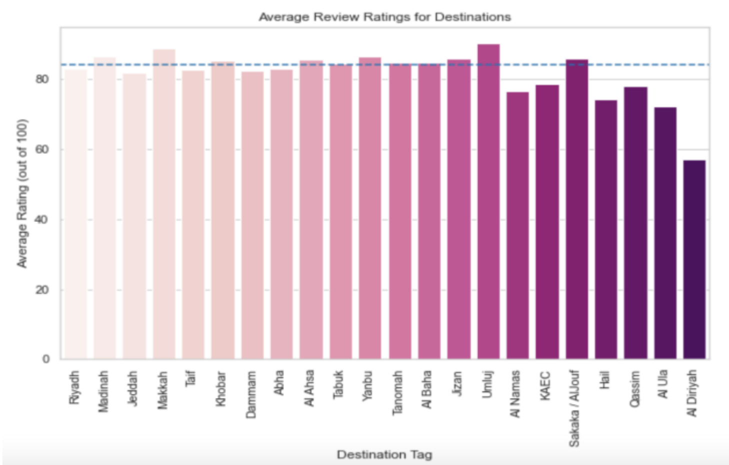
Using the mapping file to extract tags for each review. We see that maximum reviews are related to **Tourism Attraction/Sites & Accommodation** & from major cities like **Riyadh & Madinah.**



Insights

- **Majority of Religious & Accommodation** reviews come from **Makkah**, with almost negligible **Retail and Food & Beverage** reviews.
- **Tourism Attraction** being the dominant reviews overall, are found to be the least in **Jeddah & Makkah**.
- **~90% reviews in Jeddah** come from **Accommodation, Food & Beverage and Retail**
- **Madinah** is dominated by **Tourism Attraction reviews ~70%**
- **Riyadh** consists of an approximately even distribution of **Accommodation & Tourism Attraction** and **Food & Beverage & Retail** reviews.





Insights

- **Average overall rating in the dataset = 84 (blue dotted line),** average rating for **Religious** tag is **significantly higher** than the overall average. Correspondingly, average rating for **Makkah** is also **higher than the overall average** since most of its reviews come from the **Religious** category.
- Average rating for **Accommodation and Food & Beverage** tags is **lower than the overall average** of the dataset. Correspondingly, average rating for **Jeddah** is **marginally lower than the overall average** since ~65% of its reviews are derived from **Accommodation and Food & Beverage** categories

Recommendations

Overall Key Strategies	Accommodation	Food & Beverage	Tourism	Religious	Retail
1	Deploy **well trained reception staff for a smooth check-in/out experience** Offer complementary welcome drinks	Keep a check on **quality of service & experience provided** without compromising on the taste	Deploy **public facilities like bathrooms** and maintain proper hygiene and cleanliness	Effective communication of **mosque social etiquette** for tourists, respecting tradition & culture	Efficient **parking lot management** for elevated end-to-end shopping experience
2	Keep a check on **old room furniture, cleanliness & room service**	Focus on **proper functioning of air-conditioner, maintaining hygiene & cleanliness**	**Devise proper seating arrangement** for tourists at public areas.		
3	A lot of tourists visit **Makkah** for religious purposes, **positive accommodation experience** is key for tourists	Better efficiency leading **to lesser waiting time** for customers			