SENTIMENT ANALYSIS

Presented By:

Himanshu Patel , Akarsh Mathur, Harshit Jain and Kunal Sahu

0126IT181027, 0126IT181003, 0126IT181026, 0126IT181035

INFORMATION TECHNOLOGY 6TH SEM "A"

What is it??
- Input – raw text over some topic
- Output – opinion (+ve, -ve or neutral)
- Its is hard – why???

- determines the opinion on overall text rather than just subject of the topic
-- lets understand the problem

Rise of blogs, forums …
- Web 2.0 is commonly associated with web applications that facilitate interactive information sharing, interoperability, user-centered design, and collaboration on the World Wide Web – (source : Wikipedia)

Objective

- The project mainly deals analysis of product reviews from various ecommerce websites.
- Our Main objective is to get the overall polarity of reviews for a particular product.

Goals and non-goals

- Goals
  - Understand the basic ideas of sentiment analysis
  - Understand how computer-scientist text miners approach "sentiment" and "opinion"
  - Time permitting: Learn how different disciplines view these two concepts
  - Learn about some pitfalls and encourage a critical view
  - Get your hands on some tools and real data
    - Since this field is more involved than basic text mining, we will remain at a high level
  - Have pointers for inquiring and going further
- Non-goals (selection)
  - the statistical background of methods
  - A comprehensive overview of the state-of-the-art of sentiment analysis methods
    - (See the surveys in the references for this)
  - A comprehensive overview of the state-of-the-art of sentiment analysis applications in the digital humanities or social or behavioural sciences

DATA SETS

IMDb: an online database of information related to films, television programs, home videos, video games, and streaming content online — including cast, production crew and personal biographies, plot summaries, trivia, fan and critical reviews, and ratings.

1. Introduction and Importing Data

In this article, I will be using the IMDB movie reviews dataset for this study. The dataset contains 50,000 reviews — 25,000 positive and 25,000 negative reviews. An example of a review can be seen in *Fig 1*, where a user gave a 10/10 rating and a written review for the Oscar-winning movie Parasite (2020).

Figure 1

One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. The...	positive
A wonderful little production. The filming technique is very unassuming- very old-time-B...	positive
I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air con...	positive
Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his par...	negative

The IMDB dataset

In the `keras.datasets` module, we find the IMDB dataset:

- At least 7 out of 10 stars => positive (label=1)
- At most 4 out of 10 stars => negative (label=0)

An IMDB user review giving a positive rating on the movie Parasite (2020)

What is an opinion?

- "The fact is ..." and similar expressions are highly correlated with subjectivity (Riloff and Wiebe, 2003)

opinion (əˈpɪnjən)

n

1. judgment or belief not founded on certainty or proof

...

3. evaluation, impression, or estimation of the value or worth of a person or thing

...

[via Old French from Latin *opīniō* belief, from *opīnārī* to think]

Opinion orientation

- Start from lexicon
- E.g. dictionary SentiWordNet
- Assign +1/-1 to opinion words, change according to valence shifters (e.g. negation: *not* etc.)
- *But* clauses ("the pictures are good, but the battery life ...")
- Dictionary-based: Use semantic relations (e.g. synonyms, antonyms)
  - S: (adj) **great**, outstanding (of major significance or importance) *"a great work of art"; "Einstein was one of the outstanding figures of the 20th centurey"*
  - S: (adj) **great** (remarkable or out of the ordinary in degree or magnitude or effect) *"a great crisis"; "had a great stake in the outcome"*
  - S: (adj) bang-up, bully, corking, cracking, dandy, **great**, groovy, keen, neat, nifty, not bad, peachy, slap-up, swell, smashing, old (very good) *"he did a bully job"; "a neat sports car"; "had a great time at the party"; "you look simply smashing"; "we had a grand old time"*
- Corpus-based:
  - learn from labelled examples
  - Disadvantage: need these (expensive!)
  - Advantage: domain dependence

The unit of analysis

- community
- another person
- user / author
- document
- sentence or clause
- aspect (e.g. product feature)

"What makes people happy" example

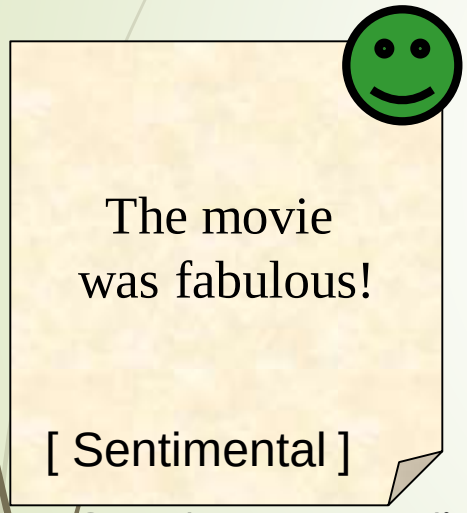
Phone example

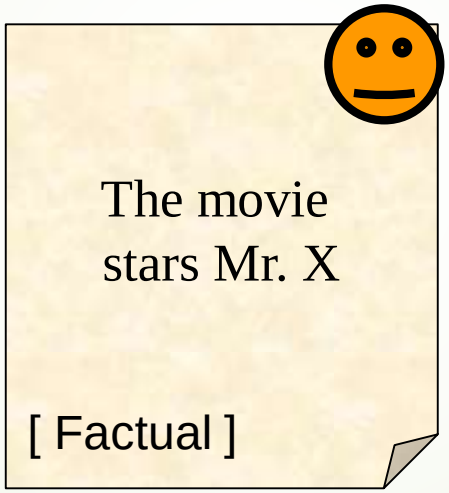
Find only the aspects belonging to the high-level object

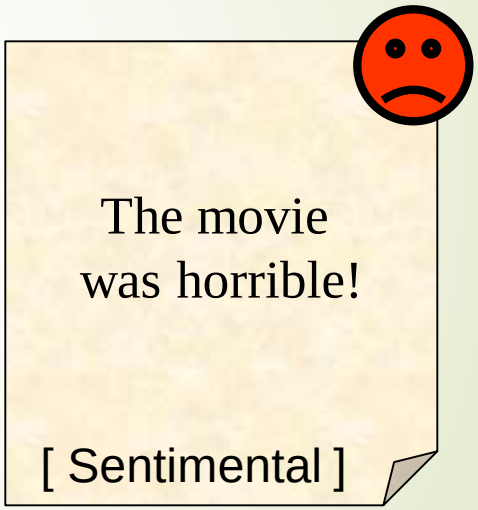
- Basic idea: POS and co-occurrence
  - find frequent nouns / noun phrases
  - find the opinion words associated with them (from a dictionary: e.g. for positive *good*, *clear*, *amazing*)
  - Find infrequent nouns co-occurring with these opinion words
  - BUT: may find opinions on aspects of other things
- Improvements on the basic method exist

Sentiment Analysis

- Identify the orientation of opinion in a piece of text







- Can be generalized to a wider set of emotions

References: google images

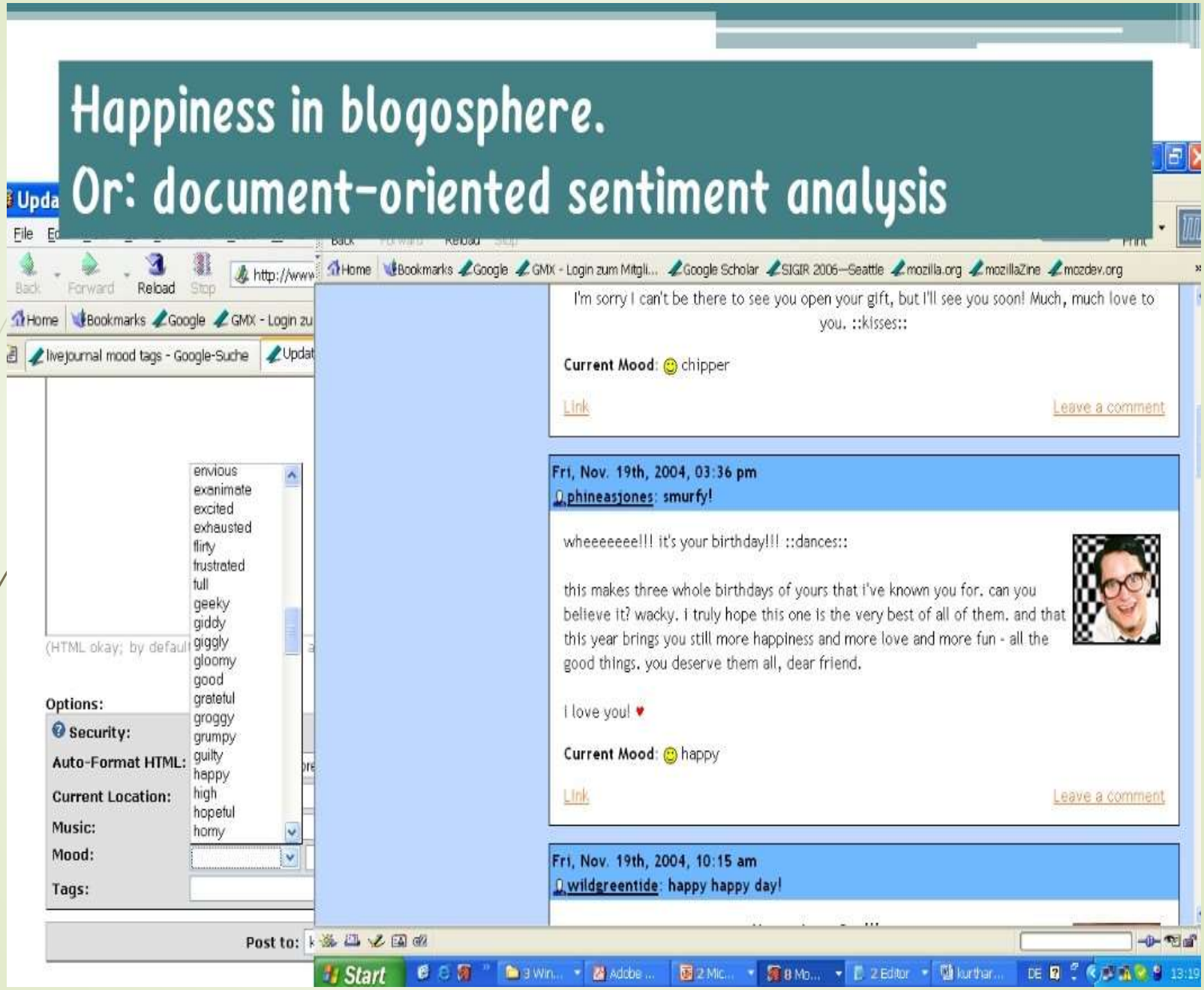
Sentiment Analysis

- *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

Sentiment analysis has many other names

- Opinion extraction
- Opinion mining
- Sentiment mining
- Subjectivity analysis

Happiness in blogosphere.
Or: document-oriented sentiment analysis



Why analyse text?

- Texts are a source of information not commonly used in official statistics
- Potential applications are, *automatically*:
  - Classify answers to open questions
  - Code description of jobs/educations/products
  - Identify activity code of companies from web site text
  - Detailed product identification from descriptions on web sites
  - Classify cause of death from medical reports
  - Sentiment analysis of messages
  - ...

Google Product Search

HP Officejet 6500A Plus e-All-in-One Color Ink-jet - Fax / copier / printer / scanner
$89 online, **$100** nearby ★★★★☆ 377 reviews
September 2010 - Printer - HP - Inkjet - Office - Copier - Color - Scanner - Fax - 250 sh

Reviews

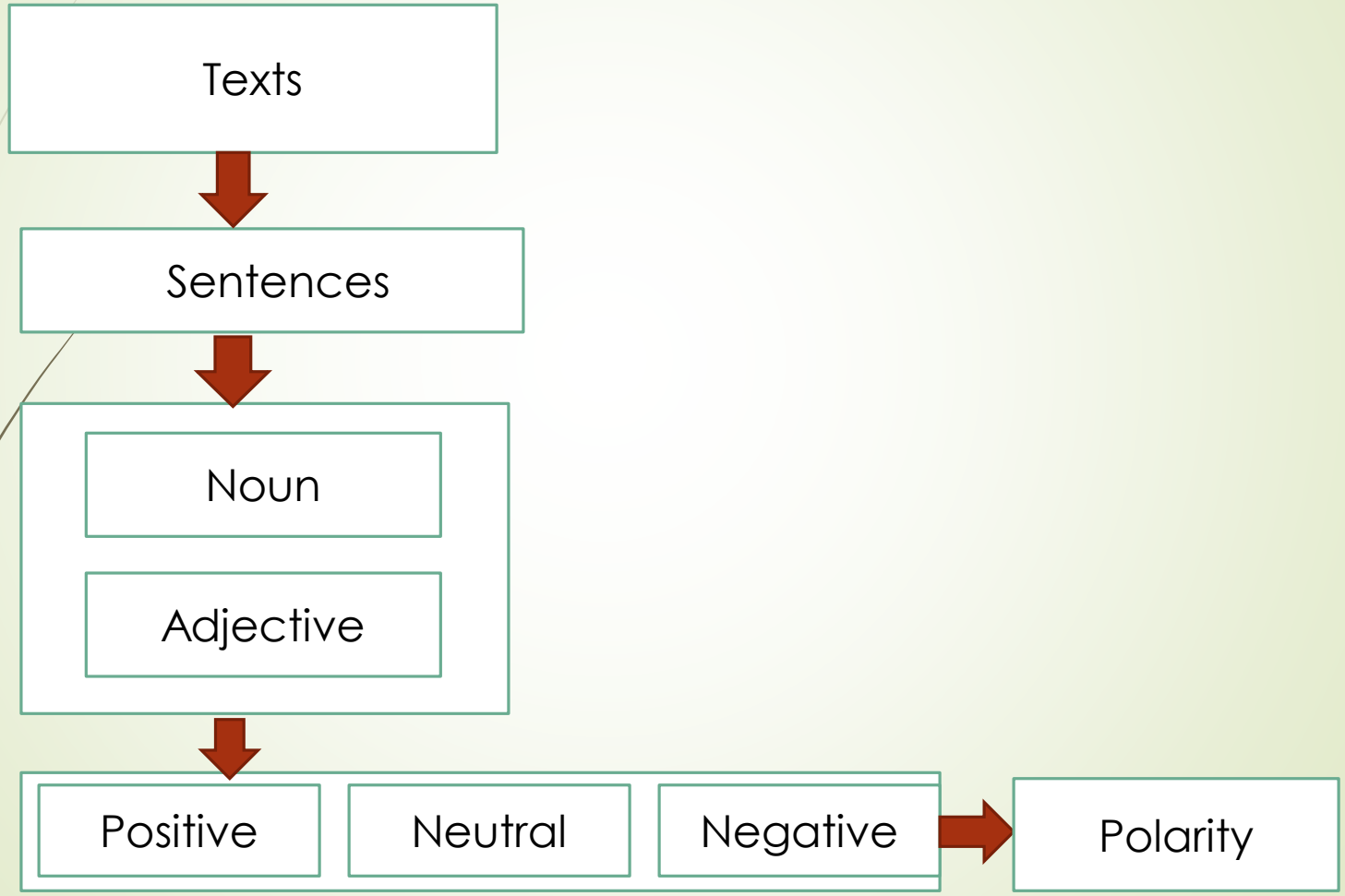
Summary - Based on 377 reviews

1 star	2	3	4 stars	5 stars

What people are saying

ease of use	"This was very easy to setup to four computers."
value	"Appreciate good quality at a fair price."
setup	"Overall pretty easy setup."
customer service	"I DO like honest tech support people."
size	"Pretty Paper weight."
mode	"Photos were fair on the high quality mode."
colors	"Full color prints came out with great quality."

Case Diagram



Meet sentiment analysis (1) (buzzilions.com)

4. On Stage MA100 - Screw Adapter + Compare

4.5 read 22 reviews

"Perfect" – Juanpianoman

"Super Useful Adapter Hard to Find!" – Mike O.

"Good for attaching to boom pole, BU..." – Russell Rules

starting at 1,95 USD

5. Verbatim Forecast Bimini 4 Piece Value Set + Compare

2.6 read 12 reviews

"I bought this luggage set ..." – RickHawaii

"bum wheels" – bumed1

"I purchase the identical ..." – whoopingcough

starting at 1,99 USD

6. Pro-Co Sound Rack Screws and Washers + Compare

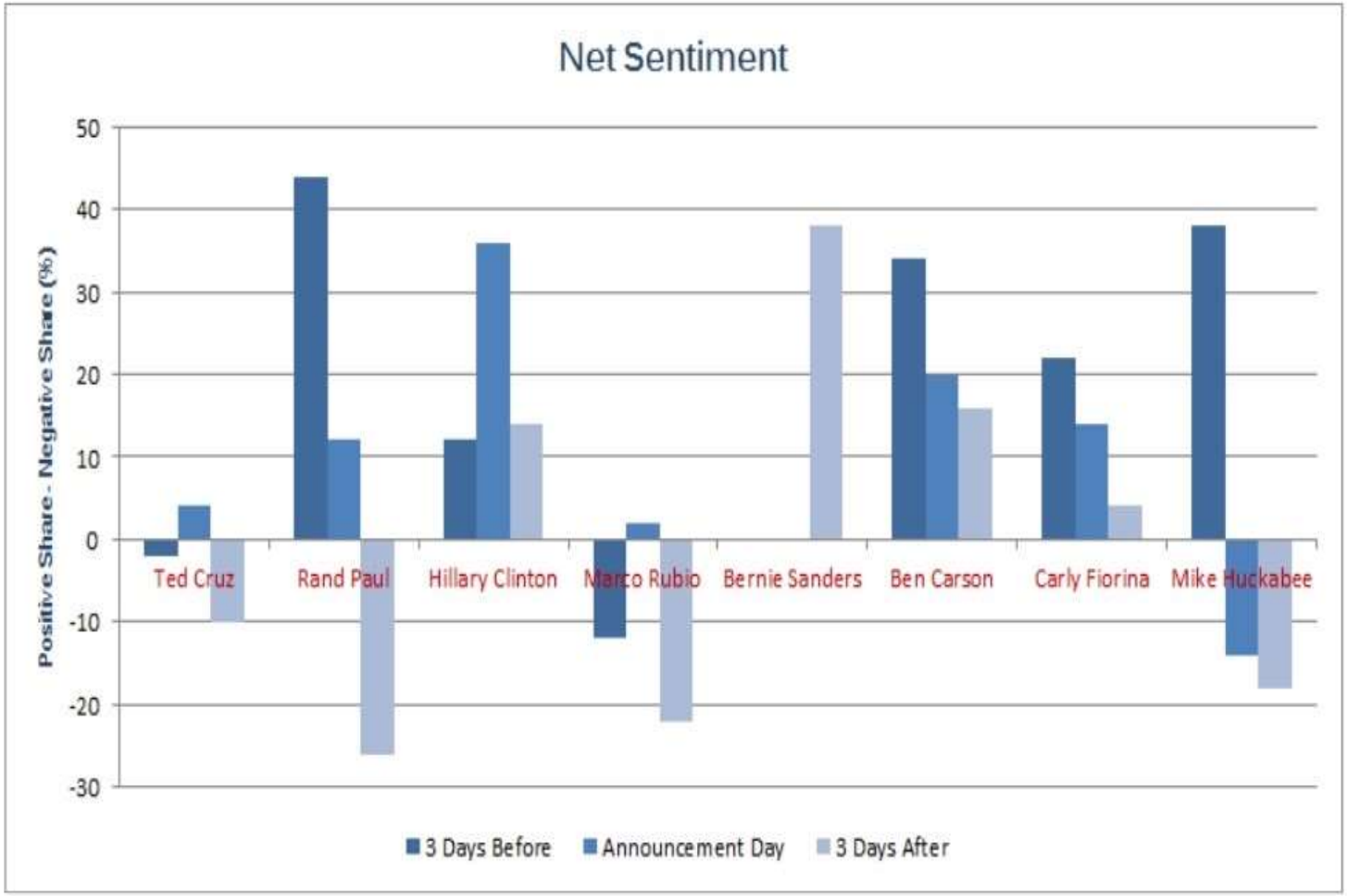
5.0 read 2 reviews

"What can you say "they are screws"" – JM

"hard to mess up a screw" – Gary

starting at 2,50 USD

Meet sentiment analysis (3)

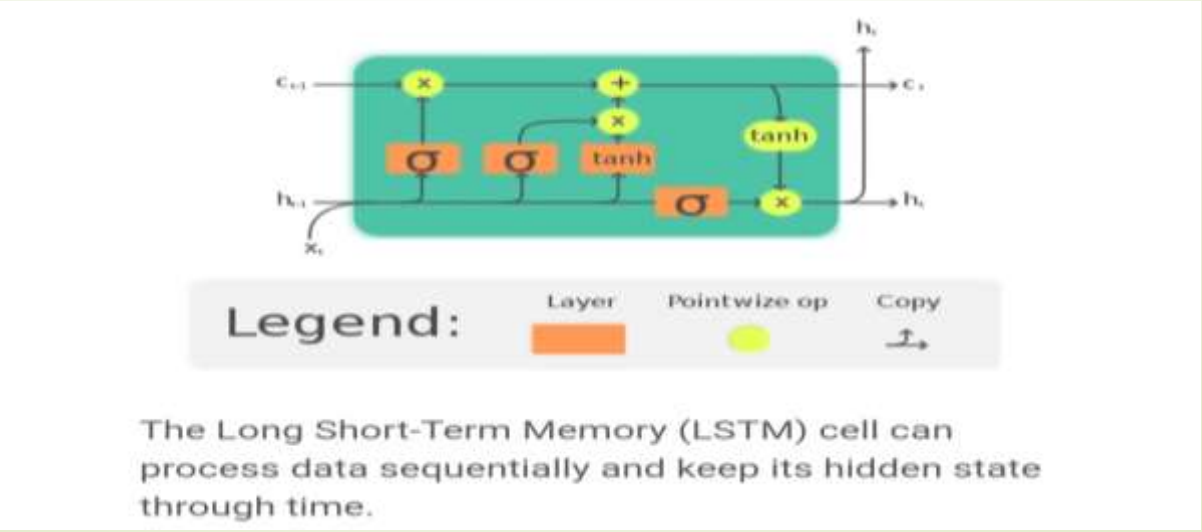


Areas To Be covered for Sentiment Analysis

- Long Short-Term Memory (LSTM)
- Recurrent Neural Networks
- Word Embeddings
- Convolutional Neural Networks
- DropOut
- Padding
- TensorFlow.Keras
- Average Pooling and Pooling Layers
- Adam Optimizer

Long Short-Term Memory (LSTM)

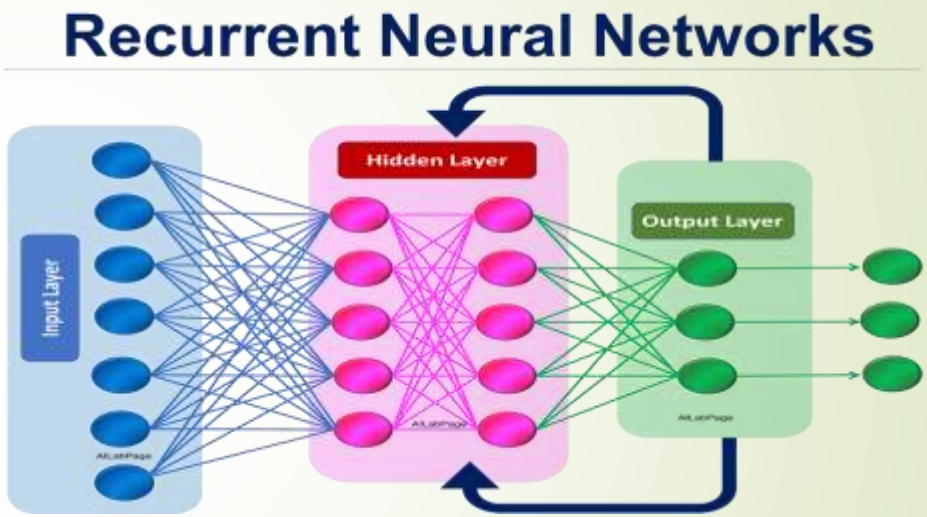
Long short-term memory (**LSTM**) is an artificial **recurrent neural network** (RNN) architecture used in the field of **deep learning**. Unlike standard **feedforward neural networks**, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected **handwriting recognition**, **speech recognition** and anomaly detection in network traffic or IDSs (intrusion detection systems).



The Long Short-Term Memory (LSTM) cell can process data sequentially and keep its hidden state through time.

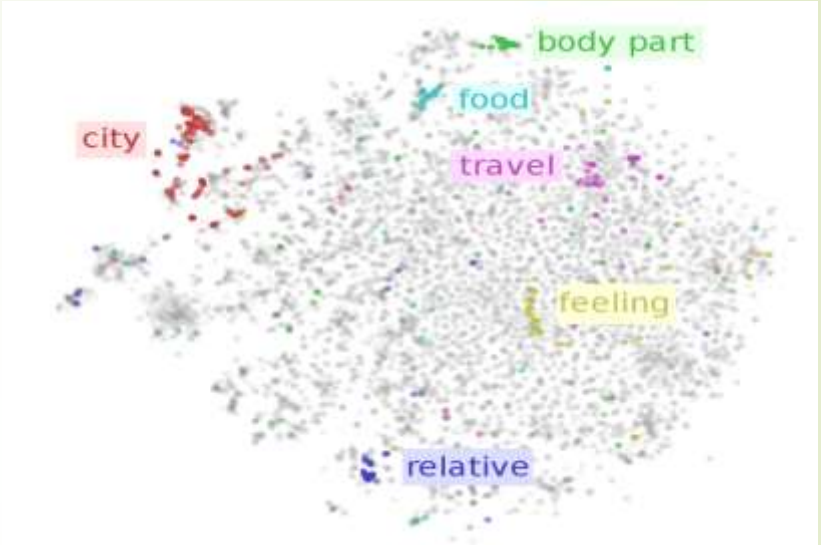
Recurrent Neural Networks

A **recurrent neural network** (**RNN**) is a class of **artificial neural networks** where connections between nodes form a **directed graph** along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from **feedforward neural networks**, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected **handwriting recognition** or **speech recognition**.



Word Embedding

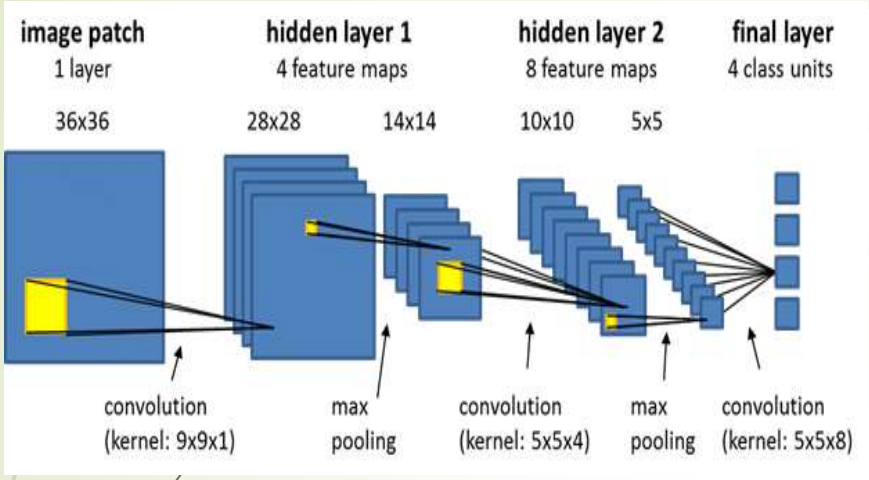
- In **natural language processing** (NLP), **Word embedding** is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning. Word embeddings can be obtained using a set of **language modeling** and **feature learning** techniques where words or phrases from the vocabulary are mapped to **vectors** of **real numbers**.



Methods to generate this mapping include **neural networks**, **dimensionality reduction** on the word **co-occurrence matrix**, probabilistic models, explainable knowledge base method, and explicit representation in terms of the context in which words appear.

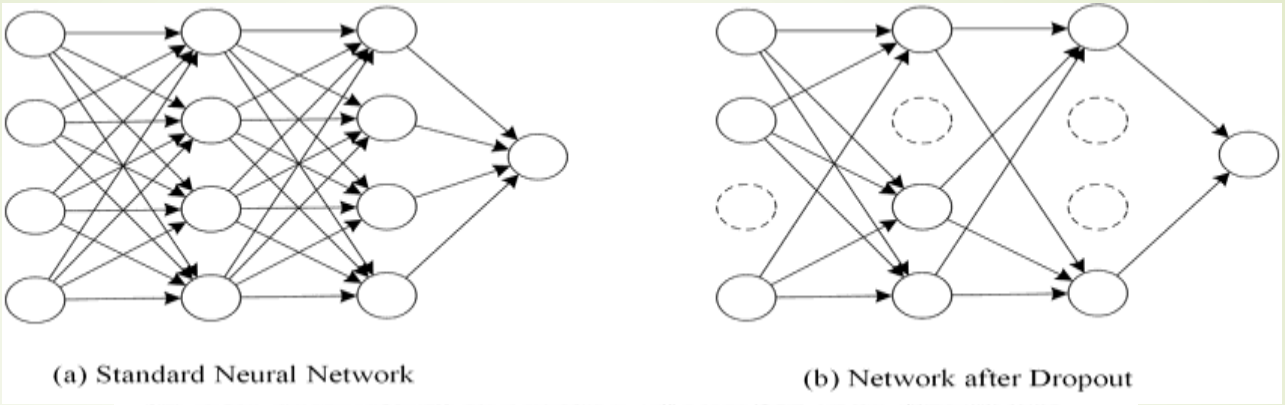
The ***vector values for a word represent its position*** in this embedding space. Synonyms are found close to each other while words with opposite meanings have a large distance between them. You can also apply mathematical operations on the vectors which should produce semantically correct results. A typical example is that the sum of the word embeddings of *king* and *female* produces the word embedding of *queen*.

Convolutional Neural Networks



In deep learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural network, most commonly applied to analyze visual imagery. They are also known as **shift invariant** or **space invariant artificial neural networks** (**SIANN**), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series.

Drop Out



(a) Standard Neural Network

(b) Network after Dropout

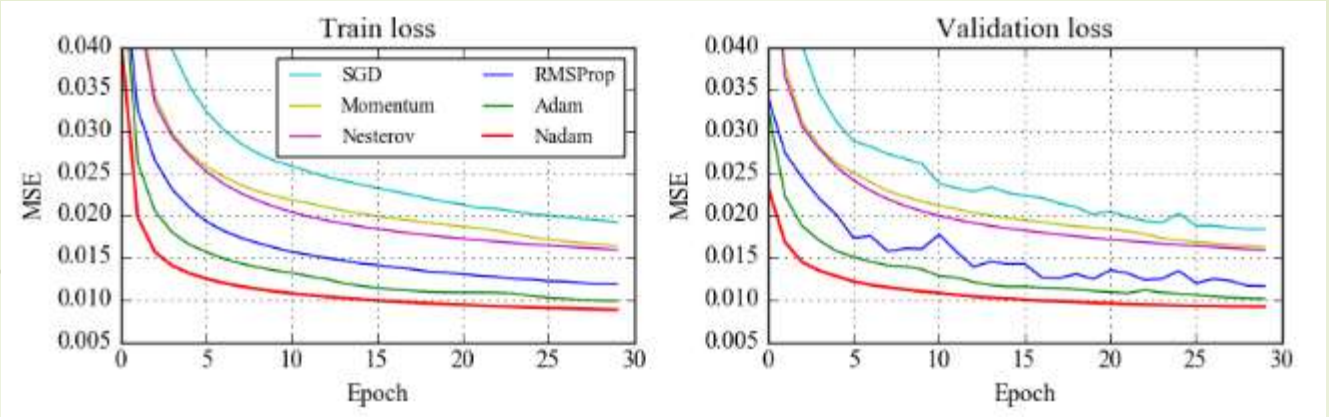
Deep learning neural networks are likely to quickly overfit a training dataset with few examples.

Ensembles of neural networks with different model configurations are known to reduce overfitting, but require the additional computational expense of training and maintaining multiple models.

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout and offers a very computationally cheap and remarkably effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all kinds.

In this post, you will discover the use of dropout regularization for reducing overfitting and improving the generalization of deep neural networks.

Adam Optimizer



Adam [1] is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. First published in 2014, Adam was presented at a very prestigious conference for deep learning practitioners —ICLR 2015. The paper contained some very promising diagrams, showing huge performance gains in terms of speed of training. However, after a while people started noticing, that in some cases Adam actually finds worse solution than stochastic gradient descent. A lot of research has been done to address the problems of Adam.

Properties Of Adam Optimizer

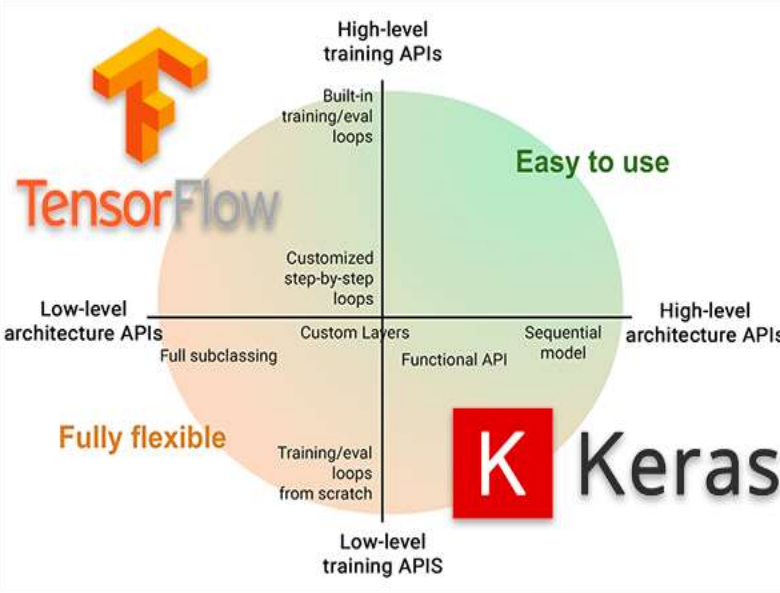
1. Actual step size taken by the Adam in each iteration is approximately bounded the step size hyper-parameter. This property add intuitive understanding to previous unintuitive learning rate hyper-parameter.
2. Step size of Adam update rule is invariant to the magnitude of the gradient, which helps a lot when going through areas with tiny gradients (such as saddle points or ravines). In these areas SGD struggles to quickly navigate through them.
3. Adam was designed to combine the advantages of Adagrad, which works well with sparse gradients, and RMSprop, which works well in on-line settings. Having both of these enables us to use Adam for broader range of tasks. Adam can also be looked at as the combination of RMSprop and SGD with momentum.

Keras

Installing Keras

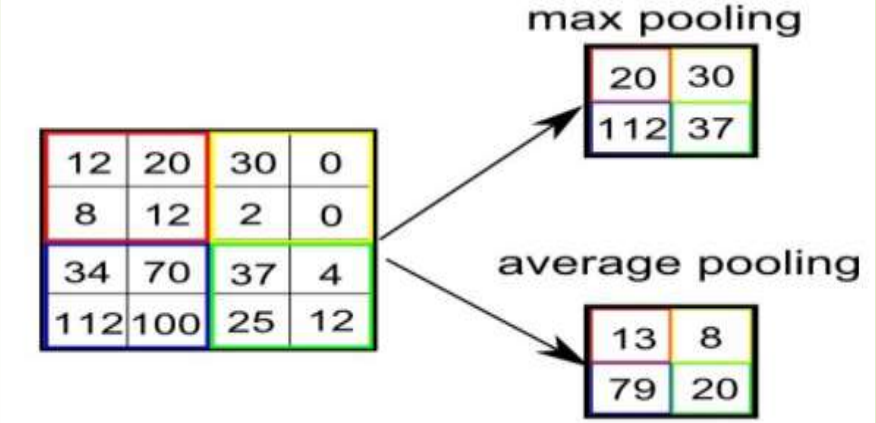
Keras is a code library that provides a relatively easy-to-use Python language interface to the relatively difficult-to-use TensorFlow library. Installing Keras involves three main steps. First you install Python and several required auxiliary packages such as NumPy and SciPy. Then you install TensorFlow and Keras as add-on Python packages.

Although it's possible to install Python and the packages required to run Keras separately, it's much better to install a Python distribution, which is a collection containing the base Python interpreter and additional packages that are compatible with one another. For my demo, I installed the Anaconda3 4.1.1 distribution (which contains Python 3.5.2), TensorFlow 1.7.0 and Keras 2.1.5.



Pooling Layer and Average Pooling

Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively.



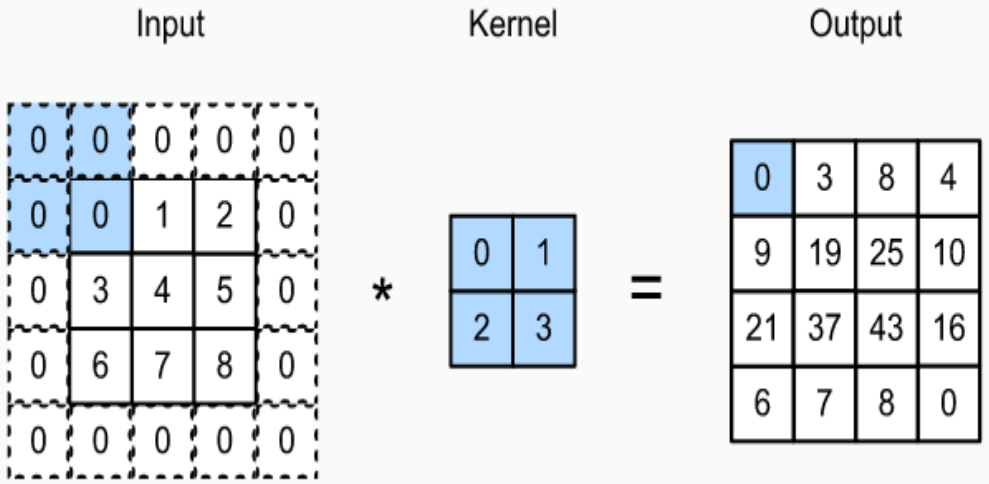
- **Average Pooling**: Calculate the average value for each patch on the feature map.
- **Maximum Pooling (or Max Pooling)**: Calculate the maximum value for each patch of the feature map.

Padding

All the neural networks require to have inputs that have the same shape and size. However, when we pre-process and use the texts as inputs for our model e.g. LSTM, not all the sentences have the same length. In other words, naturally, some of the sentences are longer or shorter. We need to have the inputs with the same size, this is where the padding is necessary.

Before we begin, here I describe how to prepare input text and then tokenize the words as well as the sentences with TensorFlow's Tokenizer tool.

Then we need to do padding, since every sentence in the text has not the same number of words, we can also define maximum number of words for each



Two-dimensional cross-correlation with padding.

Software Specification

- **FRONT END : Tkinder by Python**
- **BACK END : Python**
- **PLATFORM : Windows**

Code Description:

- processReview function:
- Input::

```
gooooooooddddd mobile
I love this handset        yIPPeeee
bad  display........HATE IT
sometimes creates problem
```

- Output::

```
>>>
gooooooooddddd mobile
i love this handset yippeeee
bad display........hate it
sometimes creates problem
```

- getFeatureVector function

```
processedReview:
this is good mobile
featureVector:
['good', 'mobile']
processedReview:
love the display
featureVector:
['love', 'display']
processedReview:
excellent performance
featureVector:
['excellent', 'performance']
processedReview:
good sound quality
featureVector:
['good', 'sound', 'quality']
processedReview:
bad touch sensitive
featureVector:
['bad', 'touch', 'sensitive']
processedReview:
poor material
featureVector:
['poor', 'material']
processedReview:
never buy this mobile
featureVector:
['never', 'buy', 'mobile']
processedReview:
buy this mobile for range upto 5000
featureVector:
['buy', 'mobile', 'range', 'upto']
processedReview:
daily use mobile
featureVector:
['daily', 'use', 'mobile']
```

- featureList(removes repetition)
- Input::

```
------BEFORE----- ['good', 'mobile', 'love', 'display', 'excellent', 'performanc
e', 'good', 'sound', 'quality', 'bad', 'touch', 'sensitive', 'poor', 'material',
 'never', 'buy', 'mobile', 'buy', 'mobile', 'range', 'upto', 'daily', 'use', 'mo
bile', 'sometimes', 'os', 'hangs', 'lags', 'switching', 'apps']
```

- Output::

```
-------AFTER------ ['os', 'sensitive', 'quality', 'never', 'performance', 'mobil
e', 'lags', 'daily', 'bad', 'excellent', 'upto', 'touch', 'range', 'love', 'use'
, 'sometimes', 'switching', 'material', 'apps', 'sound', 'display', 'poor', 'han
gs', 'good', 'buy']
```

- Training set::

```
------BEFORE----- <function extract_features at 0x04A25150>
-------AFTER------ [({'contains(sensitive)': False, 'contains(excellent)': False
, 'contains(bad)': False, 'contains(os)': False, 'contains(lags)': False, 'conta
ins(daily)': False, 'contains(love)': False, 'contains(material)': False, 'conta
ins(use)': False, 'contains(buy)': False, 'contains(quality)': False, 'contains(
mobile)': True, 'contains(upto)': False, 'contains(touch)': False, 'contains(som
etimes)': False, 'contains(poor)': False, 'contains(switching)': False, 'contain
s(never)': False, 'contains(sound)': False, 'contains(hangs)': False, 'contains(
good)': True, 'contains(apps)': False, 'contains(performance)': False, 'contains
(display)': False, 'contains(range)': False}, 'positive'), ({'contains(sensitive
)': False, 'contains(excellent)': False, 'contains(bad)': False, 'contains(os)':
 False, 'contains(lags)': False, 'contains(daily)': False, 'contains(love)': Tru
e, 'contains(material)': False, 'contains(use)': False, 'contains(buy)': False,
'contains(quality)': False, 'contains(mobile)': False, 'contains(upto)': False,
'contains(touch)': False, 'contains(sometimes)': False, 'contains(poor)': False,
 'contains(switching)': False, 'contains(never)': False, 'contains(sound)': Fals
e, 'contains(hangs)': False, 'contains(good)': False, 'contains(apps)': False, '
contains(performance)': False, 'contains(display)': True, 'contains(range)': Fal
se}, 'positive'), ...]
```

Results

- Performance of model examined using the measures:
  - Accuracy: (True Positive + True Negative)/N
    - Number of instances classified correctly
  - Recall: True Positive/(False Negative + True Positive)
    - True positive rate: How many positive instances are predicted correctly
  - Precision: True Negative / (False Positive + True Negative)
    - True negative rate: How negative instances are predicted correctly
  - F1-score: 2*(Recall * Precision)/(Recall + Precision)
    - Weighted Average of precision and recall

CONCLUSION AND FUTURE WORKS

- Sentiment analysis or opinion mining is a field of study that analyzes people's sentiments, attitudes, or emotions towards certain entities. This project deal with a fundamental problem of sentiment analysis, sentiment polarity categorization. Online product reviews from Amazon.com are selected as data used for this project.
- We plan to use more exhaustive techniques in future to build a full-fledged model that is able to analyse a large set of data in a little time and with good accuracy.

References

- https://www.GeeksforGeeks.org/
- https://www.python.org/
- https://www.coursera.org/
- https://docs.python.org/3/tutorial/
- https://www.Medium.com/
- https://www.google.com/
- https://www.Medium.com/
- https://www.Wikipedia.org/

Thank You