Switzerland's Happy Places

Finding the places to be in in the country

Yassine Benyahia, Mohammed Hamza Sayah, Felix Bohm & Juraj Korcek

EPFL, Department of Computer Science, Switzerland

Abstract

Based on geolocated data from Twitter and Instagram we performed sentiment prediction using neural network models. Final goal of this project is to scrape the web daily and to run our algorithms over the data obtained. This will result in a daily map of interesting places in Switzerland.

Introduction

When arriving in a country, everything seems overwhelming at first. It is often hazardous to find places to go to, while locals might not always be around, aiding with advice. Sites like Trip Advisor offer a plattform for rating specific activities, without showing a larger picture of the region. Locals will rarely use these plattforms, but instead share their daily lives on plattforms such as Twitter or Instagram. By leveraging machine learning, the goal of this work is to find places that users of social media plattforms appreciate.

Twitter Sentiment

Preprocessing

- Original tweet
  - @Twitto MISSIN UUU!!!! SEE U 2MORO! pic.twitter.com/SbWh
- Lowercasing
  - @twitto missin uuu!!!! see u 2moro! pic.twitter.com/SbWh
- Mentions and links replacing
  - <user> missin uuu!!!! see u 2moro! <pic>
- Punctuations separation
  - <user> missin uuu ! ! ! ! see u 2moro ! <pic>
- Repetitions removing
  - <user> missin u ! ! ! ! see u 2moro ! <pic>
- Spell checking
  - <user> missing you ! ! ! ! see you tomorrow ! <pic>

Imbalance Issue

- The main problem we had with the dataset consisted in its imbalance :

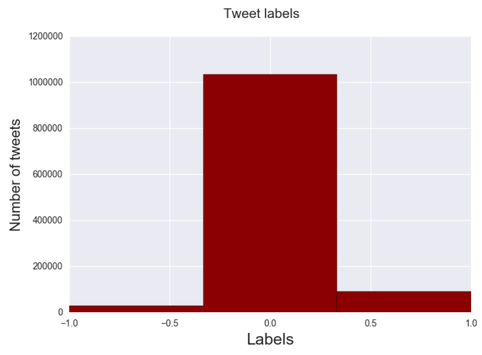

Figure 1: Imbalance Issue

- How to deal with it ?
- Under-sampling implies loss of information
- Putting neutral tweets apart from the analysis.

Results

- We used convolution network with GloVe word embeddings initialization and here are the results :

Class	Precision	Recall	F1-score
Pos	0.95	0.96	0.96
Neg	0.89	0.85	0.87

Table 1: Model With Smileys

- And for the model without smileys :

Class	Precision	Recall	F1-score
Pos	0.96	0.92	0.94
Neg	0.73	0.84	0.78

Table 2: Model Without Smileys

Neutral Tweets

- Here are the predicted probabilities histograms :

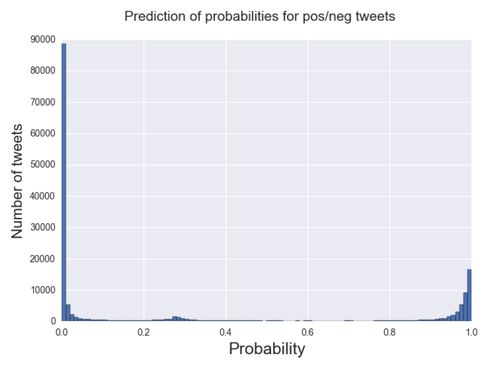

Figure 2: Prediction Probabilities over Positive/Negative Tweets

- To get rid of neutral tweets, it suffice to put severe thresholds over the predicted probabilities :

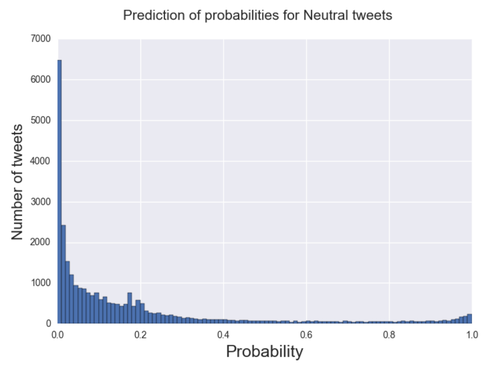

Figure 3: Prediction Probabilities over Neutral Tweets

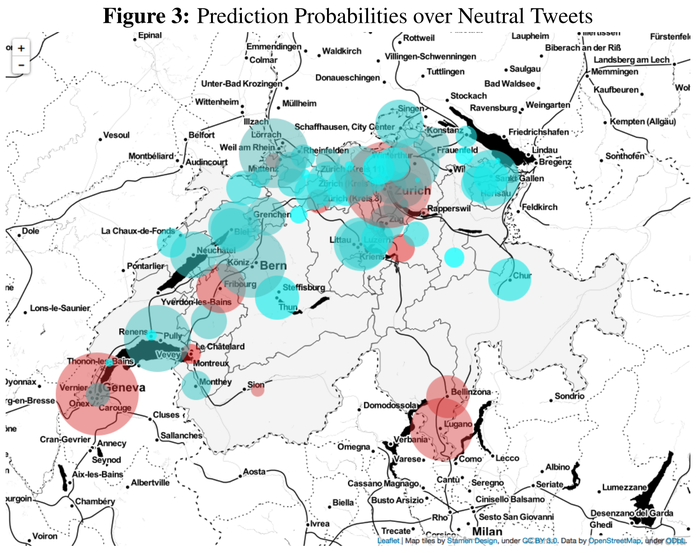

Figure 4: Predicted Values over Tweets Low - High scores

Instagram Sentiment

In this part of project our goal was to detect sentiment in images from Instagram and images only, i.e we have disregarded any textual description of the image. This was done to reduce the problem to only image classification part.

The name "Sentiment analysis" might be a bit misleading here. We do not want the sentiment per se, as in happy versus sad face occurring in the image. Instead, we consider an image with positive sentiment to be the one the depicts place that it is pleasant to walk through or by. The original plan was to do something in style of goodcitylife.com, i.e. classifying a sentiment of particular streets. However, or data does not contain location information with such a level of granularity and thus we have redefined our goal to be classifying sentiment (or pleasantness) of cities instead of streets in a city.

Strategy formulation

At first we have planned to train a new convolutional network using data from image-net.com. Image-net.com contains a big amount of images divided into categories by the object that is in the image. We want to download several categories, both positive (e.g. mountains, lakeside,...) and negative (graffiti, thrash) and train the network on them. However, we have realized that this would have taken too much time to train.

After some research, we have found out about Inception v3 - a convolutional neural network model pre-trained on all 1000 image categories of image-net.com. We have tested it on sample of our data and it performed well and therefore we have decided to include it in our pipeline.

By using Inception v3 CNN we have detected for every image 5 most probable objects contained in the image. Afterwards, we have extracted list of image categories that our dataset was assigned to. To each one of these categories we have assigned sentiment value, either -1 for negative, 0 for neutral, 1 for positive. For example, valley and lakeside are positive, while plate is neutral because we are not looking for the pictures of food.

In the end for each image we have multiplied confidence of top 5 objects in the image by the sentiment corresponding to them and summed it. That way we have obtained the sentiment of the image.

Preprocessing

The provided Instagram dataset includes images from January to October, offering a good insight into visualy interesting places throughout the seasons.

Due to the vagueness of a *happy* image, we decided to focus instead on classifying photos captured outdoors, which includes both land- and cityscapes. The assumption is that there is a high correlation between a photo being taken and the surrounding landscape being worth a visit. Our focus was therefore to filter out images of birthday cakes, parties and puppies.

The dataset included the Instagram ID of 3.84 million images, as well as links to several, presumably geographically distributed, download options for each of them.

It did not include geographical information, which was acquired by querying the API endpoints for Instagram's website. Due to many images not having assigned geographical information, as well as rate limits, the processed dataset shrunk to a total size of approx. $500,000$ images.

Technical discussion

For implementing the neural network we have decided to use TensorFlow library. However, we had problem running it on Spark cluster that was provided by EPFL. Thus, we have resorted to using Amazon AWS EC2 servers. In order to leverage the computational power of these servers we had to parallelize our computation well. Our pipeline consists of d download workers (processes) that download images from Instagram, c classification workers that classify images using TensorFlow library and Inception v3 model, 1 worker to write the results and 1 worker to report progress. To send jobs form one worker type to another we have used threadsafe queues.

While unsing a laptop we were able to get performance of only about 0.33 image dowloaded and processed per second, on the Amazon AWS EC2 server with 64 cores the performance obtained was 25 images dowloaded andprocessed per second. That is about 75-fold performance increase while the numer of cores has increased only 8 times. Full utilization of 20GBit ethernet connection played big role in this 75-fold increase.

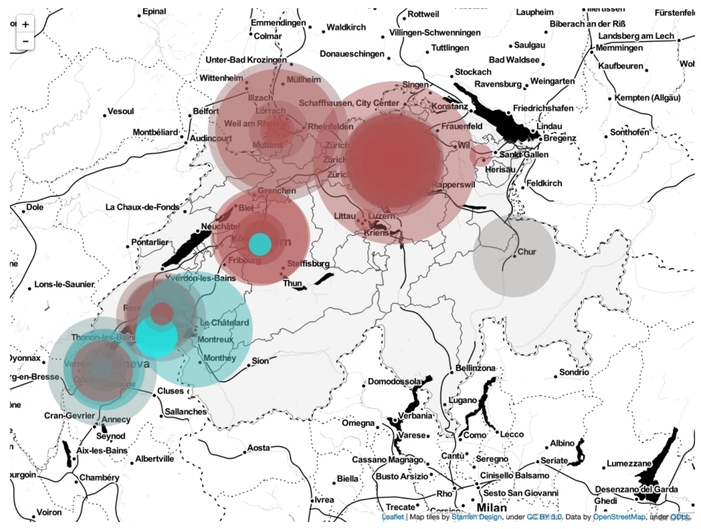

Figure 5: Image classification results for the spring dataset (January-April). Low - High scores

Conclusions

Both approaches led to different maps, highlighting the different focuses. As both approaches are viable when suggesting places for tourists to visit, a combination of both will serve most people best. Depending on their specific interests, a different balance can be struck to find the ideal spot for the next visit.