Unstructured Data Analytics (F2021)

Assignment 2

New due date: 29th October by 11:59 p.m.

Building a Crowdsourced Recommendation System

High level description: The objective of this group assignment is to create the building blocks of a crowdsourced recommendation system. This recommendation system should accept user inputs about desired attributes of a product and come up with 3 recommendations.

Obtain reviews of craft beer from beeradvocate.com. I would suggest using the following link, which shows the top 250 beers sorted by ratings:

https://www.beeradvocate.com/beer/top-rated/

The nice feature of the above link is that it is a single-page listing of 250 top-rated beers (avoids the pagination feature, which you need in cases where listings go on for many pages). The way beeradvocate.com organizes reviews is that it provides about 25 reviews per page. The output file should have 3 columns: product_name, product_review, and user_rating.

Task A. Extract about 5-6k reviews.

Task B. Assume that a customer, who will be using this recommender system, has specified 3 attributes in a product. E.g., one website describes multiple attributes of beer:

https://www.dummies.com/food-drink/drinks/beer/beer-for-dummies-cheat-sheet/

- **Aggressive (**Boldly assertive aroma and/or taste)
- **Balanced:** Malt and hops in similar proportions; equal representation of malt sweetness and hop bitterness in the flavor — especially at the finish
- **Complex:** Multidimensional; many flavors and sensations on the palate
- **Crisp:** Highly carbonated; effervescent
- **Fruity:** Flavors reminiscent of various fruits **or Hoppy:** Herbal, earthy, spicy, or citric aromas and flavors of hops o**r Malty:** Grainy, caramel-like; can be sweet or dry
- **Robust:** Rich and full-bodied

A word frequency analysis of beer reviews may be a better way to find important attributes.

Assume that a customer has specified three attributes of the product as being important to him or her.

Task C. Perform a **similarity** analysis using cosine similarity (without word embeddings) with the 3 attributes specified by the customer and the reviews. From the output file, calculate the average similarity between each product and the preferred attributes.

For similarity analysis, use cosine similarity with bag of words. The script should accept as input a file with the product attributes, and calculate similarity scores (between 0 and 1) between these attributes and each review. That is, the output file should have 3 columns – product_name (for each product, the product_name will repeat as many times as there are reviews of the product), product_review and similarity_score.

Task D. For every review, perform a sentiment analysis.

Task E. Assume an evaluation score for each beer = average similarity score + average sentiment score.

Now **recommend 3 products** to the customer.

Task F. How would your recommendation change if you use word vectors (the spaCy package would be the easiest to use with pretrained word vectors) instead of plain vanilla bag-of-words cosine similarity? One way to analyze the difference would be to consider the % of reviews that mention a preferred attribute. E.g., if you recommend a product, what % of its reviews mention an attribute specified by the customer? Do you see any difference across bag-of-words and word vector approaches? This article may be useful: https://medium.com/swlh/word-embeddings-versus-bag-of-words-the-curious-case-of-recommender-systems-6ac1604d4424?source=friends_link&sk=d746da9f094d1222a35519387afc6338

Note that the article doesn't claim that bag-of-words will always be better than word embeddings for recommender systems. It lays out conditions under which it is likely to be the case. That is, depending on the attributes you use, you may or may not see the same effect.

Task G. How would your recommendations differ if you ignored the similarity and feature sentiment scores and simply chose the 3 highest rated products from your entire dataset? Would these products meet the requirements of the user looking for recommendations? Why or why not? Justify your answer with analysis. Use the similarity and sentiment scores as well as overall ratings to answer this question.

Here is a sample web implementation of a recommender system based on the same principles (runningshoe4you.com), but in this assignment, we will not have the time for this type of full automation.

Y**our submission (python notebook) should include the following:**

(i) Names of all team members **inside** the python notebook (only one submission per team) including morning/late morning cohort information.
(ii) All scripts
(iii) The sentiment and similarity scores for the three products you recommended in task E.
(iv) Your analyses for and answer to task F. Make sure you show the ratings, similarity scores and sentiments for the products you recommend in tasks E and F. Use tables whenever possible.