# Project Report

## Introduction

This project requires us to crawl the posts, photos and corresponding comments of two users, **Swisher Sweets** and **Backwoods**, on Instagram from 2018 to 2020. After that, annotate the each posts and comments, which are text format, with 3 different labels, which is whether they contain tobacco use warnings or not, whether they contain explicit sponsorship hashtags, and whether they contain ambiguous sponsorship hashtags.

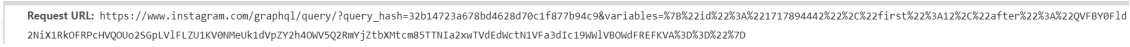
## Crawling Part

### Details

1. Before crawling, we first need to parse the URL of the corresponding user's instagram homepage. And URL of the user's instagram homepage is the URL 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.

### Problems

It is tough to crawl these data, since we have not applied the open-source tools, instead, we analyze the source code of the html, and the logic behind the request.

1. When we crawl, we actual do a request to the URL. And to request to the instagram, there should be a certification process. At this point, we should copy the cookie in the browser to the request header.
2. It is easy to reach the instagram's connection request limit, we don't record what the limit is, but 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 a time and retry to send request.

## Annotation Part

#### Explicit and Ambiguous Sponsorship Hashtags

Simple. We decide whether the sentence, among posts and comments, contains (#sponsored, #ad, #paid, #thanks, #sp, #spon, #ambassador or #collab).

#### Tobacco Use Warnings

In this part, we do not train a model, because we do not have enough labeled samples to build a training set. So we find some example sentences in the following pages.

- https://www.fda.gov/tobacco-products/labeling-and-warning-statements-tobacco-products/cigarette-labeling-and-health-warning-requirements
- https://www.cdc.gov/tobacco/data_statistics/sgr/2000/highlights/labels/index.htm

After that, we use sentence embedding to decide whether the posts and the comments are similar to example sentences, and the embedding vectors are downloaded from https://tfhub.dev/google/LaBSE/1.

We use the sample code from the page to do embedding. While experimenting, 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 find the new matrix line by line, if there is a value higher than the similarity threshold we set manually, we annotate that sentence contain tobacco use warnings.