DOCUMENT : SENDIMENT ANALYSIS FOR MARKETING

PROBLEM DEFINITION:

Sentiment analysis, also referred to as opinion mining, is an approach to natural language processing identifies the emotional tone behind a body of text. This is a popular way for organizations to and categorize opinions about a product, service or idea.

Why use sentiment analysis

- Understand your audience
  No matter if you are a social media manager, a product marketer, a copywriter, or a performance marketing manager – the more insights you have from your audience – the better you'll be able to do your job.

- Analyse your reputation
  Learn all of that using sentiment analysis data. You'll be able to craft your communication better and strengthen your reputation. And better reputation means more recommendations and more clients.

- Compare with competitors
  Hard to say but you can analyse these numbers with your competitors or other brands from your segment and see how good your opinions are and where you still have the potential for improvement.

- Measure your marketing / PR efforts
  Sadly, not everything in marketing may be measured but sentiment analysis may be a nice addition to your KPIs. Especially public relations activities may be hard to analyse because they rarely generate customers directly.

- Detect potential crisis
  Every business has dissatisfied customers. And sometimes they have every right to be angry if you or someone from your team made a mistake. The fastest you react the less damage you'll take from such a situation.

- Scan influencers / contrahents
  You are about to start a new cooperation with a social media influencer? It may be beneficial but at the same time, it's not hard to imagine how it may backfire because of a big negative audience some influencers have.

DESIGN TTHINKING

Design thinking used in marketing

Design Thinking focuses on understanding customer needs to generate creative ideas and then proceeds through rapid prototyping steps. By using Design Thinking, you can base your decisions on what customers really want, instead of being guided solely by instinct or relying only on historical data.

- ❖ Empathize: Stepping yourself in the customer's shoes is the first step. What are their problems? Why would they choose your product/service to solve that problem? How would they come across you? Such are the essential questions for this step.

- ❖ Define: Once you think from your customer's/client's perspective, it is time to define the actual problem. Once you define the problem, your mind set changes to solving the problem rather than focusing on the problem.

- ❖ Ideate: Now that the problem is well in front of you, start looking for solutions. Brainstorm. Ask all the people you know. If needed, hire a focus group. Let us continue the previous example. Some ideas for the problem are changing the CTA button size, changing the CTA text, changing the heading of the page, etc.

- ❖ Prototype: The brainstorming session is over. Everyone has their ways to solve problems. Some are good, some are bad. How do you know which one will work? Answer: Prototype. Make a minimal viable solution and get feedback.

- ❖ Test: It is time to get it to the real audience. Depending on prototyping results, use the solution on the actual product/service/website/landing page/whatever.

CONCLUSION :

Sentiment analysis is a marketing tool that helps you examine the way people interact with a brand online. This method is more comprehensive than traditional online marketing tracking, which measures the number of online interactions that customers have with a brand, like comments and shares.

INNOVATION

Fine Tuning Pre-trained Model for Sentiment Analysis

Introduction

In this tutorial, I will be fine-tuning a Roberta model for the **Sentiment Analysis** problem.

The flow of the notebook

The notebook will be divided into separate sections to provide an organized walk-through of theprocess used. This process can be modified for individual use cases. The sections are:

1. Importing Python Libraries and preparing the environment
2. Importing and Pre-Processing the domain data
3. Preparing the Dataset and Data loader
4. fine-tuning
5. Fine Tuning the Model
6. Validating the Model Performance
7. Saving the model and artifacts for Inference in the Future

Technical Details

This script leverages multiple tools designed by other teams. Details of the tools used are below.Please ensure that these elements are present in your setup to successfully implement this script.

- Data: I will be using the dataset available at the Kaggle Competition
- ▪ I will be referring only to the first CSV file from the data dump: train's
- Language Model Used:

- The Roberta model was proposed in Roberta: A Robustly Optimized BERT PretrainingApproach by Yinan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Dangi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. It is based on Google's BERT model released in 2018.
- Blog-Post
- Research Paper
- Documentation for python
- Hardware Requirements:
- Python 3.6 and above
- Pytorch, Transformers, and All the stock Python ML Libraries
- GPU-enabled setup

Analysis Application with Chat GPT and Druid

Like many in the developer community, I have tried Chat GPT from Open AI. Over my IT career, I have worked in many positions including as a data scientist/data engineer. So, asI did my assessment of Chat GPT, I thought of ways to use the technology in practice. I have done sentiment analysis before using custom algorithms that I wrote, specific NLP (Natural Language Processing) libraries, and low-code platforms like Weka, Rapid Miner, and Data Robot. Why not do something similar using Chat GPT and combine it with a real-time analyticsdatabase like Apache Druid?

There are many benefits to combining a trained, NLP model with Apache Druid for sentimentanalysis. Modern models such as GPT-3 and GPT-4 are highly effective in understanding and processing natural language. They can better identify nuances and context, resulting in more accurate results. Sentiment analysis often requires processing large volumes of data, such as social media posts, reviews, or customer feedback. And then, aggregating and analysing those sentiments at scale can reveal even more insights and identify patterns and trends – all crucial forbusinesses that need to react quickly to changes in customer sentiment.

As this blog will show, the integration is relatively easy using technologies that are publiclyaccessible.

How to Train a ChatGPT Model for Sentiment Analysis

Let's learn how to train a sentiment analysis model with ChatGPT! ChatGPT is a greatchoice for this task because it can understand and interpret human language accurately.
Traditional sentiment analysis methods rely on keyword matching or rule-based systems, whichcan lead to incomplete or inaccurate results. But ChatGPT uses deep learning algorithms to analyze text at a deeper level, considering not only individual words but also the context in which they are used. So, let's follow these six steps to train ChatGPT with a sentiment analysis model.

Step 1: Data Collection: First, you need a large dataset of text data containing sentiment. This could be customer reviews, social media posts, or any other type of text that expresses sentiment.You can collect data from popular review sites like Yelp, Amazon, or TripAdvisor.

Step 2: Data Preprocessing: Once you have your dataset, it's time to preprocess it. This means cleaning and formatting the data so that it can be used to train the ChatGPT model. For instance,you can remove stop words, punctuation, and numbers and convert the text into a numerical format using one-hot encoding or word embeddings.

Step 3: Labeling Data: To train a supervised machine learning model, you need to label the data based on the sentiment it expresses. You can assign a binary label to each piece of text data, withpositive sentiment labeled as 1 and negative sentiment labeled as 0. For example, you can label apositive review of a restaurant as 1 and a negative review as 0.

Step 4: Training the Model: There are 2 approaches to train LLMs to answer questions –

1. Model fine-tuning, and

2. Context injection.

Dataset for sentiment analysis

IMDB Movie Reviews Dataset

This large movie dataset contains a collection of about 50,000 movie reviews from IMDB. In this dataset, only highly polarised reviews are being considered. The positive and negative reviews are even in number; however, the negative review has a score of ≤ 4 out of 10, and the positive review has a score of ≥ 7 out of 10.

Sentiment140

Sentiment140 is used to discover the sentiment of a brand product or even a topic on the social media platform Twitter. Rather than working on a keywords-based approach, which leverages high precision for lower recall, Sentiment140 works with classifiers built from machine learning algorithms. The Sentiment140 uses classification results for individual tweets along with the traditional surface that aggregates metrics. The Sentiment140 is used for brand management, polling, and planning a purchase.

Twitter US Airline Sentiment

This sentiment analysis dataset contains tweets since Feb 2015 about each of the major US airlines. Each tweet is classified as either positive, negative, or neutral. The included features include Twitter ID, sentiment confidence score, sentiments, negative reasons, airline name, retweet count, name, tweet text, tweet coordinates, date and time of the tweet, and the location of the tweet.

Amazon Product Data

Amazon product data is a subset of a large 142.8 million Amazon review dataset that was made available by Stanford professor, Julian McAuley. This sentiment analysis dataset contains reviews from May 1996 to July 2014. The dataset reviews include ratings, text, helpful votes, product descriptions, category information, price, brand, and image features.

Stanford Sentiment Treebank

This dataset contains just over 10,000 pieces of Stanford data from HTML files of Rotten Tomatoes. The sentiments are rated between 1 and 25, where 1 is the most negative and 25 is the most positive. The deep learning model by Stanford has been built on the representation of sentences based on sentence structure instead of just giving points based on positive and negative words.

For example:

The Interview was neither that funny nor that witty.

Even if there are words like funny and witty, the overall structure is a negative type.

Multi-Domain Sentiment Dataset

This dataset contains positive and negative files for thousands of Amazon products. Although the reviews are for older products, this data set is excellent to use. The data was derived from the Department of Computer Science at Johns Hopkins University.

The reviews contain ratings from 1 to 5 stars that can be converted to binary as needed

Sentiment Classification

This dataset described in the previous paragraph contains review comments on several movies. Comments in the dataset are already labeled as either positive or negative. The dataset contains the following two fields separated by a tab character.

1. text:- Actual review comment
2. sentiment:- Positive sentiments are labelled as 1 and negative sentiments are labelled as 0.

Now this article will discuss a few functions of preprocessing of text datasets.

Text Pre-Processing

Unlike structured data, features are not explicitly available in text data. Thus we need to use a process to extract features from the text data. One way is to consider each word as a feature and find a measure to capture whether a word exists or does not exist in a sentence. This is called the bag-of-words(BoW) model. That is each sentence is treated as a bag of words. Each sentence is called a document and the collection of all documents is called corpus.

This is a list of preprocessing functions that can be performed on text data such as:

1. Bag-of_words(BoW) Model
2. creating count vectors for the dataset
3. Displaying Document Vectors
4. Removing Low-Frequency Words
5. Removing Stop Words
6. Distribution of words Across Different sentiments

Bag-of_words(BoW) Model

The first step in creating a Bow Model is to create a dictionary of all the words used in the corpus. At this stage, we will not worry about grammar and only the occurrence of the words is captured. Then we will convert each document to a vector that represents words available in the documents. There are three ways to identify the importance of words in the BoW Model:

- Count Vector Model
- Term Frequency Vector Model
- Term Frequency-Inverse Document Frequency(TF-IDF) Model

Creating Count Vectors for Dataset

Each document in the dataset needs to be transformed into TF or TF-IDF vectors sklearn.feature_extraction.text module provides classes for creating both TF and TF-IDF vectors from text data. We will use CountVectorizer to create count vectors.

Displaying Document Vectors

To visualize the count vectors, we will convert this matrix into a dataframe and set the column names to the actual feature names.

Removing Low_Frequency Words

One of the challenges of dealing with text is the number of words of features available in the corpus is too large. The number of features could easily go over tens of thousands. The frequency of each feature or word can be analyzed using the histogram. To calculate the total occurrence of each feature or word, we will use the np. sum() method. The histogram showed that a large number of features have very rare occurrences.

```
 ABSA_df.isnull().sum()
print "Original:", ABSA_df.shape ABSA_dd = ABSA_df.drop_duplicates() dd =
ABSA_dd.reset_index(drop=True) print "Drop Dupicates:", dd.shape dd_dn =
dd.dropna()
df = dd_dn.reset_index(drop=True) print "Drop Nulls:", df.shape
```

Getting started with Text Preprocessing

Removal of Punctuations

Removal of stopwords

Removal of Frequent words

```
from collections import Counter cnt = Counter()
for text in df["text_wo_stop"].values: for word in text.split():
cnt[word] += 1 cnt.most_common(10)
Out[6]:
[('I', 1437),
('us', 752),
('DM', 514),
('help', 479),
('Please', 376),
('We', 338),
('Hi', 293),
('Thanks', 287),
('get', 279),
('please', 247)]
In [7]:
linkcode
FREQWORDS = set([w for (w, wc) in cnt.most_common(10)]) def
remove_freqwords(text):
"""custom function to remove the frequent words"""
```

```
return " ".join([word for word in str(text).split() if word not in FREQWORDS])
df["text_wo_stopfreq"] = df["text_wo_stop"].apply(lambda text:
remove_freqwords(text)) df.head()
```

Sentiment Analysis & NLP in Digital Marketing

Ever wondered how to tap into the thoughts and feelings of your target audience? ***Sentiment analysis*** powered by natural language processing ***(NLP) and Python*** is here to save the day! Let’s dive into the world of sentiment analysis and explore some awesome tools and techniques that’ll helpyou better understand your customers.

Why Sentiment Analysis Matters in Digital Marketing

In the digital age, understanding customer sentiment is crucial for effective marketing. Here’s why:

- ❖ Brand reputation management: Monitor customer feedback and address any issues promptly tomaintain a positive brand image.
- ❖ Campaign performance evaluation: Assess the public’s reaction to your marketing campaignsand make data-driven adjustments for better results.
- ❖ Competitor analysis: Gain insights into your competitors’ strengths and weaknesses byanalyzing their customer sentiment.

Getting Started with Sentiment Analysis in Python

Python offers various libraries for sentiment analysis and NLP, making it easy for digital marketers to dive into text data analysis. Here are some popular Python libraries for sentiment analysis:

- ***TextBlob***: A simple, beginner-friendly library that provides sentiment analysis functionality withjust a few lines of code.
- ***VADER***: A specialized library for sentiment analysis on social media data, which is particularlyuseful for digital marketers.
- ***NLTK***: A powerful library for NLP tasks, including sentiment analysis, with extensivedocumentation and resources.

Description

This Dataset is an updated version of the Amazon review dataset released in 2014. As in theprevious version, this dataset includes reviews (ratings, text, helpfulness votes), product

metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs). In addition, this version provides the following features:

More reviews:

The total number of reviews is 233.1 million (142.8 million in 2014).

Newer reviews:

Current data includes reviews in the range May 1996 - Oct 2018.

Metadata:

We have added transaction metadata for each review shown on the review page. Such information includes:

Product information, e.g. color (white or black), size (large or small), package type (hardcover or electronics), etc.

Product images that are taken after the user received the product.

Added more detailed metadata of the product landing page. Such detailed information includes:

Bullet-point descriptions under product title.

Technical details table (attribute-value pairs).

Similar products table.

More categories:

Includes 5 new product categories.

You can also download the review data from our previous datasets.

Amazon review (2014)

Amazon review (2013)

Citation

Please cite the following paper if you use the data in any way:

Justifying recommendations using distantly-labeled reviews and fined-grained aspects

Jianmo Ni, Jiacheng Li, Julian McAuley

Empirical Methods in Natural Language Processing (EMNLP), 2019 pdf

News

05/2021 We updated high resolution image urls to the metadata!

08/2020 We have updated the metadata and now it includes much less HTML/CSS code. Feel free to download the updated data!

Note

We provide a colab notebook that helps you parse and clean the data. For example:

Load the metadata (e.g. as JSON or DataFrame)

Check if title has HTML contents and filter them

We provide a collab that helps you find target products and obtain their reviews!

We appreciate any help or feedback to improve the quality of our dataset! Feel free to reach us at jin018@ucsd.edu if you meet any following questions:

Unparsed HTML contents

Duplicate items which have the same reviews

Directory

Files

Complete review data

Please only download these (large!) files if you really need them. We recommend using thesmaller datasets (i.e. k-core and CSV files) as shown in the next section.

raw review data (34gb) - all 233.1 million reviews

ratings only (6.7gb) - same as above, in csv form without reviews or metadata

5-core (14.3gb) - subset of the data in which all users and items have at least 5 reviews (75.26 million reviews)

Per-category data - the review and product metadata for each category.

To download the complete review data and the per-category files, the following links will direct you to enter a form. Please contact me if you can't get access to the form.

Data format

Format is one-review-per-line in json. See examples below for further help reading the data.

Sample review:

{ "image": ["https://images-na.ssl-images- amazon.com/images/I/71eG75FTJJL._SY88.jpg"], "overall": 5.0, "vote": "2", "verified": True, "reviewTime": "01 1, 2018", "reviewerID": "AUI6WTTT0QZYS", "asin": "5120053084", "style": { "Size:": "Large", "Color:": "Charcoal" }, "reviewerName": "Abbey", "reviewText": "I now have 4 of the 5 available colors ofthis shirt... ", "summary": "Comfy, flattering, discreet--highly recommended!", "unixReviewTime": 1514764800 }
{ "reviewerID": "A2SUAM1J3GNN3B", "asin": "0000013714", "reviewerName": "J. McDonald", "vote": 5, "style": { "Format:": "Hardcover" }, "reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think thebook was published for singing from more than playing from. Great purchase though!", "overall": 5.0, "summary": "Heavenly Highway Hymns", "unixReviewTime":1252800000, "reviewTime": "09 13, 2009" }

where

- `reviewerID` - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- `asin` - ID of the product, e.g. 0000013714
- `reviewerName` - name of the reviewer
- `vote` - helpful votes of the review
- `style` - a disctionary of the product metadata, e.g., "Format" is "Hardcover"
- `reviewText` - text of the review
- `overall` - rating of the product
- `summary` - summary of the review
- `unixReviewTime` - time of the review (unix time)
- `reviewTime` - time of the review (raw)
- `image` - images that users post after they have received the product

Metadata

Metadata includes descriptions, price, sales-rank, brand info, and co-purchasing links:

metadata (24gb) - metadata for 15.5 million products

Sample metadata:

{ "asin": "0000031852", "title": "Girls Ballet Tutu Zebra Hot Pink", "feature": ["Botiquecutie Trademark exclusive Brand", "Hot Pink Layered Zebra Print Tutu", "Fits girls up to a size 4T", "Hand wash / Line Dry", "Includes a Botiquecutie TMExclusive hair flower bow"], "description": "This tutu is great for dress up playfor your little ballerina. Botiquecute Trade Mark exclusive brand. Hot Pink Zebraprint tutu.", "price": 3.17, "imageURL": "http://ecx.images-amazon.com/images/I/51fAmVkTbyL._SY300_.jpg", "imageURLHighRes": "http://ecx.images-amazon.com/images/I/51fAmVkTbyL.jpg", "also_buy": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O", "0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S", "0000031895", "B003AVKOP2", "B003AVEU6G", "B003IEDM9Q", "B002R0FA24", "B00D23MC6W", "B00D2K0PA0", "B00538F5OK", "B00CEV86I6", "B002R0FABA", "B00D10CLVW", "B003AVNY6I", "B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C", "B008UBQZKU", "B00D103F8U", "B007R2RM8W"], "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y", "B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E", "B003AVKOP2", "B00D9C1WBM", "B00CEV8366", "B00CEUX0D8", "B0079ME3KU", "B00CEUWY8K", "B004FOEEHC", "0000031895", "B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2", "B00EM7KAG6", "B00AMQ17JA",

"B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y", "B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ", "B00538F5OK", "B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U", "B00CEUWUZC", "B00IJVASUE", "B00GOR07RE", "B00J2GTM0W", "B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G", "B008VV8NSQ", "B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S", "B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M", "B00EHAGZNA", "B0046W9T8C", "B00E79VW6Q", "B00D10CLVW", "B00B0AVO54", "B00E95LC8Q", "B00GOR92SO", "B007ZN5Y56", "B00AL2569W", "B00B608000", "B008F0SMUC",
"B00BFXLZ8M"], "salesRank": {"Toys & Games": 211836}, "brand": "Coxlures", "categories": [["Sports & Outdoors", "Other Sports", "Dance"]] }

where

Code

Reading the data

Data can be treated as python dictionary objects. A simple script to read any of the abovethe data is as follows:

```
def parse(path): g = gzip.open(path, 'r') for l in g: yield
json.loads(l)
```

Pandas data frame

This code reads the data into a pandas data frame:

```
import pandas as pd import gzip def parse(path): g = gzip.open(path, 'rb') for lin g: yield json.loads(l) def getDF(path): i = 0 df = {} for d in parse(path): df[i] = d i += 1 return pd.DataFrame.from_dict(df, orient='index') df = getDF('reviews_Video_Games.json.gz')
```

```
ratings = [] for review in parse("reviews_Video_Games.json.gz"): ratings.append(review['overall']) print sum(ratings) / len(ratings)
```