A Pipeline for Characterizing Fashion Items' Online Popularity

Lisa Kailai Han, Anwen Huang, Lexie Li, Joyce Moon, Dasson Tan

Project Advisors: Peter Freeman, Rebecca Nugent, Xiaoyi Yang

Introduction & Dataset

Background

- The fashion retail industry is interested in using novel sources of data to forecast item popularity. As a preliminary step, it could be useful to explore how to generate data that best characterize an item's online presence.

Objective

- Identify features that characterize an apparel's online popularity and explore clustering relationships among items.

Data Set

- Provided by **Chain of Demand** firm which uses AI to track and predict sales demand for fashion brands
- 2 spreadsheets from brands *Uniqlo* and *Esprit* containing item-level features
- **2382** Uniqlo items; **188** Esprit items

Product.Name <fctr>	Category_1 <fctr>	Category_2 <fctr>	Color <fctr>	Material <fctr>
AIRism SLEEVELESS TOP 422267	women	sport utility wear	GRAY	59% Nylon, 31% Cupro, 10% Spandex
HEATTECH SLEEVELESS TOP 418327	women	innerwear & loungewear	OFF WHITE	38% Polyester, 32% Acrylic, 21% Rayo
CORDUROY MINI SKIRT 418882	women	bottoms	YELLOW	99% Cotton, 1% Spandex
FLEECE SET (LONG SLEEVE) 421705	women	tops	BROWN	Tops: 100% Polyester/ Rib: 90% Cottc
HEATTECH WARM LINED PANTS 420360	women	bottoms	NAVY	Shell: 90% Polyester, 10% Spandex/ L

Table 1: Five Random Samples From Uniqlo Dataset

Data Processing

Search Term Generation

- **The goal:** from a row in the dataset *(e.g. Table 1)*, automatically extract search terms similar to how people would actually search this item online
- **Examples:** {sleeveless top; heattech innerwear; off white loungewear; polyester sleeveless top} are all valid search terms people would choose to query this item online

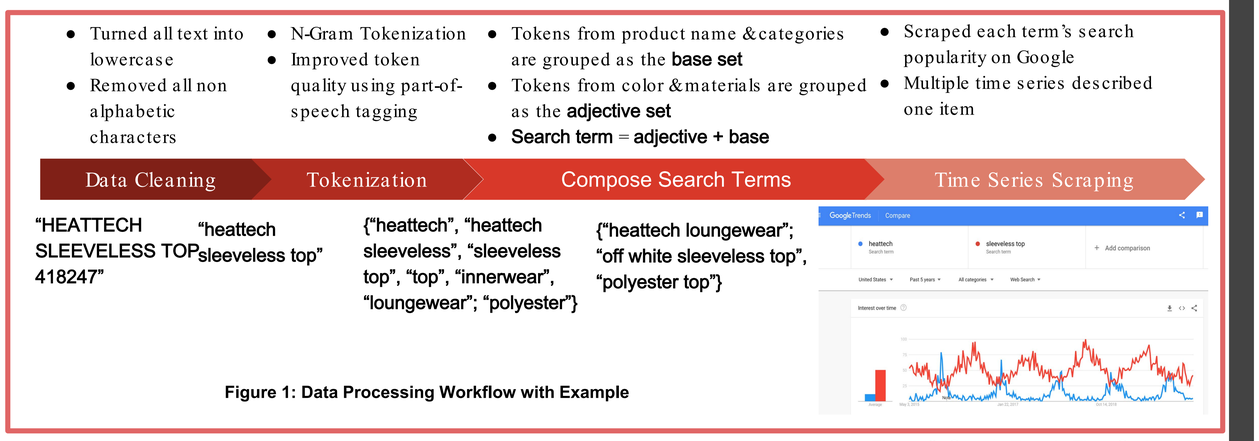

Figure 1: Data Processing Workflow with Example

Figure 2: Google Trends Time Series

Time Series Generation

- Used pytrends, an unofficial API for Google Trends, to download historical interest over time for each search term.
- Weekly timestamps from **Jan. 2016 to Dec. 2019**
- Score from 0 to 100 is assigned to indicate each term's current search frequency compared to its historical high
  - Resulting numbers not comparable across search terms, but meaningful within each term over time
- Maxed out on Google's rate limit for API calls; only generated time series for 100 *Uniqlo* items

Feature Extraction & Clustering

Feature Aggregation

- For each item, we have several time series
- All term's trend scores are averaged for each item before feature extraction
- Some features encompass multiple values and are thus broken into different features

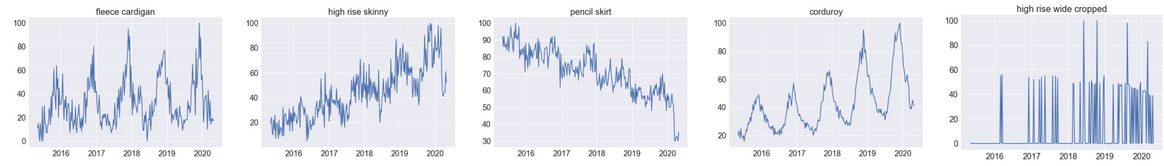

Figure 3: Example Time Series Trends We Wish to Capture

Time Series Feature Extraction

- Want to capture **cyclical, rising, dropping, or booming trends**
- Features extracted using the TSFEL (Time Series Feature Extraction Library) package that provides three domains:
  - **Spectral**: describe the spectrum by representing the series with combinations of sinusoids
  - **Statistical**: describe the observed values' marginal distribution
  - **Temporal**: describe the time features of the native time series
- Focus on features like autocorrelation, entropy, percentile, slope, and model coefficients.

Results

Clustering

- Clustering done using K-means Algorithm
- Use 3 and 4 clusters

Improvements

- Trends of clusters not very clear
- Data points in sample too similar
  - API limitation on downloadable data
- Search term categories too heavily weighted

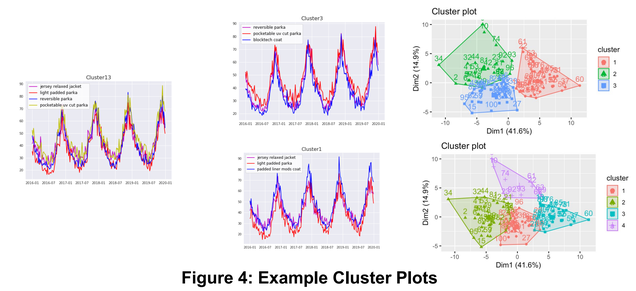

Figure 4: Example Cluster Plots

Social Media

Summary

- Google: Monthly search volume from Google Ads
  - Monthly search volume's range is to big (ex. 1K-10K,10K-100K)
- Instagram: Attempted to input search terms into a keyword/hashtag tracking service
  - Tracking service not applicable for high volume of keywords

References

- RWeka Library in R for tokenization
- Natural Language Toolkit library nltk for part of speech tagging
- Pytrends for scraping Google Trends data
- TSFEL (Time Series Feature Extraction Library) Package for time series feature extraction
- Base R K-means Algorithm for clustering
- Factoextra Package in R for visualization of clustering
- Data was provided by *Chain of Demand*, a firm which uses AI and big data to track and predict sales demand for fashion brands