Final Project Report:

LDA Analysis,

Using Google Vision API and DUI

Jesus Aguas

Instructor: Rahul Singh

CSC 664, Multimedia Systems

12th May 2020

Contents

Introduction

The objective of this project is to analyze posts related with opiates and drug addiction and extract interesting insights that can be applied in future investigations in the drug domain.

The data used in this project will come from Reddit posts from the subreddit r/opiates. It will include posts with images and posts with images and text.

The content of this report is structured in 3 sections:

- Problem formulation
- Prior work
- Description of the work
- Experimental Evaluations
- Conclusions

The first section, **Problem formulation**, will introduce the topic, establishing the historical background and reviewing basic concepts fundamental to follow this paper.

The following section, **Prior work**, will explain in detail my prior work in this area and how is it related to the solution process this project.

Description of the work will provide an overview of the design and development of this project, as well as explaining in detail each stage of the project.

The **Experimental Evaluations** will show some experiments and its results describing the results and trying to explain the reason for those specific results.

Lastly, the **Conclusion** will conclude the report giving a brief but concise overview of the things learned and the conclusions extracted form the results.

Proposed problem formulation

Specifically, the objective of this project is to combine the power of two great tools, **Google Vision API** & **DUI** (Drug Use Insights) and use it to perform a LDA (Latent Dirichlet Allocation) analysis to identify topics of the data.

The stages of this problem are the following:

- Examine the data from the Reddit posts r/opiates
- Get the relevant data
- Perform a Google Vision API analysis
- Perform a DUI analysis
- Combine both in a single dataset
- Perform a LDA analysis on this dataset and identify topics

The reason why I think this is a good approach to achieve the objective of 'identify topics on opiates Reddit posts with images' is because the Google Vision API is the best tool to identify features on images. It uses a machine learning algorithm developed by Google that has been trained for years and has proven a good efficacy in most of the cases. And for analyzing the text of the posts I believe that the DUI tool is the best one I could use because it is oriented to the drugs analysis and I have experimentally concluded it works really well for most of the cases.

So my idea is combining these tools, Google Vision API to analyze the images and DUI to analyze the text, I can get relevant features of each post. And once I have that, I can treat each post as a document and perform a LDA analysis to get the topics that all these posts share.

Prior work in the area

In this section I will be talking about my prior work in this area, using these tools and how is it related to the solution process this project.

Semantics of media

During the course of CSC 874: Big Data I studied several papers about the emergent semantics of media and how this semantics could be extracted and treated. The fact of extracting semantics from media is related with this project as I am ultimately extracting topics from the media, which would correspond to semantic categories which these posts can have.

Google Vision API

I have already used this tool before, last semester, on CSC: 847: Cloud and Distributed Computing, where I created an application that could recognize the ingredients in the labels of the products that you can buy at a grocery store. This tool would use the Google Vision API to recognize the text in the images and find the ingredients of the food in order to evaluate if the product had any unhealthy ingredients.

From this experience I got a better understanding of this tool, how it has to be implemented and what were its strong features and weaknesses.

Machine Learning (LDA)

I am familiar with machine learning and the 'Scikit learning' library for python to execute Machine learning algorithms. It is a topic that I am really interested, specially doing clarification systems and generally in supervised learning, however I also have some experience in unsupervised learning doing Clustering on data. However I have never performed a LDA analysis before, it was Deeptanshu Jha who introduce it to me and showed me its potential.

Because of my background in machine learning I didn't have many problems understanding how it works.

Text Analysis

I have had many experiences with text analysis, I would say that my first one was last year on a Business Intelligence class were I performed a Text Analysis to build a Data Warehouse and used the SAP software to manage the data and extract conclusions. I have also done other courses like CSC 874: Big Data where the topic of text analysis was include on the lectures.

So I could apply some things that I already knew to this project, specially when preprocessing the text.

Description of the work

In this section I will provide an overview of the design and development of this project, as well as explaining in detail each stage of the project.

The stages of the project where the following:

- Data examination
- Data extraction
- Process data and combine it in a single dataset
- Perform a LDA analysis on this dataset and identify topics

Data examination

In this stage I spend hours looking at Reddit posts in r/opiates I went trough 500 image posts that I first store in a relational database (MySQL). While examining them I would annotate interesting insights about them and once I have seen enough to identify categories I start classifying them into categories and counting the number of posts in each category.

The results where not the desired, out of 600 posts, 81% (486 posts) where classified as memes, 8% (48 posts) where related with drug recovery, 6% (36 posts) where related with drug addiction, 5% (30 posts) where classified as other, or no relevant.

A few time later I discovered the reasons behind this distribution, and that is that pictures of drugs, pills and paraphernalia are not allowed on Reddit, and therefore they get removed.

Luckily, Deeptanshu Jha found a really interesting type of post that combined text and images, and those images were urls to other webpages where this kind of images were allowed. The result was that no memes were found in this type of posts.

So once it was clear what kind of data was interesting for the analysis, I proceeded to the next stage

Data extraction

In this stage I performed a formalized data extraction of the posts on Reddit, and I say ‘formalized’ because I had already extracted data form these posts for the analysis, however, in this case the schema of the database was clear.

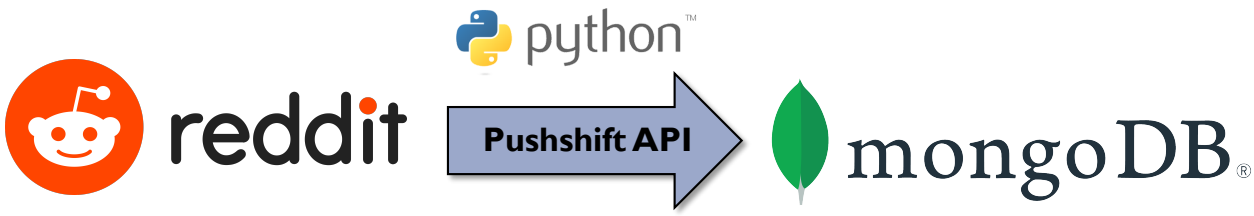
Key	Value	Type
(1) {_id : 87j6rd}	{ 10 fields }	Document
_id	87j6rd	String
author	piicklechiick	String
title	1 year ago today...	String
selftext	I finally made the decision to stop destroying my life and ch...	String
date	1522163610	Int32
num_comments	4	Int32
url	https://www.reddit.com/r/OpiatesRecovery/comments/87j6...	String
withtext	true	Bool
comments	I hope it works out for us too! I work for a property manage...	String
images	[1 elements]	Array
0	{ 7 fields }	Object
url	https://i.imgur.com/fb4SdE3.png	String
objects	[0 elements]	Array
labels	[4 elements]	Array
web_entities	[9 elements]	Array
faces	[0 elements]	Array
logos	[0 elements]	Array
text	UG 74%\n8:00 AM\nQuote of the day\nIt is in your moment...	String
(2) {_id : xapuh}	{ 10 fields }	Document
(3) {_id : 2kci8w}	{ 10 fields }	Document
(4) {_id : 5pu0lc}	{ 10 fields }	Document

The data was extracted using the Pushshift API, I followed the documentation to do the specific requests that I needed. [3]

Most of the posts that I used came from a list of posts that Deeptanshu Jha shared with me, after preprocessing this data the result was the ‘text posts.txt’ file that can be found on the submission folder.

While getting data from these posts, I performed the Google Vision API analysis, the python files to realize this task (*pushshift_image_only.py* and *pushshift_text.py*) can be found on the submission folder. This files all clearly commented so there is no problem to understand the code. [1]

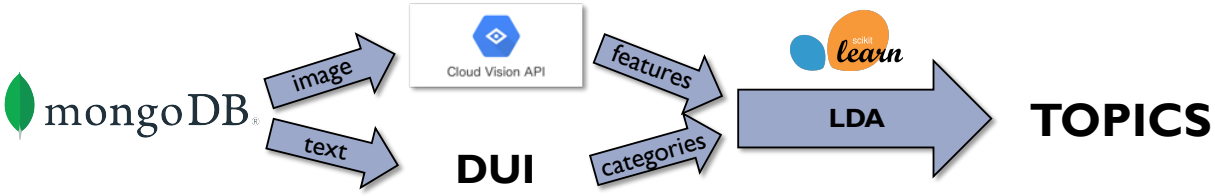
This next image shows a simple and clear representation of this stage:



Process data and combine it in a single dataset

In this stage I used the data that I stored on MongoDB on the last stage and I performed a DUI analysis with the title, text and comments of the posts, from the output of the DUI tool, I used the category names and the key words that belonged to those categories and I combined them with the Google Vision API features that I extracted before, all combined into a single file: *"data.txt"*. I used a python file to do this, which is called '*googleVision_DUI.py*'. Both of these files can be found on the submission folder.

So we have the *"data.txt"* file, which is the dataset that we will be using in the last stage, the LDA analysis. Each line in this dataset corresponds to a post (document) and it is the combination of the features found on the images by the Google Vision API and the categories and key words that the DUI tool identified in the title, text and comments of the post.



LDA Analysis

In this stage I used the dataset that I created on the last stage (*"data.txt"*) and I extracted all the documents in the file in a python list. Once I had all the documents I performed the LDA Analysis, which was pretty easy to implement thanks to the *"Scikit Learn*" library which already had functions that helped in developing this algorithm. [2]

The file that contains this algorithm and that is used to get the finals results is the "*LDA.py*" python file, which contains comments that clearly explain the process.

And finally we get to the results, that will be described and analyzed in the following section.

Experimental Evaluations

This section will show some experiments and its results describing the results and trying to explain the reason for those specific results.

During all this project I have been doing lots of experiments with the data, changing parameters and see how these affect the output. I would always end up using the model that worked better. However when doing this experiments I didn't thought about reporting them so now I don't have a way to track them.

However, I can talk about the most recent experiments I have done, and that directly affect the final result. These are related with changing parameters in the LDA algorithms to get different outcomes of the algorithms. I tried different number of topics and see how the algorithm would work. Another thing we have to take into account is that because of the nature of the results it cannot be stablished a quantitative evaluation of the results, and this evaluation can only be realized by inspecting the results.

No.topics = 5

```
LDA:
Topic 1:
opioids dope goods packaged drug_use_general food weed administration acquisition water
Topic 2:
person hair opioids meme internet weed shoe clothing hand drug_use_general
Topic 3:
art coffee illustration shirt sleeve arts water visual photography design
Topic 4:
leg skin close arm joint injection hand histamine human flesh
Topic 5:
goods packaged font black line meter logo text opioids white
```

In this case we can there is no much sense within the topics, the semantic meaning of the key words for each topic seem to be very distant form each other, this could be caused because of the number of topics established is too low that some topics are combined to meet the 5 topic requirement.

No.topics = 10

```
LDA:
Topic 1:
meme food internet image opioids coin cuisine rice person humour
Topic 2:
shoe spoon tableware cutlery utensil kitchen sneakers shoes metal fork
Topic 3:
shirt sleeve betty earrings paraphernalia clothing museum games legal outerwear
Topic 4:
histamine leg reaction filters micron knee injection human swelling ankle
Topic 5:
goods packaged font black meter opioids line product logo text
Topic 6:
water blue administration vehicle font wheel opioids text device packaged
Topic 7:
plant poppy flower water opium tree leaf botany opioids stem
Topic 8:
dog hair nail opioids grooming dope goods packaged drug_use_general person
Topic 9:
opioids dope drug_use_general weed administration person acquisition recovery_general hair finance
Topic 10:
wood floor flooring art design stain car hardwood sport jean
```

This case looks better than the last one, the topics seem to be better defined, for example, we can see that the first topic is related with internet memes, the second one is related with cutlery and utensils and so on.

No.topics = 15

```
LDA:
Topic 1:
toilet coin comedy graphics clinic constipation zoom opioids art clip
Topic 2:
spoon shoe cutlery tableware utensil kitchen sneakers shoes men footwear
Topic 3:
macro photography abdomen navel paraphernalia beige black wallpaper product legal
Topic 4:
morphine drug injection filters oxycodone reduction pharmaceutical harm wheel parallel
Topic 5:
black font line logo text photography brand white organism meter
Topic 6:
text device zoom legal friend night writing paraphernalia wine multimedia
Topic 7:
coffee emblem water appliance tea sea horizon extract filter drink
Topic 8:
hair person dope human goods packaged imgur black drug_use_general forehead
Topic 9:
person hair meme internet clothing opioids black dope drug_use_general photography
Topic 10:
wood floor flooring design stain car art hardwood sport jean
Topic 11:
dog breed shih companion tzu snout puppy canidae carnivore mammal
Topic 12:
spoon packaged goods blue flooring reaction person cop tableware hair
Topic 13:
plant poppy flower opium stem botany leaf screenshot flowering wildflower
Topic 14:
goods packaged opioids dope drug_use_general weed product food administration water
Topic 15:
leg arm skin joint close human flesh histamine reaction hand
```

Now we are trying with a big number of topics, which doesn't look bad at all, looks like the fact of having more topics allows a better distribution of the terms. For example, we can notice a very interesting topic, the topic number 4 is clearly related with drugs.

Conclusions

This project was really good for my own learning process, as all the problems and doubts that I solved made me more aware of the mistakes that I make and I believe that I have learned a methodology of working that would save me from many problems in the future.

One very remarkable is that I have learn to think things twice before doing them, as this can save you lot of useless time spent if you end up finding you started with a wrong conception of the problem.

For the results of the project, we can conclude that decrementing the number of topics makes all the terms more compacted in categories and therefore this topic are less precise, and incrementing the number of topics allows a better distributions of the terms among the topics. However the higher the number of topics the less useful is our study, as we want to reduce the number of defined categories to a reasonable number, we need to find a balance between the precision of the topic and the number of terms on each topic. Because we could have as many topics as terms, and that would be 100% precise but there would be no point.

Find this balance and the perfect number of topic is complicated, as it varies depending the data and the amount of data you have, and the huge reason why it is complicated to establish a perfect number of topics is because the result cannot be accurately quantized. So it is difficult to determine when you are improving the algorithm and when you are making it worse.

Acknowledgment

• First I want to thank the creators of the DUI tool: **Zachary Prince, Deeptanshu Jha and Rahul Singh**, who allowed me to use it for this project.

• I also want to thank **Deeptanshu Jha and Rahul Singh** for all the help and indications throughout the project, for solving all the doubts, questions and problems that I encountered during this project.

Without their help this project would not had been the same.

References

[1] **Google**, “Cloud Vision documentation”, https://cloud.google.com/vision/docs

[2] **Scikit Learn**, “Scikit-learn documentation”, https://devdocs.io/scikit_learn/

[3] **Baumgartner**, Jason M. *“Pushshift Documentation” Release 4.0,* https://readthedocs.org/projects/reddit-api/downloads/pdf/latest/