Project Report

Author: Xiaoxin Gan, Jianqi Ma, Kevin Delgado

Introduction

Our project focuses on Instagram data curation for tobacco research. As the use of tobacco becomes more prevail among modern society, having appropriate tobacco use warning can effectively shape people's idea on the usage of tobacco. Therefore, it is important to see whether people have appropriate warnings when talking about tobacco related topics.

In this project, we crawled posts, photos and corresponding comments from two user accounts which contain heavy tobacco content, **Swisher Sweets** and **Backwoods**, on Instagram from 2018 to 2020. We annotated through every posts and comment, which are in a text format, and implemented 3 different labels:

1. whether they contain tobacco use warnings or not
2. whether they contain explicit sponsorship hashtags (#sponsored, #ad, #paid) or not
3. whether they contain ambiguous sponsorship hashtags (#thanks, #sp, #spon, #ambassador, #collab) or not

Find our code on Github

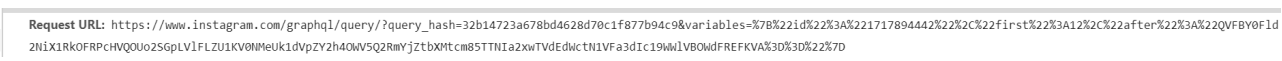
Data Crawling

Details

1. Before crawling, we first need to parse the URL of the corresponding user's instagram homepage. The URL of the user's instagram homepage given in the project requirement, like "https://www.instagram.com/swishersweets/". We can obtain the text of the html file by sending a GET request to this URL. After getting the source code of the html, we find that all the posts pages or URLs is in a Javascript object.

   We can get the first 12 posts urls in this object, but the first 12 posts is created in 2021, and what we need is the posts from 2018 to 2020.

2. At this point, we need to find how to find more posts in the homepage. When we scroll down the page, more pages occur, and new requests in the browser are generated. We find that the browser send request to this page.



And we look up the response of this request. We can see that there are`has_next_page` and `end_cursor` two values, and actually we can get next 12 posts by using `end_cursor` and the `user_id`, which got from the source page.

3. And we find that all the data, like photos URLs, posts and time to create the posts are contained in a json object.

We just need to save what we need in a list to build our own dataset, and when searching for each post , we should compare the `taken_at_timestamp` with the start timestamp and the end timestamp to filter out the posts from 2018 to 2020.

4. According to the comments, all the comments are saved in an independent URL, same as posts.

5. After saving what we need, such as photos URLs, comments and posts, we just write the comments and posts on disks, then iterate all the photos URLs to send a GET request and download the photos.

Challenges

It is tough to crawl the data, since we have not applied the open-source tools, instead, we analyze the HTML source code of Instagram, and the logic behind each requests made to query the desired data. The following are some problems we met during scraping data:

1. When we crawl, we actually make an HTTP request to the URL. In order to query to the Instagram, there should be a certification process. To solve this, we copied the certification cookie we got in the browser to the request header.
2. It is easy to reach Instagram's connection request limit if we don't record what the limit is. If we send request to Instagram many times in a short time, it will return a 429 code. So, we add a sleep action after each request action, when we receive a 429 code, we sleep for a period of time and retry to send request. Doing so allow us to bypass Instagram's request limitation.

Data Annotation

Explicit and Ambiguous Sponsorship Hashtags

For explicit and ambiguous sponsorship hashtags, it is simple to annotate. In order to make the process easier, we extract all hastags contained in the posts/captions/comments, and simply compare that with our target hastags. If it contains any target hashtags, we annotate it with True, otherwise, False. The extracted hashtags are stored in the result csv file as well for future uses.

- Target Explicit Sponsorship Hashtags: **#sponsored**, **#ad**, **#paid**
- Target Ambiguous Sponsorship Hashtags: **#thanks**, **#sp**, **#spon**, **#ambassador**, **#collab**

Tobacco Use Warnings

We label posts/captions/comments for whether it has tobacco warning by utilizing the Language-agnostic BERT sentence embedding (LaBSE). The LaBSE is a multiligual sentence embedding model released by Google, which are often used for transltaion, text classification, semantic similarity, clustering and other natural language tasks.

We started by collecting sample tobacco use warning from the following government regualtions pages:

- Cigarette Labeling and Health Warning Requirements
- 2000 Surgeon General's Report Highlights: Warning Labels

With the knowledge of what possible tobacco use warning looks like, we use LaBSE to encode those sample into high-demension vectors as a standard used to determine whether a post/caption/comment contain any tobacco use warning.

After that, we also encoded posts/captions/comments we scraped from Instagram into vectors using LaBSE. The multilingual property of LaBSE could be useful when we try to classify posts/captions/comments since there are content wirtten in language other than English (Specifically, Japanese and Spanish).

Now that we have both target and sample encoded for comparison, we can calculate the similarity between those vectors using dot product similarity. We choose the dot product similarity over cosine similarity because the computation process for dot product are much simpler than cosine similarity, and dot product similarity take the length of a vector into consideration while cosine similarity only look at the direction. So we think using dot product similarity might generally be a better choose.

As we were using some of the sample code from LaBSE to do embedding, it's easy to reach the limit of RAM, since when we send many sentences together to the model, the weights size can be extremely large. So like the comments, we set a batch size as 1000, and the max sentence length 256 to do embedding.

After embedding finishing, we just multiply the comments embeddings or the posts embeddings with the example embeddings, note that this is matrix multiplication. And we look at the new matrix line by line, if there is a value higher than the similarity threshold we set manually, which is 0.5 for intuition, we annotate that sentence containing tobacco use warnings.

Result

The following are numeric results from our annotation of **Swisher Sweets** and **Backwoods**'s posts/captions/comments using LaBSE:

- Over the 1484 posts we examinated, there are 0 explicit hashtag found, 0 ambiguous hastag found, and only 3 posts with possible tobacco warning.
- Over the 27798 comments we examinated, there are 0 explicit hashtag found, 5 ambiguous hastag found, and about 811 comments have a similarity over 0.5 compare to the government's cigarette labeling.
- Among the 797 posts by **Swisher Sweets**, no sign of any form of tobacco warning (including sponsorship hastags and tobacco warning sentences) are observed.
- Among the 687 posts by **Backwoods**, no sponsorship hashtags are observed; yet there are 3 posts containing possible tobacco warning sentences.
- Among the 11722 comments by **Swisher Sweets**, there are 0 explicit hashtag found, 5 ambiguous hastag found, and 9 comments with a similarity over 0.5 compare to the government's cigarette labeling.
- Among the 16076 comments by **Backwoods**, there are 0 explicit hashtag found, 0 ambiguous hastag found, and 802 comments with a similarity over 0.5 compare to the government's cigarette labeling.

Drawbacks

In general, we believe text classification for comments would be more accurate than posts. In contrast to the all-text content of the comments, the posts have pictures as well. So some potential warning may be contained in the picture instead of the caption of the post, which would eventually lead to a bias on the percentage of how many posts contain tobacco warning. We have all pictures from both **Swisher Sweets** and **Backwoods** in 2018, 2019, and 2020 scraped for future use.

Other than that, the threshold we set for comparing similarity is 0.5, intuitively, but this might result in some false classification. Future studies should try to collect enough data to train a model or try to find a better threshold.