Sentiment

Dereje T. Abzaw

Outline

- Text Categorization
- Binary Classification
- Sentiment Analysis
- Approaches in Sentiment Analysis
- Spam Detection
- VADER
- Introduction to Naive Bayes

Sentiment, Classification , Text Categorization

- Classification lies at the heart of both human and machine intelligence.
- Many language processing tasks involve classification
- Whether **single-word tokens**, **n-grams**, **stems**, or **lemmas** contain some important information called **sentiment**
- Sentiment is the overall feeling or emotion that the word invokes.

Contd …

- We focus on one common text categorization task, sentiment analysis, the extraction of sentiment, the **positive** or **negative** orientation that a writer expresses toward some object, simplest of this being a **binary classification**.
  - Reviews with four or more stars can be marked as positive, three or fewer stars as negative
  - An editorial or political text expresses sentiment toward a candidate or political action.
  - Tweets containing happy emoticons can be marked as positive, sad emoticons as negative
  - Extracting consumer or public sentiment is thus relevant for fields from **marketing** to **politics**.

Contd ...

- *...zany characters and richly applied satire, and some great plot twists*
- *It was pathetic. The worst part about it was the boxing scenes...*
- *...awesome caramel sauce and sweet toasty almonds. I love this place!*
- *...awful pizza and ridiculously overpriced...*

A little offshore … “Related Problems”

- **Subjectivity**
  - Identifying the parts of a text that express **subjective** opinions
- **Stance Classification**
  - In debates, each participant takes a side: for example, advocating for or against proposals
  - Identifying author’s position is a problem
- **Targeted Sentiment Analysis**
  - When the expression of sentiment is more than just a simple **binary label**
  - “Go to Heaven for the climate, Hell for the company.” **–Mark Twain**

Alternate Approaches to Sentiment Analysis

1. **Regression**: A more challenging version of sentiment analysis is to determine not just the class but its rating on a numerical scale.
   - Linear Regression (Least squares)
   - Ridge Regression
2. **Ordinal Ranking**: In many problems, the labels are ordered but discrete:
   - For example, product reviews are often integers on a scale of 1 − 5, and grades are on a scale of A − F .
   - Such problems can be solved by discretizing the score $\theta \cdot x$ into “ranks”
3. **Lexicon-based Classification**
   - Sentiment analysis is one of the only NLP tasks where hand-crafted feature weights are still widely employed.
   - In lexicon-based classification the user creates a list of words for each label, and then classifies each document based on how many of the words from each list are present

Spam Detection

- **Spam detection** is another important commercial application, the binary classification task of assigning an email to one of the two classes **spam** or **not-spam**.
- Junk Mail, Troll Messages
- For example **suspicious phrases** like:
  - “online pharmaceutical” , “WITHOUT ANY COST” , “Dear Winner”
- The goal of classification is to take a single observation, extract some useful features, and thereby classify the observation into one of a set of discrete classes.

Sentiment Analysis...AGAIN

- Sentiment analysis can be framed as a direct application of document classification assuming reliable labels can be obtained.
- Sentiment analysis, simply, is a two or three-class problem, with sentiments of POSITIVE, NEGATIVE, and NEUTRAL
- Companies they often will provide some way for you to give feedback.
  - A star rating , which is **quantitative data** about how people feel about products they've purchased.
  - But a more natural way is to use natural language comments by giving users a blank slate (an empty text box) to fill up with comments, a **qualitative data**, about your product can produce **more detailed feedback.**
- Humans are **remarkably bad** at reading feedback, especially criticism or negative feedback, which machines don't have biases on.

Approaches of Sentiment Analysis

1. Rule Based Algorithm - **VADER** Algorithm
    - Human-designed rules, called **heuristics**, to measure sentiment.
    - A common rule-based approach to sentiment analysis is to **find keywords** in the text and **map** each one to **numerical scores** or weights in a dictionary or “mapping”
    - **Hand-composed** dictionary of keyword is needed
2. ML Model - **Naive Bayes**
    - Relies on **a labeled set** of statements or documents to train a **machine learning model** to create those rules
    - **ML sentiment model** is trained to process input text and output a numerical value for the sentiment you are trying to measure, like positivity or spamminess or trolliness.
    - Needs a lot of data, text labeled with the sentiment scores.

VADER

- A Rule-based Model for Sentiment Analysis of Social Media Text by Hutto and Gilbert
- Short for **Valence** , **Aware**, **Dictionary** for, s**Entiment** ,**Reasoning**.
- The NLTK package has an implementation of the VADER algorithm.
- * Code Implementation Here*

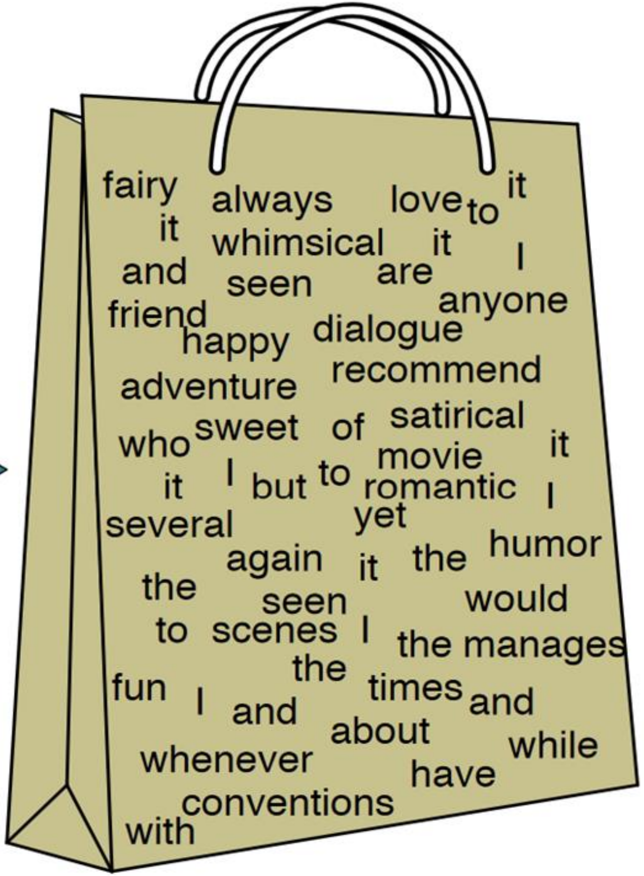
Disclaimer: It is not inspired from the name Darth Vader

Bag of Words

- A text document should be represented as if it were a bag of words :- that is, an unordered set of words with their position ignored, keeping only their frequency in the document.

- Instead of representing the word order in all the phrases like “I love this movie” and “I would recommend it”,
  - The word “I” occurred 5 times in the entire excerpt,
  - The word “it” 6 times,
  - The words “love”, “recommend”, and “movie” once.

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Naive Bayes

- A Naive Bayes model tries to find keywords in a set of documents that are predictive of your target (output) variable.
- When your target variable is the sentiment you are trying to predict, the model will find words that predict that sentiment.
- Naive Bayes model scores are like VADER score but **you won't have to be limited** to just what an individual human decided those scores should be.
- The machine will find the "best" scores for any problem.

Contd …

- Naive Bayes is a **probabilistic classifier**
- This idea of Bayesian inference has been known since the work of Bayes and was first applied to text classification by Mosteller and Wallace.
- The intuition of Bayesian classification is to use Bayes' rule to transform probabilities.

Contd …

- Naive Bayes , a **probabilistic classifier**, meaning that for a document d, out of all classes **c ∈ C** the classifier returns the **class** $\hat{c}$ which has the maximum posterior probability given the document.
- The $\hat{c}$ refers an **estimate of a class**

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d)$$

Contd … Naive Bayes

- The inference is based on the conditional probability of the bayes rule.

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

Contd ...

$$\underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{\cancel{P(d)}}$$

Doesn't change for each class for the same document d, which must have the **same probability P(d)**

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

Contd ...

$$\underset{c \in C}{\operatorname{argmax}} \underbrace{P(d|c)}_{\text{words are generated by sampling}} \underbrace{P(c)}_{\text{class is sampled}}$$

words are generated by sampling

class is sampled

→ Making Naive Bayes algorithm a **Generative Model.**

Contd ...

$$\underset{c \in C}{\operatorname{argmax}} \underbrace{P(d|c)}_{\text{Likelihood}} \underbrace{P(c)}_{\text{Prior}}$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, ..., f_n|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

Representing a document **d** as a set of features **f1, f2, ..., fn**

Contd …

- Naive Bayes classifiers make two simplifying assumptions:
  - **Bag-of-words Assumption:** Word position doesn't matter
  - **Naive Bayes Assumption:** this is the conditional independence assumption that the probabilities are independent so it can be 'naively' multiplied as:

$$P(f_1, f_2, \ldots, f_n | c) = P(f_1|c) \cdot P(f_2|c) \cdot \ldots \cdot P(f_n|c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c) \prod_{f \in F} P(f|c)$$

Evaluation: Precision, Recall, F-Measure

- Consider **spam detection (**Binary Classification**)**:
  - spam category (“positive”) or not in the spam category (“negative”)
- A metric for knowing how well our spam detector is doing is needed.
- To evaluate any system for detecting things, we have a **confusion matrix.**

		gold standard labels		
		gold positive	gold negative	
system output labels	system positive	**true positive**	**false positive**	$\textbf{precision} = \frac{\text{tp}}{\text{tp+fp}}$
	system negative	**false negative**	**true negative**	
		$\textbf{recall} = \frac{\text{tp}}{\text{tp+fn}}$		$\textbf{accuracy} = \frac{\text{tp+tn}}{\text{tp+fp+tn+fn}}$

Contd ...

- **Precision** measures the percentage of the items that the system detected that are in fact positive.
- **Recall** measures the percentage of items actually present in the input that were recalled correctly identified by the system.
- There are many ways to define a single metric that incorporates aspects of both precision and recall,the simplest being **F-measure.**
- **F-measure** is the harmonic mean of Recall and Precision

$$F_1 = \frac{2PR}{P+R}$$

Contextual Considerations

- **High Precision**: Typically, a precision score above 80% is considered high.
    - In domains where false positives carry a high cost (e.g., spam detection),
    - Aiming for a higher precision is crucial.
- **Low Precision**: A precision score below 50% is generally considered low
    - Indicating that the model's predictions of the **positive class** are often incorrect.
- **High Recall**: A recall score above 80% is often considered high.
    - In contexts where **missing out** on **true positive** cases carries significant consequences (e.g., fraud detection, disease screening), a high recall is essential.
- **Low Recall**: A recall score below 50% is viewed as low
    - Suggesting that the model is missing a substantial number of positive cases.

Contd …

- **High Accuracy**: An accuracy above 90% is typically seen as high
  - If datasets are balanced
  - Each class is equally represented.
- **Low Accuracy**: Accuracy below 70%
- **High F1-Score**: An F1-score above 70% is generally considered good
  - Indicating a **strong balance** between precision and recall.
- **Low F1-Score**: An F1-score below 50% is considered poor
  - Suggesting the model **is not effectively balancing** precision and recall.

Example - 1

Let's assume the results from classifying tweets , from a certain model , as "Positive" or "Negative" are as follows:

- **True Positives** (**TP**): 80 (tweets correctly identified as Positive)
- **False Positives** (**FP**): 30 (tweets incorrectly identified as Positive)
- **True Negatives** (**TN**): 70 (tweets correctly identified as Negative)
- **False Negatives** (**FN**): 20 (tweets incorrectly identified as Negative)

Calculate **Recall**, **Precision**, **F1-Score** and **Accuracy** of the model?

Example - 2

Suppose you are developing a machine learning model to diagnose a rare but serious disease based on patient symptoms and test results. Given the outcomes:

- **True Positives (TP)**: 50 (patients correctly identified as having the disease)
- **False Positives (FP)**: 10 (patients incorrectly identified as having the disease)
- **True Negatives (TN)**: 900 (patients correctly identified as not having the disease)
- **False Negatives (FN)**: 40 (patients incorrectly identified as not having the disease)

Calculate **Recall**, **Precision**, **F1-Score** and **Accuracy** of the model?