Text Mining Analysis of Nike Instagram Posts

AnhThu Duong

1. Objectives

The goal of this project was to perform text mining and topic modeling on posts from Nike's Instagram account. By analyzing the text data, the objective was to uncover insights about common themes, sentiments, and topics discussed in Nike's social media content.

2. Data Collection

Data was collected by downloading all posts and captions from Nike's official Instagram account using the Python instaloader library. This allowed programmatic access to scrape the text data from Nike's posts.

3. Data Preprocessing

The raw text data from the Nike Instagram posts went through an extensive preprocessing pipeline to prepare it for topic modeling analysis. Here are the specific steps:

3.1.Tokenization to split text into individual words/tokens

The text was split into individual words or tokens using the gensim.utils.simple_preprocess function. This breaks up the sequences of text into lists of tokens.

3.2.Removing URLs, hashtags, and usernames

Removing URLs, Hashtags, and Usernames Regular expressions were used to remove any URLs starting with "http" or "https". Hashtags (words starting with #) and usernames (words starting with @) were also stripped out, as these tend to be noisy for topic modeling.

```
# remove http or https
df["link"] = df["Text"].apply(lambda s: " ".join(w for w in s.split() if w.startswith("http")))
df["AnalyzedTextWithoutHttp"] = df["Text"].apply(lambda s: " ".join(w for w in s.split() if not w.startswith("http")))
```

```
# Remove hashtag
df["hashtag"] = df["AnalyzedTextWithoutHttp"].apply(lambda s: " ".join(w for w in s.split() if w.startswith("#")))
df["AnalyzedTextWithoutHttpHashtag"] = df["AnalyzedTextWithoutHttp"].apply(lambda s: " ".join(w for w in s.split() if not w.startswith("#")))
```

```
# Remove username
df["user_mentioned"] = df["AnalyzedTextWithoutHttpHashtag"].apply(lambda s: " ".join(w for w in s.split() if w.startswith("@")))
df["AnalyzedTextWithoutHttpHashtagUserName"] = df["AnalyzedTextWithoutHttpHashtag"].apply(lambda s: " ".join(w for w in s.split() if not w.startswith("@")))
```

```
# Remove non-alphabatic
df["AnalyzedTextWithoutHttpHashtagUserNameNonAlpha"] = df["AnalyzedTextWithoutHttpHashtagUserName"].apply(lambda s: " ".join(w for w in s.split() if w.isalpha()))
```

3.3.Removing stop words (common words like "the", "and", etc.)

Common English stopwords like "the", "and", "a", etc. were removed using the nltk.corpus stop words list. These ubiquitous words can bias the topic modeling.

```
import nltk
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
df["AnalyzedText"] = df["AnalyzedTextWithoutHttpHashtagUserNameNonAlpha"].apply(lambda s: " ".join(w for w in s.split() if w.lower() not in stop_words))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/anhthu/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
df.head(2)
```

	Text	link	AnalyzedTextWithoutHttp	hashtag	AnalyzedTextWithoutHttpHashtag	user_mentioned	AnalyzedTextWithoutHttpHashtagUserName	AnalyzedTextWithoutHttpHashtagUserNameNonAlpha	AnalyzedText
0	Yesterday I found new purpose in tomorrow. Tod...		Yesterday I found new purpose in tomorrow. Tod...		Yesterday I found new purpose in tomorrow. Tod...		Yesterday I found new purpose in tomorrow. Tod...	Yesterday I found new purpose in	Yesterday found new purpose
1	The Nike LunarGlide+ 4 Chicago edition. Built ...		The Nike LunarGlide+ 4 Chicago edition. Built ...	#nike #running #marathon	The Nike LunarGlide+ 4 Chicago edition. Built ...		The Nike LunarGlide+ 4 Chicago edition. Built ...	The Nike Chicago Built to go the Designed in c...	Nike Chicago Built go Designed collaboration w...

3.4.Lemmatization

Each token was lemmatized, converting it to the root form of the word using nltk's WordNetLemmatizer. For example, "runs" becomes "run". This groups together inflected forms of a word.

3.5.Generating Bigrams and Trigrams

The gensim.models.Phrases library was used to model common two-word (bigram) and three-word (trigram) phrases in the data, such as "find_way" or "take_first_step". This allows multi-word phrases to be treated as single tokens in the topic modeling.

```
data_words = list(sent_to_words(data))
```

```
## remember bigram and trigram? Let's check whether there is a differnce between usage of bigram vs trigram for text mining
# Build the bigram and trigram models
bigram = gensim.models.Phrases(data_words, min_count=5, threshold=100) # higher threshold fewer phrases.
trigram = gensim.models.Phrases(bigram[data_words], threshold=100)
# Phrases: Automatically detect common phrases – multi-word expressions / word n-grams – from a stream of sentences

# get a sentence clubbed as a trigram/bigram
bigram_mod = gensim.models.phrases.Phraser(bigram)
trigram_mod = gensim.models.phrases.Phraser(trigram)

# See bigram and trigram example
print(bigram_mod[bigram_mod[data_words[5]]])
print(trigram_mod[bigram_mod[data_words[5]]])
```

```
['get', 'playing']
['get', 'playing']
```

```
## you can do additional analysis

def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]

def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

def make_trigrams(texts):
    return [trigram_mod[bigram_mod[doc]] for doc in texts]

# lemmatization: achieve the root forms
def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags]) #token.lemma_: root of token; token.
    return texts_out
```

After this preprocessing, the clean tokenized text data was ready for input into the topic modeling algorithms.

4. Topic Modeling

4.1. Topic Modeling Definition and Purpose

Topic modeling is an unsupervised machine learning technique used to discover the abstract "topics" that occur in a collection of documents. It analyzes the word frequencies and co-occurrences within the documents to learn the topics automatically from the data itself. The goal of topic modeling is to uncover the hidden thematic structure and dominant subject areas present in a text corpus. This can provide valuable insights when analyzing large volumes of unlabeled and unstructured text data such as social media posts, news articles, academic papers, etc.

4.2. Latent Dirichlet Allocation (LDA)

LDA is one of the most common algorithms for topic modeling. It is a generative statistical model that explains the documents as arising from multiple topics, where each topic is a distribution over words. The documents themselves are distributions over the topics.

Specifically, LDA assumes:

- Each document exhibits multiple topics in different proportions.
- Each word in a document is drawn from one of the topics.

The "topics" extracted are distributions over the words - the words that tend to co-occur most frequently are grouped into the same topic.

LDA has several advantages:

- It automatically learns the topics without requiring any upfront topic definitions.
- The number of topics is a user-defined parameter that can be tuned.
- It accounts for the fact that documents have multiple topics present.

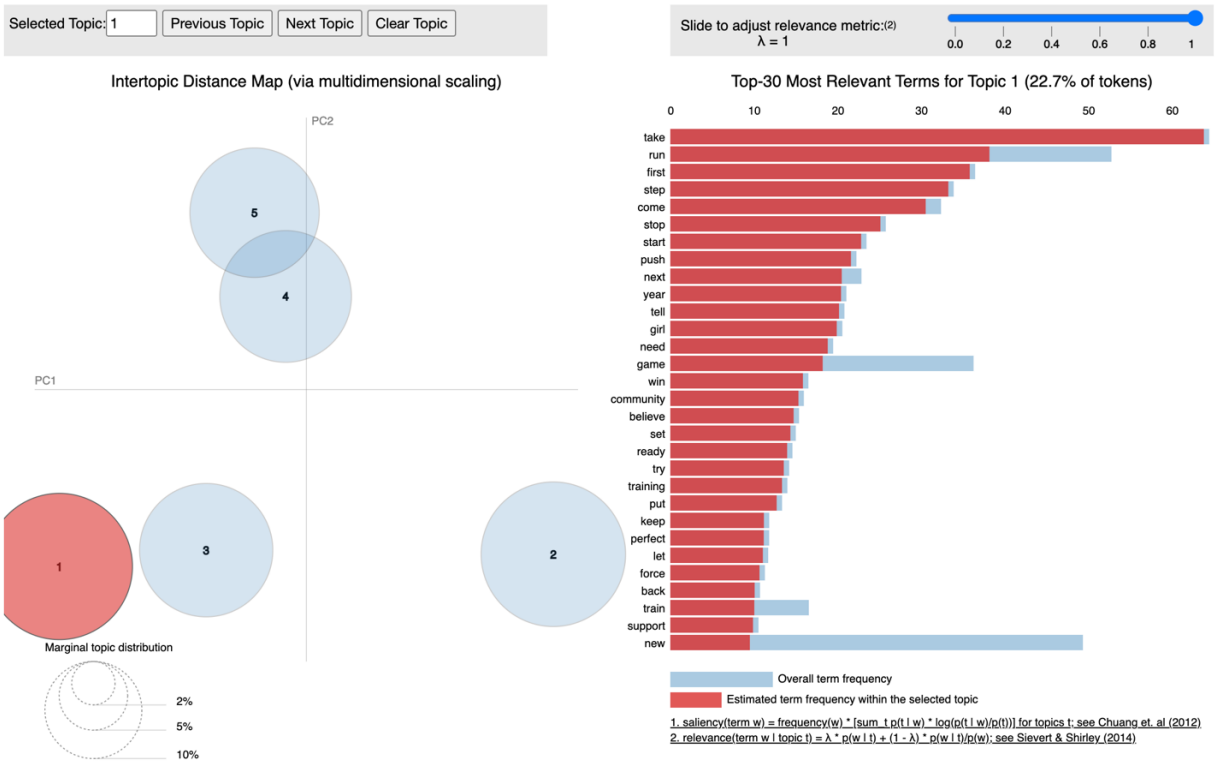
4.3. Using LDA for Nike Post Analysis

For this analysis, the gensim LDA model implementation was used. After the text data preprocessing, the tokenized posts were used to train the LDA model. Based on initial model evaluations, the optimal number of topics was set to 5 for this dataset.

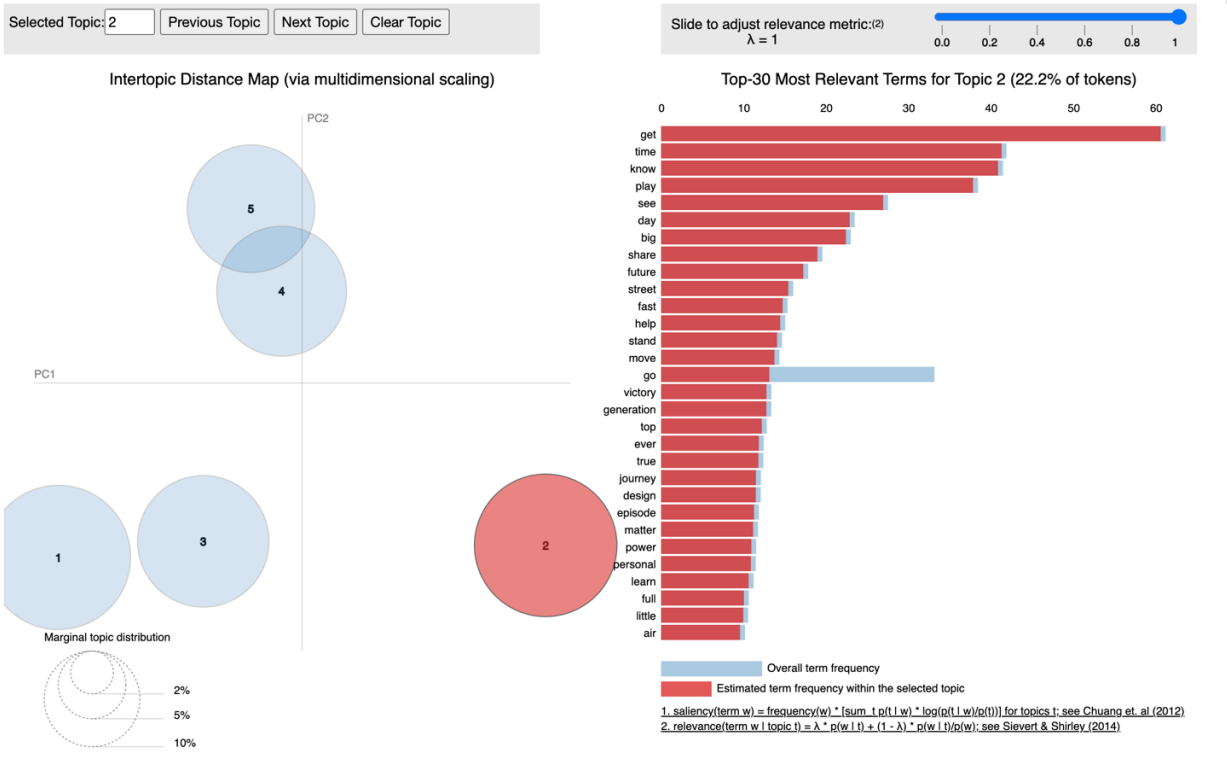
```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=5,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

The LDA model then extracted the 5 key topics and their associated word distributions present in the Nike posts. Each post was labeled with its most prevalent topic, which enabled analyzing the proportions of different themes in the content. Some of the top words for the 5 topics were:

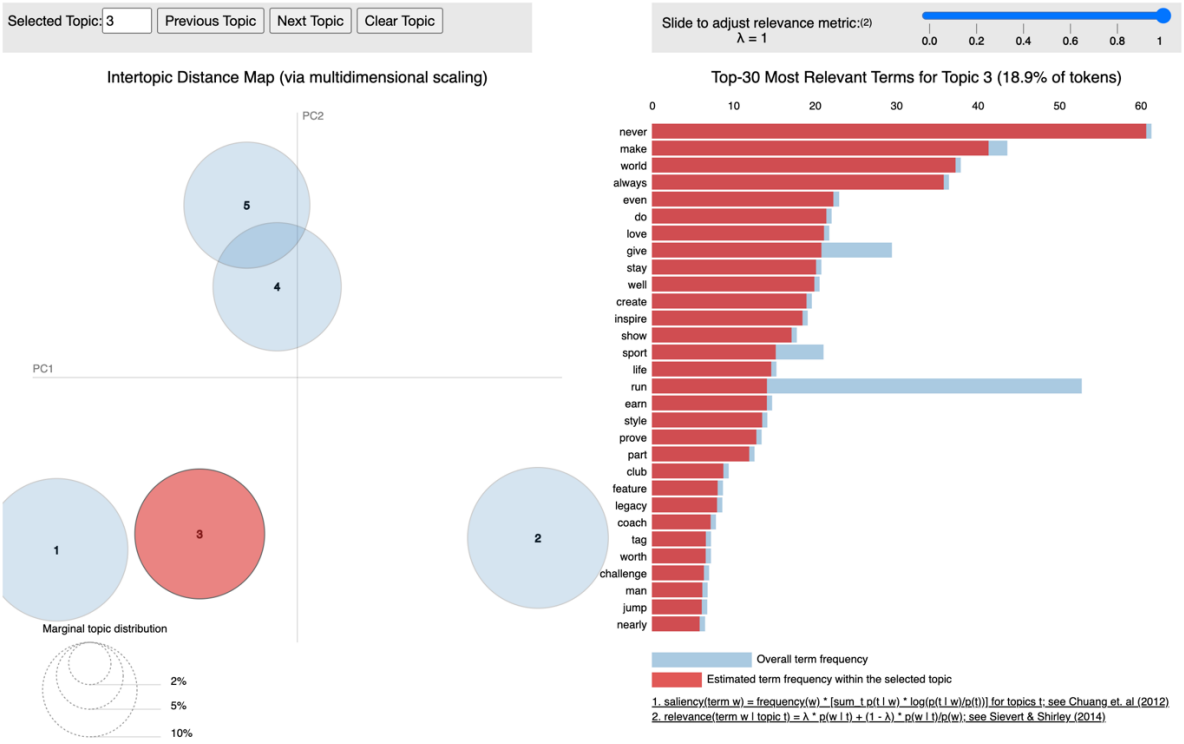
Topic 1: find, way, want, mean, game



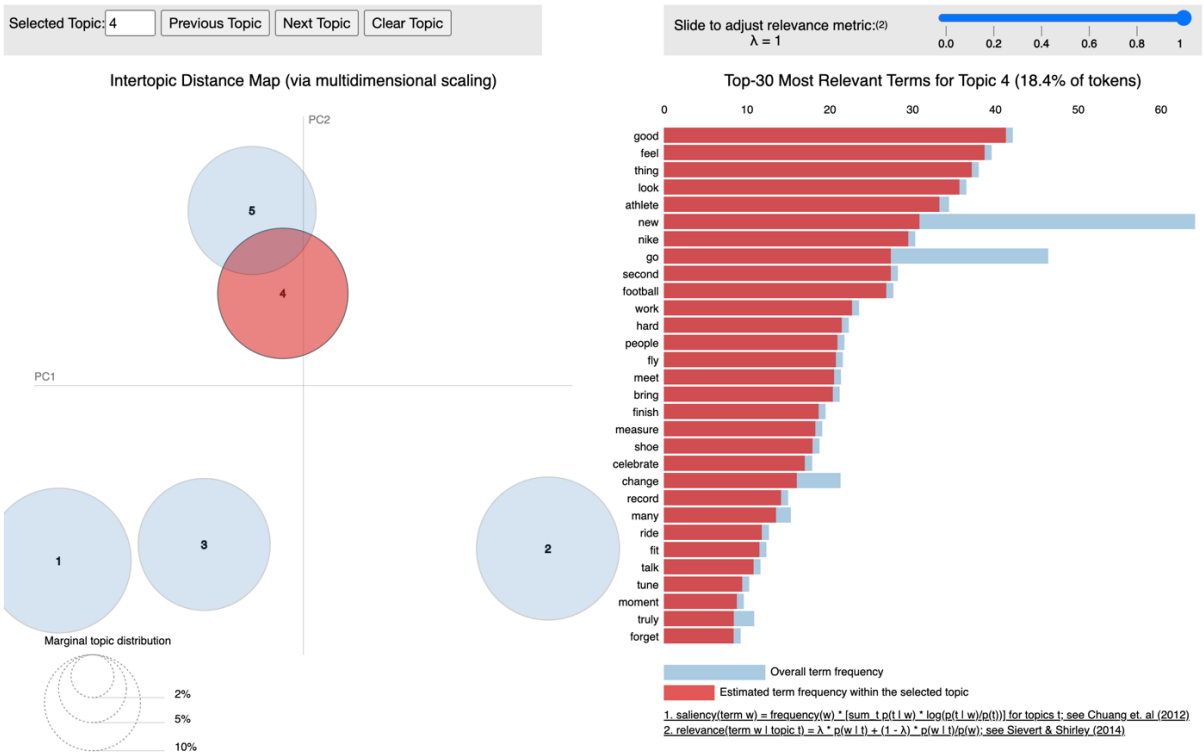
Topic 2: take, run, first, step, come



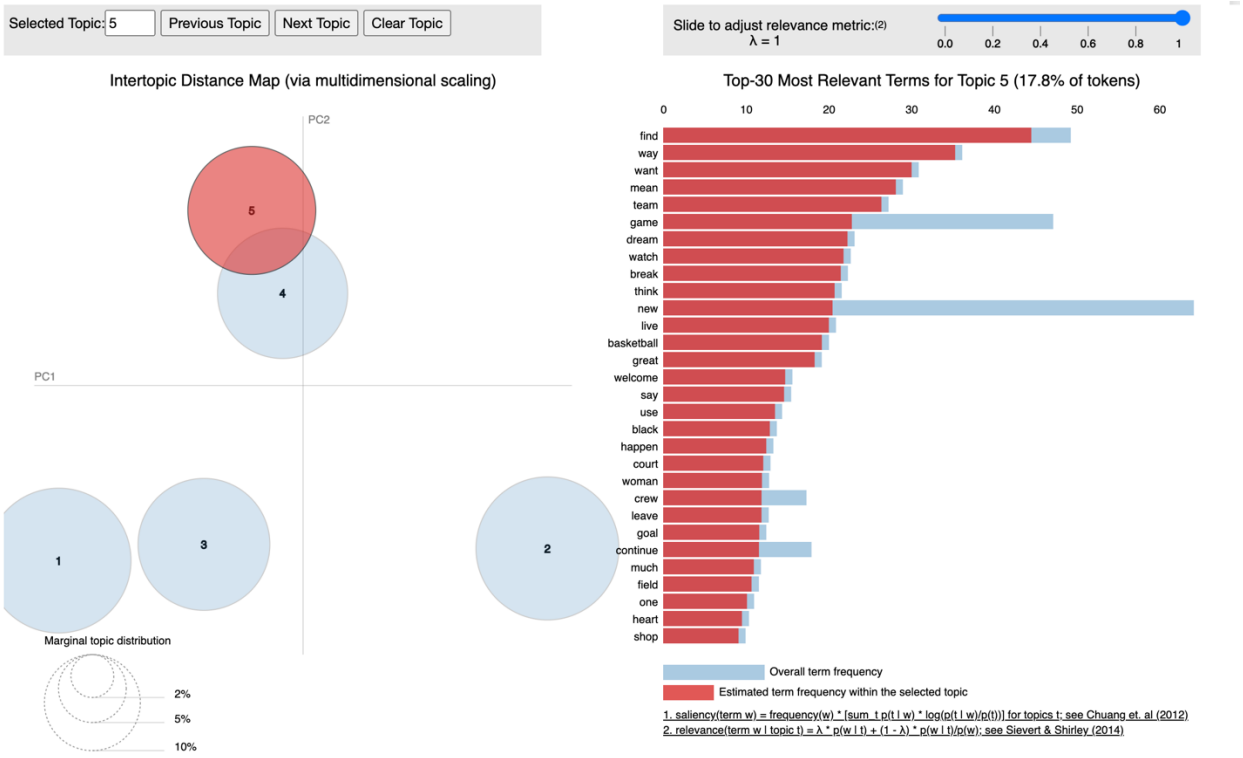
Topic 3: get, time, know, play, see



Topic 4: never, make, world, always, love



Topic 5: good, feel, thing, look, athlete



Interpreting these topics provides insight into the common subject areas and messages that Nike was expressing over social media to their audience. Based on analyzing the top words for each of the 5 topics from the LDA model, we can label them with representative names:

Topic 1 (***Motivation/Purpose***) relates to messages about finding your purpose, desires, and meaning through athletic pursuits and competition.

Topic 2 (***Action/Determination***) covers content encouraging people to take action, make a start, push themselves, and themes of grit and determination.

Topic 3 (***Living in the Moment***) comprises posts about being present, knowing yourself, playing in the moment and cherishing the journey.

Topic 4 (***Perseverance/Passion***) captures persevering no matter what, making things happen, having passion, and creating positive change in the world.

Topic 5 (***Athletics/Nike Products***) represents posts about the joy of athletics, discussing Nike's products, and promoting their athletic brand.

By naming the topics, it becomes easier to interpret and summarize the main themes present in the Nike Instagram posts. LDA proved to be a powerful unsupervised technique to automatically discover and quantify the underlying themes present in this text data in a data-driven manner.

5. Sentiment Analysis

In addition to topic modeling, the posts were scored for positive, negative, and neutral sentiment using the VADER (Valence Aware Dictionary and Sentiment Reasoner) analyzer. This provided insight into the emotional tone of the Nike content.

The analysis found that Topic 1 (find, way, want...) had the highest positive sentiment, while Topic 2 (take, run, first...) had the highest negative sentiment.

6. Key Insights

Some key insights that can be drawn from the analysis:

- Nike's Instagram posts cover a range of themes like finding purpose, taking action, the joy of sports/athletics, making the world better, and promoting their brand/products.
- While much of the content has an overall positive, inspirational tone, there are some streams discussing the difficulties of pushing oneself (Topic 2's negative sentiment).
- Combining topic and sentiment analysis can pinpoint which specific themes resonate as more positive or negative with the audience.

This type of analysis could help brands like Nike understand what types of content and messaging best resonates with their audience on social media. The insights could then inform their ongoing social media and content marketing strategy.