|  |
| --- |
| **(Autonomous Institute Affiliated to VTU)**  **Department of Information Science and Engineering** |
|  |
| A Seminar Report on |
| **Sentiment Analysis** |
| *Submitted in partial fulfillment of the CIE for the subject*  **Natural Language Processing(ISEA2)** |
| by |
| **Shrinidhi KR** |
| **1MS16IS094** |
|  |
| *Under the guidance*  *of* |
| **Rajeshwari S B** |
| Assistant Professor |
| Department of ISE, MSRIT |
| Bengaluru-560054 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| 1. | Introduction and Applications |  |
| 2. | Approach, Methodology and Algorithms |  |
| 3. | Sentiment Lexicon |  |
| 4. | Problems, Challenges and Issues |  |
| 5. | Simple Code for Sentiment Analysis |  |
|  |  |  |
|  |  |  |

**INTRODUCTION AND APPLICATIONS**

* Introduction

Sentiment Analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations.

Attempts to identify the opinion/sentiment that a person may hold towards an object or aspect.

It is also known as opinion mining, sentiment mining and opinion extraction.

The movie

was fabulous!

The movie

stars Mr. X

The movie

was horrible!

* Applications
* *Movie*: is this review positive or negative?
* *Products*: what do people think about the new iPhone?
* *Public sentiment*: how is consumer confidence? Is despair increasing?
* *Politics*: what do people think about this candidate or issue?
* *Prediction*: predict election outcomes or market trends from sentiment
* Motivation
* Knowing sentiment is a very natural ability of a human being.

Can a machine be trained to do it?

* SA aims at getting sentiment-related knowledge especially from the huge amount of information on the internet
* Can be generally used to understand opinion in a set of documents
* Components of Opinion and Levels of mining opinion
* Basic components of an opinion:
  + Opinion holder: The person or organization that holds a specific opinion on a particular object.
  + Object: on which an opinion is expressed
  + Opinion: a view, attitude, or appraisal on an object from an opinion holder.
* Levels of Opinion Task
  + At the sentence level
  + At the document (or review) level
  + At the feature level

**Scherer Typology of Affective States**

* **Emotion**: brief organically synchronized … evaluation of a major event
  + *angry, sad, joyful, fearful, ashamed, proud, elated*
* **Mood**: diffuse non-caused low-intensity long-duration change in subjective feeling
  + *cheerful, gloomy, irritable, listless, depressed, buoyant*
* **Interpersonal stances**: affective stance toward another person in a specific interaction
  + *friendly, flirtatious, distant, cold, warm, supportive, contemptuous*
* **Attitudes**: enduring, affectively colored beliefs, dispositions towards objects or persons
  + *liking, loving, hating, valuing, desiring*
* **Personality traits**: stable personality dispositions and typical behavior tendencies
  + *nervous, anxious, reckless, morose, hostile, jealous*

**Complexity and levels of Analysis**

* Simplest task:
  + Is the attitude of this text positive or negative?
* More complex:
  + Rank the attitude of this text from 1 to 5
* Advanced:
  + Detect the target, source, or complex attitude types

**APPROACH AND METHODOLOGY**

* Machine Learning

This approach needs

1. A good classier such as Naive Byes, Support Vector Machine, etc

2. A training set for each class

* Natural Language Processing

This approach uses Semantics to understand the language. Major tasks in NLP that helps in extracting sentiment from a sentence:

1. Extracting part of the sentence that reflects the sentiment

2. Understanding the structure of the sentence

3. Different tools which help process the textual data

**ALGORITHMS**

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as:

* **Rule-based** systems that perform sentiment analysis based on a set of manually crafted rules.
* **Automatic** systems that rely on machine learning techniques to learn from data.
* **Hybrid** systems that combine both rule based and automatic approaches.

**Rule-based Approaches**

Usually, rule-based approaches define a set of rules in some kind of scripting language that identify subjectivity, polarity, or the subject of an opinion.

The rules may use a variety of inputs, such as the following:

* Classic NLP techniques like *stemming*, *tokenization*, *part of speech tagging* and *parsing*.
* Other resources, such as lexicons (i.e. lists of words and expressions).

**Automatic Approaches**

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. The sentiment analysis task is usually modeled as a classification problem where a classifier is fed with a text and returns the corresponding category, e.g. positive, negative, or neutral (in case polarity analysis is being performed).

* **The Training and Prediction Processes**

In the training process (a), our model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the test samples used for training. The feature extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g. positive, negative, or neutral) are fed into the machine learning algorithm to generate a model.

* **Feature Extraction from Text**

The first step in a machine learning text classifier is to transform the text into a numerical representation, usually a vector. Usually, each component of the vector represents the frequency of a word or expression in a predefined dictionary (e.g. a lexicon of polarized words). This process is known as feature extraction or text vectorization and the classical approach has been bag-of-words or bag-of-ngrams with their frequency.

* **Classification Algorithms**

The classification step usually involves a statistical model like Naïve Bayes, Logistic Regression, Support Vector Machines, or Neural Networks

**Hybrid Approaches**

The concept of hybrid methods is very intuitive: just combine the best of both worlds, the rule-based and the automatic ones. Usually, by combining both approaches, the methods can improve accuracy and precision.

**SENTIMENT LEXICON**

* SentiWordNet
* Lexical resource for sentiment analysis
* Built on the top of WordNet synsets
* Attaches sentiment-related information with synsets
  + Home page: <http://sentiwordnet.isti.cnr.it/>
  + All WordNet synsets (set/group of synonyms) automatically annotated for degrees of positivity, negativity, and neutrality/objectiveness
  + [estimable(J,3)] “may be computed or estimated”

Pos 0 Neg 0 Obj 1

* + [estimable(J,1)] “deserving of respect or high regard”

Pos .75 Neg 0 Obj .25

Other sentiment lexicons are:

* MPQA
* Opinion Lexicon
* General Inquirer
* LIWC

**PROBLEMS, CHALLENGES AND ISSUES**

Most of the work in sentiment analysis in recent years has been around developing more accurate sentiment classifiers by dealing with some of the main challenges and limitations in the field.

* **Subjectivity and Tone**

The detection of subjective and objective texts is just as important as analyzing their tone. In fact, so called objective texts do not contain explicit sentiments. Say, for example, you intend to analyze the sentiment of the following two texts:

The package is nice.

The package is red.

Most people would say that sentiment is positive for the first one and neutral for the second one, right? All predicates (adjectives, verbs, and some nouns) should not be treated the same with respect to how they create sentiment. In the examples above, nice is more subjective than red.

* **Context and Polarity**

All utterances are uttered at some point in time, in some place, by and to some people, you get the point. All utterances are uttered in context. Analyzing sentiment without context gets pretty difficult. However, machines cannot learn about contexts if they are not mentioned explicitly. One of the problems that arise from context is changes in [polarity](https://en.wikipedia.org/wiki/Affirmation_and_negation). Look at the following responses to a survey:

Everything of it.

Absolutely nothing!

* **Irony and Sarcasm**

Differences between literal and intended meaning (i.e. irony) and the more insulting or ridiculizing version of irony (i.e. sarcasm) usually change positive sentiment into negative whereas negative or neutral sentiment might be changed to positive. However, detecting irony or sarcasm takes a good deal of analysis of the context in which the texts are produced and, therefore, are really difficult to detect automatically.

For example, look at some possible answers to the question Have you had a nice customer experience with us? below.

Yeah. Sure.

Not one, but many!

What sentiment would you assign to the responses above? Probably, you have listened to the first response so many times, you would have said negative, right? The problem is there is no textual cue that will make a machine learn that negative sentiment since most often, yeah and sure belong to positive or neutral texts. How about the second response? In this context, sentiment is positive, but we’re sure you can come up with many different contexts in which the same response can express negative sentiment.

* **Comparisons**

How to treat comparisons in sentiment analysis is another challenge worth tackling. Look at the texts below:

This product is second to none.

This is better than old tools.

This is better than nothing.

There are some comparisons like the first one above that do not need any contextual clues in order to be classified correctly.

The second and third texts are a little more difficult to classify, though. Would you classify them as neutral or positive? Probably, you are more likely to choose positive for the second one and neutral for the third, right? Once again, [context](https://monkeylearn.com/sentiment-analysis/#context-and-polarity) can make a difference. For example, if the old tools the second text talks about were considered useless in context, then the second text turns out to be pretty similar to the third text. However, if no context is provided, these texts feel different.

* **Emojis**

There are two types of emojis according to [Guibon et al.](https://hal-amu.archives-ouvertes.fr/hal-01529708/document). Western emojis (e.g. :D) are encoded in only one character or in a combination of a couple of them whereas Eastern emojis (e.g. ¯ \ \_ (ツ) \_ / ¯) are a longer combination of characters of a vertical nature. Particularly in tweets, emojis play a role in the sentiment of texts.

* **Defining Neutral**

Defining what we mean by neutral is another challenge to tackle in order to perform accurate sentiment analysis. As in all classification problems, defining your categories -and, in this case, the neutral tag- is one of the most important parts of the problem. What you mean by neutral, positive, or negative does matter when you train sentiment analysis models. Since tagging data requires that tagging criteria be consistent, a good definition of the problem is a must.

* **Sentiment Tokenization Issues**
* Deal with HTML and XML markup
* Twitter mark-up (names, hash tags)
* Capitalization (preserve for

words in all caps)

* Phone numbers, dates
* Emoticons
* Useful code:
  + [Christopher Potts sentiment tokenizer](http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py)
  + [Brendan O’Connor twitter tokenizer](http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py)
* **Extracting Features for Sentiment Classification**

How to handle negation

I **didn’t** like this movie vs I really like this movie

Add NOT\_ to every word between negation and following punctuation:

* didn’t like this movie , but I
* didn’t NOT\_like NOT\_this NOT\_movie but I

**SIMPLE CODE FOR SENTIMENT ANALYSIS USING LEXICON APPROACH**

import nltk

import csv

import numpy as np

negative = []

with open("words\_negative.csv", "r") as file:

reader = csv.reader(file)

for row in reader:

negative.append(row)

print(negative[:10])

positive = []

with open("words\_positive.csv", "r") as file:

reader = csv.reader(file)

for row in reader:

positive.append(row)

print(positive[:10])

def sentiment(text):

temp = []

text\_sent = nltk.sent\_tokenize(text)

for sentence in text\_sent:

n\_count = 0

p\_count = 0

sent\_words = nltk.word\_tokenize(sentence)

for word in sent\_words:

word=word.lower()

for item in positive:

if(word == item[0]):

p\_count +=1

for item in negative:

if(word == item[0]):

n\_count +=1

if(p\_count > 0 and p\_count > n\_count ):

print( "+ve : " + sentence)

temp.append(1)

elif(n\_count > 0 and n\_count>p\_count):

print( "-ve : " + sentence)

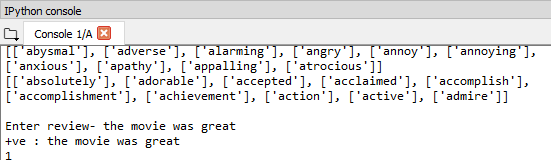
temp.append(-1)

else:

print ("neutral : " + sentence)

temp.append(0)

return sum(temp)

****print(sentiment(input("Enter review- ")))

****

**BIBLIOGRAPHY**

* <https://web.stanford.edu/class/cs124/lec/sentiment.pptx>
* <https://nptel.ac.in/courses/106105158/61>
* <https://monkeylearn.com/sentiment-analysis/>

**THANK YOU**