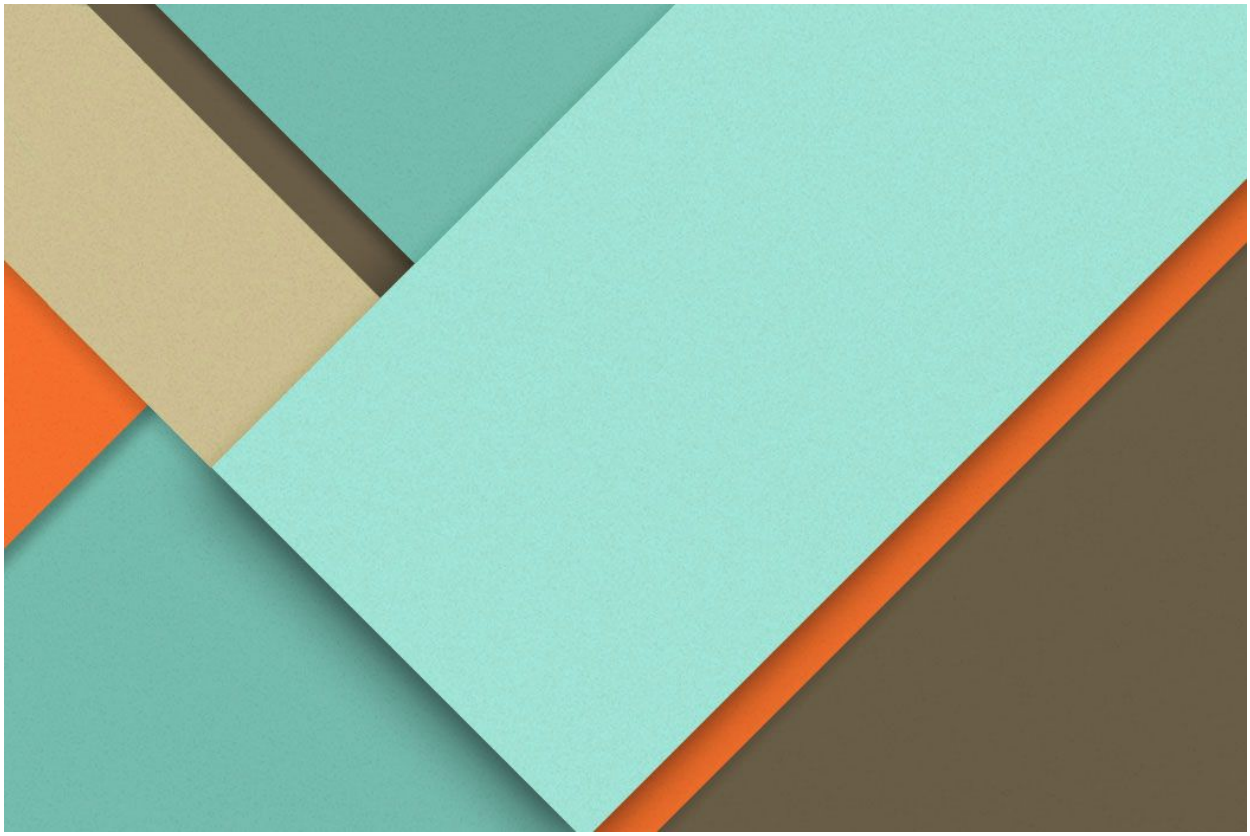


TrueReview

05.07.19

Demo link https://youtu.be/YAZkpULu_H8

Abstract

An iOS application designed to attempt to generate an accurate rating system based on Yelp Reviews

Introduction

For my final project, the main purpose of the project was to experiment with developing an a web service with Django and hosting it on AWS. I also wanted more experience in developing with swift. Finally, I wanted to see if it was possible to create a more accurate ranking system of businesses based on sentiment analysis of their Yelp reviews. Although I was not able to create an accurate rating system due to Yelp's API limitations, I learned a lot about creating apps and Web Services.

Specifications

The project uses Swift and Xcode for the frontend, along with the http library AlamoFire.

The Backend use Django as its base and is hosted on AWS Elastic Beanstalk.

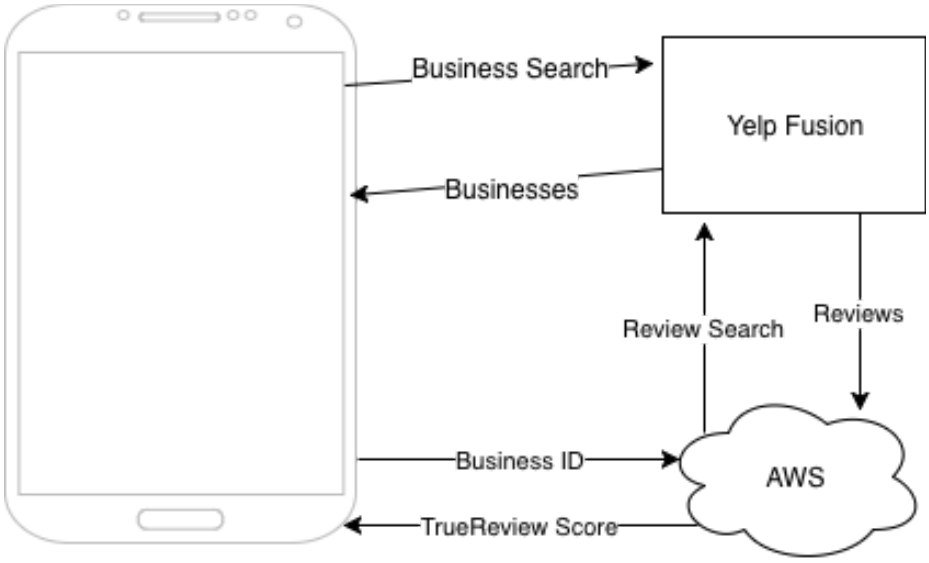
Front-end

The Front-end of TrueReview is written in Swift with the Xcode IDE. It uses data from Yelp to populate a MapKit view with location markers. To handle HTTP requests, AlamoFire was used to simplify requests and responses.

Back-end

The Back-end of the project uses Django as it's template. The Django project is hosted on AWS Elastic Beanstalk. The back-end is sent a restaurant id, which in turn is used to request Yelp reviews from that restaurant. Those reviews are then passed through an NLP python script to determine a rating out of 100.

Architecture



The client searches for a business or term. An array of businesses matching that term are returned to the client, and their locations are added to the map. When a user clicks on a location, a popup with simple information is pulled up. If the user clicks the information accessory, the Id of the selected business is sent to the AWS Elastic Beanstalk. The Django project hosted on the beanstalk takes the business ID and uses it to request a list of reviews about the business. These reviews are then passed to an NLP system within the Django project and scores are calculated. The score is then passed back to the client and displayed alongside various other information about the business.

Discussion

Overall, this project was very informative in many ways. First, I learned a lot about time management and scope management. I did not have the best idea of how long the project would take to complete, or which sections I would struggle with. For example, I thought that designing the iOS client would take me a few days, but I was able to complete most of the code in just a day and a half. On the other hand, I figured that creating a Django project and hosting it online would be straightforward but in reality, given my inexperience on the subject, it took me much longer than I had planned.

As for the effectiveness of the program, there are a few things that went wrong. First of which is the Yelp API. Yelp Fusion only allows access to 3 reviews per business, and 160

characters per review. This made for too small of a sample size to generate a good score. On top of that, the NLP model that I was using was improperly trained; as I found out after I presented, the model did not successfully train itself from the list of training material that was provided. In the future, the model can be properly trained and yelp can be scraped for reviews. This would decrease speed but increase accuracy exponentially.

Task/Description List

Task/step Description	Hours spent
Writing iOS Client	15
Learning about Django	10
Writing Django / python	25
Debugging	10
Total	60

References

Alamofire/Alamofire. (2019). Retrieved from https://github.com/Alamofire/Alamofire

Deploying a Django Application to Elastic Beanstalk - AWS Elastic Beanstalk. (2019). Retrieved from https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html

Django documentation | Django documentation | Django. (2019). Retrieved from https://docs.djangoproject.com/en/2.1/

Documentation - Yelp Fusion. (2019). Retrieved from https://www.yelp.com/developers/documentation/v3

TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation. (2019). Retrieved from https://textblob.readthedocs.io/en/dev/