PEOPLE ARE ALWAYS ONLINE THESE DAYS...

BUYING THINGS..

READING THINGS..

WATCHING THINGS..

..AND EXPRESSING THEIR OPINION ABOUT THINGS

ANYONE WHO IS SELLING A PRODUCT OR PROVIDING A SERVICE - WHETHER IT'S OFFLINE OR ONLINE WANTS AND NEEDS TO UNDERSTAND WHAT PEOPLE ARE SAYING ABOUT THEM

ANYONE WHO IS SELLING A PRODUCT OR PROVIDING A SERVICE - WHETHER IT'S OFFLINE OR ONLINE

WANTS AND NEEDS TO UNDERSTAND WHAT PEOPLE ARE SAYING ABOUT THEM

REVIEWS COMMENTS EMAILS

TWEETS STATUS MESSAGES

ALL THESE CARRY INFORMATION ABOUT PEOPLE'S OPINION

ANYONE WHO IS SELLING A PRODUCT
OR PROVIDING A SERVICE -
WHETHER IT'S OFFLINE OR ONLINE

WANTS AND NEEDS TO UNDERSTAND
WHAT PEOPLE ARE SAYING ABOUT THEM

REVIEWS COMMENTS
EMAILS
TWEETS STATUS MESSAGES

ALL THESE CARRY INFORMATION
ABOUT PEOPLE'S OPINION

DO THEY LIKE YOUR BRAND?

OR DO THEY HATE IT?

DO THEY FEEL ANGER OR TRUST YOU?

OPINION MINING

ALSO KNOWN AS

SENTIMENT ANALYSIS

IS A FIELD OF NLP THAT TRIES TO EXTRACT THIS
KIND OF SUBJECTIVE INFORMATION FROM TEXT

SENTIMENT ANALYSIS

REVIEWS

IS THIS REVIEW POSITIVE OR NEGATIVE?

EMAILS COMMENTS

ARE PEOPLE SATISFIED OR
DISSATISFIED WITH MY SERVICE?

TWEETS

FROM THIS SAMPLE OF TWEETS -
HOW ARE PEOPLE RESPONDING TO AN AD
OR AN EVENT?

STATUS MESSAGES

HOW ARE PEOPLE REACTING TO A
CANDIDATE'S SPEECH
DURING AN ELECTION CAMPAIGN ?

ALL THESE CARRY INFORMATION
ABOUT PEOPLE'S OPINION

SENTIMENT ANALYSIS

IS THIS REVIEW POSITIVE OR NEGATIVE?

(DETERMINING THE POLARITY or SEMANTIC ORIENTATION OF A DOCUMENT)

THIS IS THE SIMPLEST AND MOST POPULAR TASK IN SENTIMENT ANALYSIS

THERE ARE MANY DIFFERENT WAYS OF APPROACHING THIS PROBLEM:

RULE BASED APPROACHES

MACHINE LEARNING BASED APPROACHES

SENTIMENT ANALYSIS

RULE BASED APPROACHES

HERE IS ONE SIMPLE RULE BASED APPROACH

THIS WOULD REQUIRE A LEXICON -
A RESOURCE WHERE ALL WORDS HAVE BEEN CLASSIFIED AS POSITIVE OR NEGATIVE

1. LOOK AT ALL THE WORDS IN THE TEXT AND CLASSIFY EACH OF THEM AS POSITIVE/NEGATIVE

THERE ARE MANY SUCH HAND ANNOTATED LEXICONS MADE AVAILABLE BY UNIVERSITY RESEARCHERS (MORE ON THIS LATER..)

2. IF THERE ARE MORE POSITIVE WORDS THAN NEGATIVE WORDS - CLASSIFY THE DOCUMENT AS POSITIVE

" I REALLY LIKE THE NEW IPHONE. IT'S AWESOME! **POSITIVE**

"I HATE APPLE!" **NEGATIVE**

SENTIMENT ANALYSIS

RULE BASED APPROACHES

THERE ARE MANY RULE BASED APPROACHES SUGGESTED FOR SENTIMENT ANALYSIS

SOME ARE VERY COMPLEX AND VERY GOOD AT CLASSIFYING AS WELL

VADER IS ONE SUCH RULE-BASED MODEL

USE OF CAPS AND EXCLAMATION POINTS, EMOTICONS

"this food is AMAZING!!! :)"

SENTIMENT ANALYSIS

RULE BASED APPROACHES

THERE ARE MANY RULE BASED APPROACHES SUGGESTED FOR SENTIMENT ANALYSIS

SOME ARE VERY COMPLEX AND VERY GOOD AT CLASSIFYING AS WELL

VADER IS ONE SUCH RULE-BASED MODEL

USE OF CAPS AND EXCLAMATION POINTS, EMOTICONS

WORDS THAT SIGNAL A SHIFT IN EMOTION - BUT, HOWEVER

"I liked the book initially but not after the first 100 pages"

SENTIMENT ANALYSIS

RULE BASED APPROACHES

THERE ARE MANY RULE BASED APPROACHES SUGGESTED FOR SENTIMENT ANALYSIS

SOME ARE VERY COMPLEX AND VERY GOOD AT CLASSIFYING AS WELL

VADER IS ONE SUCH RULE-BASED MODEL

USE OF CAPS AND EXCLAMATION POINTS, EMOTICONS

WORDS THAT SIGNAL A SHIFT IN EMOTION - BUT, HOWEVER

ADVERBS THAT ACT AS INTENSIFIERS - EXTREMELY, HARDLY, VERY

"this restaurant is really good"

"this food is hardly edible"

SENTIMENT ANALYSIS

RULE BASED APPROACHES

THERE ARE MANY RULE BASED APPROACHES SUGGESTED FOR SENTIMENT ANALYSIS

SOME ARE VERY COMPLEX AND VERY GOOD AT CLASSIFYING AS WELL

VADER IS ONE SUCH RULE-BASED MODEL

USE OF CAPS AND EXCLAMATION POINTS, EMOTICONS

WORDS THAT SIGNAL A SHIFT IN EMOTION - BUT, HOWEVER

ADVERBS THAT ACT AS INTENSIFIERS - EXTREMELY, HARDLY, VERY

THESE ARE A FEW EXAMPLES OF THINGS THE RULES THAT VADER USES ARE BASED ON

SENTIMENT ANALYSIS

IS THIS REVIEW POSITIVE OR NEGATIVE?

(DETERMINING THE POLARITY or SEMANTIC ORIENTATION OF A DOCUMENT)

THIS IS THE SIMPLEST AND MOST POPULAR TASK IN SENTIMENT ANALYSIS

THERE ARE MANY DIFFERENT WAYS OF APPROACHING THIS PROBLEM:

RULE BASED APPROACHES

MACHINE LEARNING BASED APPROACHES

SENTIMENT ANALYSIS

MACHINE LEARNING BASED APPROACHES

APPROACH IT AS A CLASSIFICATION PROBLEM

CLASSIFY A DOCUMENT AS POSITIVE OR NEGATIVE

NAIVE BAYES CLASSIFICATION

SUPPORT VECTOR MACHINES

THESE ARE COMMONLY USED FOR SENTIMENT ANALYSIS

THERE ARE SOME TRICKY DETAILS THOUGH...

WHAT DO YOU USE AS TRAINING DATA?

WHAT FEATURES DO YOU CHOOSE?

WHAT DO YOU USE AS TRAINING DATA?

TO CLASSIFY A DOCUMENT AS POSITIVE OR NEGATIVE

SEVERAL HUMAN ANNOTATED CORPORA ARE AVAILABLE

YOU NEED DOCUMENTS THAT ARE ALREADY MARKED AS POSITIVE OR NEGATIVE

NIEK SANDERS ~5000 LABELLED TWEETS

AMAZON PRODUCT REVIEWS (JOHNS HOPKINS CS)

MOVIE REVIEWS (CORNELL CS)

WHAT DO YOU USE AS TRAINING DATA?

TO CLASSIFY A DOCUMENT AS POSITIVE OR NEGATIVE

YOU NEED DOCUMENTS THAT ARE ALREADY MARKED AS POSITIVE OR NEGATIVE

SOME DOCUMENTS COME WITH IMPLICIT LABELS

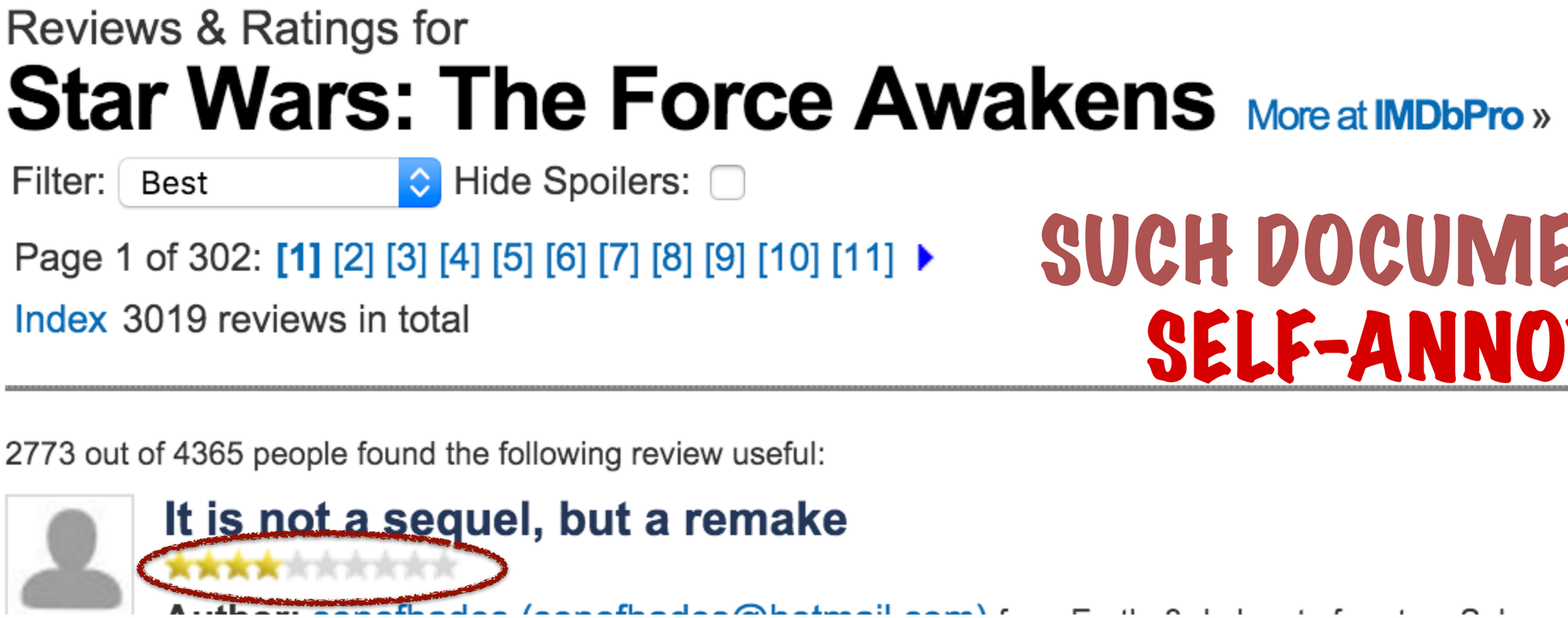
A REVIEW USUALLY COMES WITH A RATING

A CUSTOMER EMAIL MIGHT BE IN CONTEXT OF A SURVEY (WHICH INCLUDES SOME RATING)

WHAT DO YOU USE AS TRAINING DATA?

SOME DOCUMENTS COME WITH IMPLICIT LABELS

A REVIEW USUALLY COMES WITH A RATING



SUCH DOCUMENTS ARE SELF-ANNOTATED

SENTIMENT ANALYSIS

MACHINE LEARNING BASED APPROACHES

APPROACH IT AS A CLASSIFICATION PROBLEM

CLASSIFY A DOCUMENT AS POSITIVE OR NEGATIVE

NAIVE BAYES CLASSIFICATION

SUPPORT VECTOR MACHINES

THESE ARE COMMONLY USED FOR SENTIMENT ANALYSIS

THERE ARE SOME TRICKY DETAILS THOUGH...

WHAT DO YOU USE AS TRAINING DATA?

WHAT FEATURES DO YOU CHOOSE?

WHAT FEATURES DO YOU CHOOSE?

THE SIMPLEST WAY IS TO LOOK AT THE INDIVIDUAL WORDS IN THE DOCUMENT

WHAT FEATURES DO YOU CHOOSE?

THE SIMPLEST WAY IS TO LOOK AT THE INDIVIDUAL WORDS IN THE DOCUMENT

IF YOU ARE USING NAIVE - BAYES - COMPUTE POSTERIOR PROBABILITIES

$$P(\text{DOCUMENT IS POSITIVE/WORDS}) = \frac{P(\text{DOCUMENT IS POSITIVE}) * P(W1/\text{DOCUMENT IS POSITIVE}) * P(W2/\text{DOCUMENT IS POSITIVE}) * \ldots}{P(W1) * P(W2) * \ldots}$$

$$P(\text{DOCUMENT IS NEGATIVE/WORDS}) = \frac{P(\text{DOCUMENT IS NEGATIVE}) * P(W1/\text{DOCUMENT IS NEGATIVE}) * P(W2/\text{DOCUMENT IS NEGATIVE}) * \ldots}{P(W1) * P(W2) * \ldots}$$

SINCE THE DENOMINATORS ARE SAME, COMPUTE THE NUMERATORS

PICK THE CLASS WHOSE POSTERIOR PROBABILITY IS GREATER

WHAT FEATURES DO YOU CHOOSE?

IF YOU ARE USING SUPPORT VECTOR MACHINES

EXPRESS EACH DOCUMENT AS A VECTOR

IF ALL THE WORDS IN ALL THE DOCUMENTS ARE
[W1,W2,W3....WN]

ANY DOCUMENT CAN BE REPRESENTED AS [X1,X2,X3....XN]

EACH Xi INDICATES THE PRESENCE OR ABSENCE OR THE WORD Wi

IF W1 IS PRESENT IN THE DOCUMENT, X1 =1 ELSE X1=0

WHAT FEATURES DO YOU CHOOSE?

IF ALL THE WORDS IN ALL THE DOCUMENTS ARE
[W1,W2,W3....WN]

ANY DOCUMENT CAN BE REPRESENTED AS [X1,X2,X3....XN]

YOU CAN ALSO USE
WEIGHTS TO INDICATE HOW
POSITIVE OR NEGATIVE THE
WORD **Wi** IS

IF W1 IS PRESENT IN THE DOCUMENT
IF W1 IS POSITIVE, X1 = 1
IF W1 IS NEGATIVE, X1 = -1

TO DETERMINE WHETHER A
WORD IS POSITIVE OR
NEGATIVE, USE A **LEXICON**

WHAT FEATURES DO YOU CHOOSE?

TO DETERMINE WHETHER A WORD IS POSITIVE OR NEGATIVE, USE A **LEXICON**

A LEXICON IS A RESOURCE WITH INFORMATION ABOUT WORDS

A DICTIONARY IS A PERFECT EXAMPLE OF A LEXICON

THERE ARE SEVERAL LEXICONS AVAILABLE WHICH PROVIDE INFORMATION LIKE POLARITY OF A WORD (POSITIVE/NEGATIVE) ETC

SENTIWORDNET IS A SPECIAL LEXICON THAT PROVIDES **POSITIVE, NEGATIVE AND OBJECTIVITY** SCORES FOR EVERY WORD

WHAT FEATURES DO YOU CHOOSE?

IF ALL THE WORDS IN ALL THE DOCUMENTS ARE
[W1,W2,W3....WN]

ANY DOCUMENT CAN BE REPRESENTED AS [X1,X2,X3....XN]

YOU CAN ALSO USE WEIGHTS TO INDICATE HOW POSITIVE OR NEGATIVE THE WORD Wi IS

IF W1 IS PRESENT IN THE DOCUMENT
IF W1 IS POSITIVE, X1 =1
IF W1 IS NEGATIVE, X1 = -1

ONE LITTLE DETAIL HERE..

CONSIDER ALL WORDS BETWEEN A NEGATION (NOT, NO ETC) AND A PUNCTUATION MARK AS NEGATIVE

"THIS FOOD IS NOT GOOD"

SENTIMENT ANALYSIS

WHAT FEATURES DO YOU CHOOSE?

SOMETIMES ITS BETTER TO LOOK AT COMBINATIONS OF WORDS

"really good" "hardly edible"

BI-GRAMS OR N-GRAMS

SENTIMENT ANALYSIS

WHAT FEATURES DO YOU CHOOSE?

BI-GRAMS OR N-GRAMS

GENERATE ALL PAIRS OF WORDS (BI-GRAMS) IN ALL DOCUMENTS [P1,P2,P3,..PN]

THE FEATURE VECTOR FOR A DOCUMENT INDICATES THE PRESENCE OR ABSENCE OF THESE BIGRAMS

A LEXICON IS A RESOURCE WITH INFORMATION ABOUT WORDS

A SENTIMENT LEXICON HAS INFORMATION SUCH AS

LISTS OF WORDS WHICH ARE POSITIVE AND NEGATIVE

WHAT EMOTION DOES A WORD EXPRESS?
(PLEASURE, PAIN, ANTICIPATION...)

INTENSITY OF A WORD
(GOOD VS GREAT.. GREAT HAS HIGHER INTENSITY)

GENERAL INQUIRER, MPQA, LIWC, SENTIWORDNET
ARE SOME OF THE COMMONLY USED SENTIMENT LEXICONS

SENTIWORDNET

A SENTIMENT LEXICON THAT'S BASED ON WORDNET

WORDNET IS LIKE A VERY SPECIAL KIND OF THESAURUS

IT IS LIKE A NETWORK OF
RELATIONSHIPS BETWEEN WORDS -
BASED ON THEIR MEANING

WORDNET IS LIKE A VERY SPECIAL KIND OF THESAURUS

A WORD CAN HAVE MANY DIFFERENT MEANINGS

TAKE THE WORD DOG

" I love my dog!"

ANIMAL
NOUN

CHASE
VERB

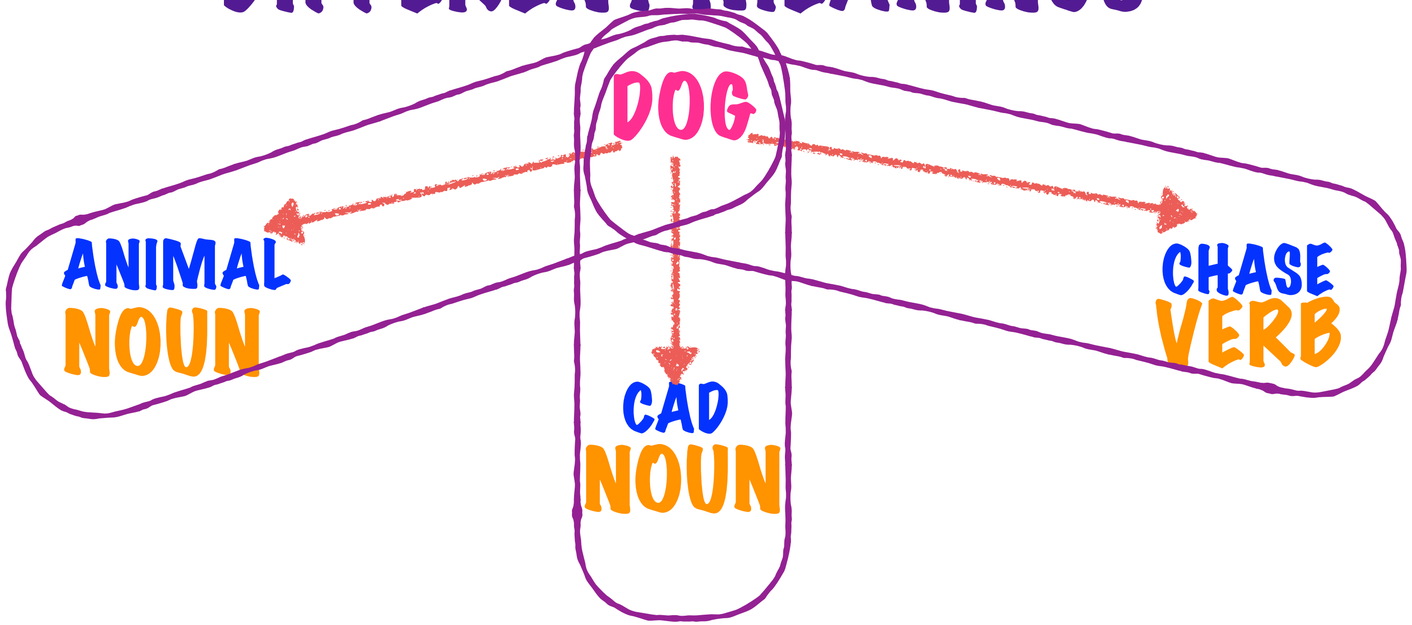
"He has been dogging my steps."

"That man is a dog, you can't trust him"

CAD
NOUN

WORDNET IS LIKE A VERY SPECIAL KIND OF THESAURUS

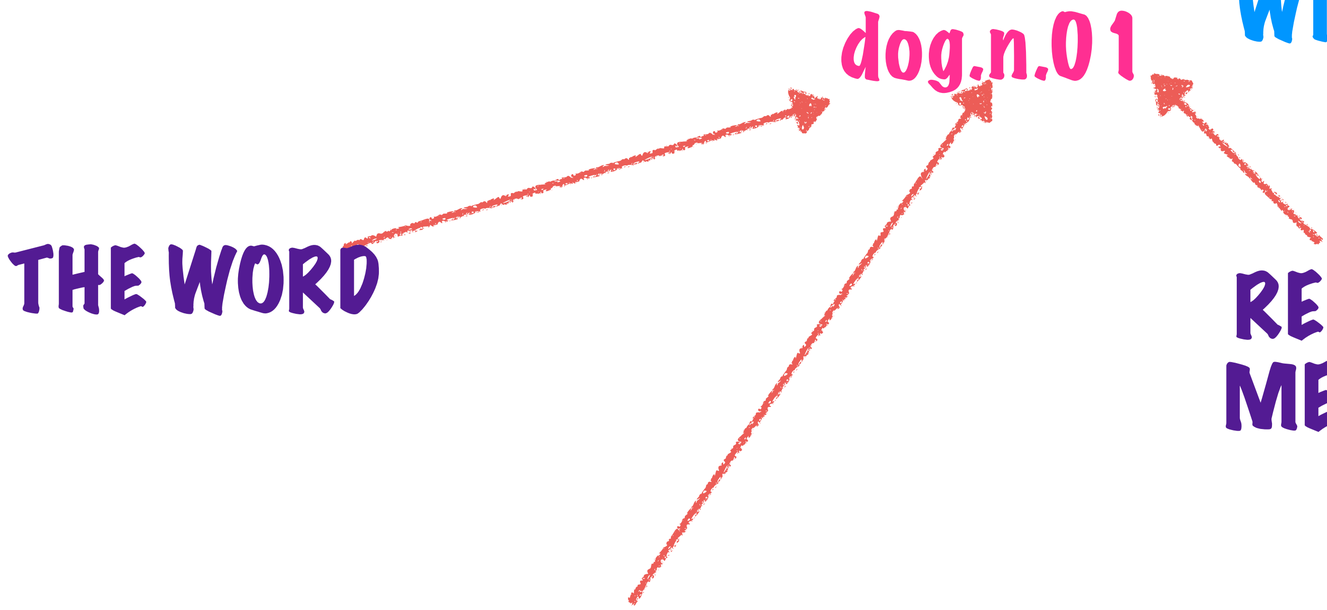
A WORD CAN HAVE MANY DIFFERENT MEANINGS



A (WORD, MEANING) PAIR IS CALLED A LEMMA

HERE IS AN EXAMPLE OF A LEMMA REPRESENTED IN WORDNET

A SYNSET GROUPS TOGETHER LEMMAS (WORD-MEANING PAIRS) WITH THE SAME MEANING



HERE IS AN EXAMPLE OF A SYNSET

A SYNSET GROUPS TOGETHER LEMMAS (WORD-MEANING PAIRS) WITH THE SAME MEANING

{'sanely.r.01.sanely',
'sanely.r.01.sensibly',
'sanely.r.01.reasonably'}

ALL OF THE ELEMENTS IN THIS SYNSET HAVE THE SAME MEANING/ DEFINITION

WORDNET ALSO PROVIDES RELATIONSHIPS BETWEEN SYNSETS

"with good sense or in a reasonable or intelligent manner"

WORDNET ALSO PROVIDES RELATIONSHIPS BETWEEN SYNSETS

SENTIWORDNET TAKES THE SYNSETS IN WORDNET AND ASSIGNS THEM A POLARITY SCORE

{oak} —TYPE-OF→ {tree}

{family, family unit} —HAS-MEMBER→ {child, kid}

{snore, saw wood} —ENTAILS→ {sleep,slumber}

SENTIWORDNET TAKES THE SYNSETS IN WORDNET AND ASSIGNS THEM A POLARITY SCORE

EVERY SYNSET HAS 3 SCORES:

A POSITIVE POLARITY SCORE

A NEGATIVE POLARITY SCORE

AN OBJECTIVITY SCORE

THESE 3 SCORES ADD UP TO 1

EVERY SYNSET HAS 3 SCORES:

TAKE THE SYNSET FOR 'happy.a.01'

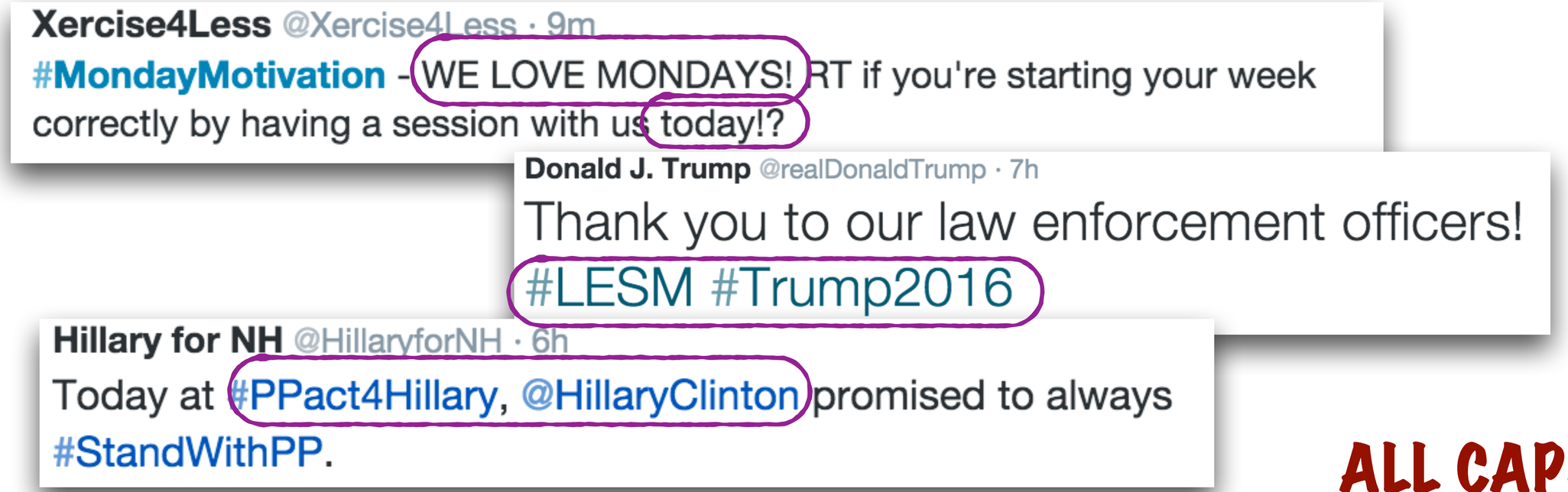
A POSITIVE POLARITY SCORE	0.875
A NEGATIVE POLARITY SCORE	0
AN OBJECTIVITY SCORE	0.125

THESE 3 SCORES ADD UP TO 1

ALL MEMBERS OF THIS SYNSET HAVE THE MEANING

"enjoying or showing or marked by joy or pleasure"

LET'S SAY YOU WANTED TO PROCESS SOME TWEETS



THESE TWEETS HAVE TEXT WITH CERTAIN PATTERNS

ALL CAPS

WORDS BEGINNING WITH @/#

WORDS ENDING WITH !,?

THESE TWEETS HAVE TEXT WITH CERTAIN PATTERNS

ALL CAPS

WORDS BEGINNING WITH @/#

WORDS ENDING WITH !,?

HOW DO YOU EXTRACT THESE PATTERNS ?

THESE CHARACTERS ARE EXPRESSED IN A CERTAIN SYNTAX

REGULAR EXPRESSIONS

A REGULAR EXPRESSION IS A SEQUENCE OF CHARACTERS THAT DEFINE A SEARCH PATTERN

A REGULAR EXPRESSION IS A WAY TO EXPRESS A PATTERN, THEN YOU CAN USE IT TO FIND ALL WORDS THAT MATCH THAT PATTERN

REGULAR EXPRESSIONS

A REGULAR EXPRESSION IS A WAY TO EXPRESS A PATTERN, THEN YOU CAN USE IT TO FIND ALL WORDS THAT MATCH THAT PATTERN

1. TO SEARCH FOR A SPECIFIC WORD - EG. grey

'grey'

2. TO SEARCH FOR ONE OF A SET OF WORDS EG. grey OR white

'grey|white'

USE | TO EXPRESS ALTERNATIVES

3. TO SEARCH FOR A SET OF WORDS WITH SOME COMMON PATTERN EG:GRAY OR GREY

'gr(ale)y'

USE () TO SEPARATE OR GROUP A PATTERN FROM OTHER CHARACTERS AND SPECIFY ITS POSITION

USE | TO EXPRESS ALTERNATIVES

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

4. TO SPECIFY THAT A PATTERN REPEATS, AND THE NUMBER OF TIMES IT REPEATS

USE QUANTIFIERS

?, * , +,{n},{min,max}

? matches 0 or 1 occurrences of the previous character/element

'colou?r' ⟶ 'color' or 'colour'

USE | TO EXPRESS ALTERNATIVES

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

4. TO SPECIFY THAT A PATTERN REPEATS, AND THE NUMBER OF TIMES IT REPEATS

USE QUANTIFIERS

?, * , +,{n},{min,max}

* matches 0 or more occurrences of the previous character/element

'colou*r' ⟶ 'color' or 'colour' or 'colouur' or 'colouuuuur' and so on...

USE | TO EXPRESS ALTERNATIVES

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

4. TO SPECIFY THAT A PATTERN REPEATS, AND THE NUMBER OF TIMES IT REPEATS

USE QUANTIFIERS

?, * , +,{n},{min,max}

+ matches 1 or more occurrences of the previous character/element

'colou+r' ⟶ 'colour' or 'colouur'
or 'colouuuuur' and so on...
will not match 'color'

USE | TO EXPRESS ALTERNATIVES

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

4. TO SPECIFY THAT A PATTERN REPEATS, AND THE NUMBER OF TIMES IT REPEATS

USE QUANTIFIERS

?, * , +,{n},{min,max}

{n} matches exactly n occurrences of the previous character/element

'colou{2}r' → will only match 'colouur'

USE | TO EXPRESS ALTERNATIVES

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

4. TO SPECIFY THAT A PATTERN REPEATS, AND THE NUMBER OF TIMES IT REPEATS

USE QUANTIFIERS

?, * , +,{n},{min,max}

{n,} matches at least n or more occurrences of the previous character/ element

'colou{2,}r' →

'colouur'
or 'colouuuuur' and so on...
will not match 'color' and 'colour'

USE | TO EXPRESS ALTERNATIVES

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

4. TO SPECIFY THAT A PATTERN REPEATS, AND THE NUMBER OF TIMES IT REPEATS

USE QUANTIFIERS

?, * , +,{n},{m,n}

{m,n} matches at least m and at most n occurrences of the previous character/ element

'colou{1,2}r' → 'colour' or 'colouur'

will not match 'color' and 'colouuuur'

USE | TO EXPRESS ALTERNATIVES

USE QUANTIFIERS

?, * , +,{n},{min,max}

USE () TO SEPARATE OR GROUP A PATTE
FROM OTHER CHARACTERS AND SPECI
ITS POSITION

5. TO SPECIFY A SET OF CHARACTERS ANY ONE OF WHICH CAN BE MATCHED

'[bhc]at' → 'bat' or 'cat' or 'hat'

[a-z] Any character from a-z

[a-z0-9] Any character from a-z or 0-9

[a-cx-z] a,b,c,x,yz

USE [] TO MATCH ANY ONE OF THE CHARACTERS INSIDE THE BRACKETS

USE | TO EXPRESS ALTERNATIVES

USE QUANTIFIERS

?, * , +,{n},{min,max}

USE () TO SEPARATE OR GROUP A PATTERN FROM OTHER CHARACTERS AND SPECIFY ITS POSITION

USE [] TO MATCH ANY ONE OF THE CHARACTERS INSIDE THE BRACKETS

6. TO SPECIFY A SET OF CHARACTERS THAT SHOULD NOT BE MATCHED

'[^bhc]at'

ALL WORDS ENDING WITH 'at' EXCEPT 'bat' or 'cat' or 'hat'

USE [^] TO NOT MATCH ANY ONE OF THE CHARACTERS INSIDE THE BRACKETS

USE | TO EXPRESS ALTERNATIVES

USE QUANTIFIERS

?, * , +,{n},{min,max}

USE [^] TO NOT MATCH ANY ONE OF THE CHARACTERS INSIDE THE BRACKETS

USE [] TO MATCH ANY ONE OF THE CHARACTERS INSIDE THE BRACKETS

USE () TO SEPARATE OR GROUP A PATTERN FROM OTHER CHARACTERS AND SPECIFY ITS POSITION

WRITE A REGEXP TO MATCH ANY WORD THAT BEGINS WITH '#'

#[a-z0-9]+

BEGINS WITH #

THE WORD CAN HAVE ANY LETTER FROM a-z OR FROM 0-9

THERE CAN BE 1 OR MORE SUCH LETTERS

REGULAR EXPRESSIONS IN PYTHON

THE re MODULE

ONCE YOU HAVE CONSTRUCTED A REGULAR EXPRESSION SEARCH FOR THAT PATTERN WITHIN A STRING USING FUNCTIONS IN THIS MODULE

FIND THE POSITION WHERE THE PATTERN OCCURS

FIND ALL THE OCCURRENCES OF THE PATTERN

SUBSTITUTE ALL OCCURRENCES OF THE PATTERN WITH ANOTHER STRING

THE re MODULE

FIND THE POSITION WHERE THE PATTERN OCCURS

RETURNS NONE IF THE PATTERN DOES NOT OCCUR

re.search()

IF THE PATTERN OCCURS, RETURNS AN OBJECT WHICH HAS METHODS TO FIND THE POSITION OF THE PATTERN

```
>>> email = "tony@tiremove_thisger.net"
>>> m = re.search("remove_this", email)
>>> print email[:m.start()]
```

tony@ti

NOTE THAT THIS RETURNS ONLY THE FIRST OCCURRENCE OF THE PATTERN

THE re MODULE

FIND ALL THE OCCURRENCES OF THE PATTERN

re.findall(), re.finditer()

findall() RETURNS A LIST OF STRINGS WHEREVER THE PATTERN IS MATCHED

```
>>> tweet = "#mondays #mondayblues I hate Mondays!"
>>> re.findall("#[a-z]+", tweet)
```

[#mondays, #mondayblues]

finditer() CAN BE USED WHEN YOU WANT THE POSITIONS OF THE PATTERNS AS WELL AS THE TEXT

```
>>> tweet = "#mondays #mondayblues I hate Mondays!"
>>> for m in re.finditer("#[a-z]+", tweet):
        print tweet[m.start(),m.end()]
```

[#mondays, #mondayblues]

THE re MODULE

SUBSTITUTE ALL OCCURRENCES OF THE PATTERN WITH ANOTHER STRING

re.sub()

```
>>> tweet = "#mondays #mondayblues I hate Mondays!"
>>> re.sub("#[a-z]+", "HASHTAG",tweet)
```

"HASHTAG HASHTAG I hate Mondays!"

THE OBJECTIVE IS TO ACCEPT A SEARCH TERM FROM A USER AND FIND THE CURRENT SENTIMENT FOR THAT TERM ON TWITTER

THE OBJECTIVE IS TO ACCEPT A SEARCH TERM FROM A USER AND FIND THE CURRENT SENTIMENT FOR THAT TERM ON TWITTER

1. ACCEPT A SEARCH TERM AND DOWNLOAD THE LAST 100 TWEETS FOR THAT SEARCH TERM

2. FOR EACH OF THESE TWEETS, USE A MACHINE LEARNING CLASSIFIER AND CLASSIFY IT AS POSITIVE/NEGATIVE

3. TAKE THE MAJORITY VOTE AND THE % OF TWEETS WITH THAT SENTIMENT AND PRINT IT AS OUTPUT

1. ACCEPT A SEARCH TERM AND DOWNLOAD THE LAST 100 TWEETS FOR THAT SEARCH TERM

ACCESS THE TWITTER API USING THE python-twitter MODULE

REGISTER YOUR APPLICATION ON TWITTER AND GENERATE AN API KEY AND CREDENTIALS

https://apps.twitter.com/

THE OBJECTIVE IS TO ACCEPT A SEARCH TERM FROM A USER AND FIND THE CURRENT SENTIMENT FOR THAT TERM ON TWITTER

1. ACCEPT A SEARCH TERM AND DOWNLOAD THE LAST 100 TWEETS FOR THAT SEARCH TERM

2. FOR EACH OF THESE TWEETS, USE A MACHINE LEARNING CLASSIFIER AND CLASSIFY IT AS POSITIVE/NEGATIVE

3. TAKE THE MAJORITY VOTE AND THE % OF TWEETS WITH THAT SENTIMENT AND PRINT IT AS OUTPUT

2. FOR EACH OF THESE TWEETS, USE A MACHINE LEARNING CLASSIFIER AND CLASSIFY IT AS POSITIVE/NEGATIVE

1. DOWNLOAD A CORPUS TO USE AS TRAINING DATA

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

3. TRAIN A CLASSIFIER ON THE TRAINING DATA

4. USE THE CLASSIFIER TO CLASSIFY THE PROBLEM INSTANCES

1. DOWNLOAD A CORPUS TO USE AS TRAINING DATA

WE'LL USE NIEK SANDER'S TWEET CORPUS -
~5000 CLASSIFIED TWEETS

EACH TWEET IS LABELLED AS POSITIVE,
NEGATIVE, NEUTRAL OR IRRELEVANT

ONE LITTLE CATCH:

TWITTER DOESN'T ALLOW THE TWEET TEXT TO
BE SHARED DIRECTLY, SO THE CORPUS CONTAINS
ONLY TWEET ID AND A LABEL

DOWNLOAD THE TEXT FOR EACH OF THE
TWEETS USING THE TWITTER API

2. FOR EACH OF THESE TWEETS, USE A MACHINE LEARNING CLASSIFIER AND CLASSIFY IT AS POSITIVE/NEGATIVE

1. DOWNLOAD A CORPUS TO USE AS TRAINING DATA

PREPROCESS THE TWEETS BEFORE STEP 2

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

3. TRAIN A CLASSIFIER ON THE TRAINING DATA

4. USE THE CLASSIFIER TO CLASSIFY THE PROBLEM INSTANCES

PREPROCESS THE TWEETS BEFORE STEP 2

1. CONVERT TO LOWER CASE

2. REPLACE LINKS WITH THE STRING 'URL'

3. REPLACE @... WITH 'AT_USER'

4. REPLACE #WORD WITH THE WORD

USE REGULAR EXPRESSIONS

5. REMOVE STOPWORDS (INCLUDING URL AND USER)

6. TOKENIZE THE TWEET INTO WORDS (A LIST OF WORDS)

USE NLTK

2. FOR EACH OF THESE TWEETS, USE A MACHINE LEARNING CLASSIFIER AND CLASSIFY IT AS POSITIVE/NEGATIVE

1. DOWNLOAD A CORPUS TO USE AS TRAINING DATA

PREPROCESS THE TWEETS BEFORE STEP 2

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

3. TRAIN A CLASSIFIER ON THE TRAINING DATA

4. USE THE CLASSIFIER TO CLASSIFY THE PROBLEM INSTANCES

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

WE'LL DO THE CLASSIFICATION IN 2 WAYS :

NAIVE BAYES CLASSIFICATION

SUPPORT VECTOR MACHINES

THE FEATURE VECTOR IS SLIGHTLY DIFFERENT IN BOTH

NAIVE BAYES CLASSIFICATION

1. BUILD A VOCABULARY (LIST OF ALL THE WORDS IN ALL THE TWEETS IN THE TRAINING DATA)

2. REPRESENT EACH TWEET WITH THE PRESENCE/ ABSENCE OF THESE WORDS IN THE TWEET

{'THE', 'WORST', 'THING', 'IN', 'THE', 'WORLD'} VOCABULARY

{'THE', 'WORST', 'THING'} TWEET

(1,1,1,0,0,0) FEATURE VECTOR

USE NLTK'S BUILT IN NAIVE BAYES CLASSIFIER TO TRAIN THE CLASSIFER

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

WE'LL DO THE CLASSIFICATION IN 2 WAYS :

NAIVE BAYES CLASSIFICATION

SUPPORT VECTOR MACHINES

THE FEATURE VECTOR IS SLIGHTLY DIFFERENT IN BOTH

SUPPORT VECTOR MACHINES

THE FIRST TWO STEPS ARE THE SAME AS FOR NAIVE BAYES'

1. BUILD A VOCABULARY (LIST OF ALL THE WORDS IN ALL THE TWEETS IN THE TRAINING DATA)

SENTIWORDNET WILL HAVE A POS_SCORE, NEG_SCORE AND OBJECTIVITY SCORE FOR EVERY SYNSET

WE'LL USE THE FIRST SYNSET FOR THE WORD- THIS IS THE MOST COMMON MEANING

IF POS_SCORE>NEG_SCORE, USE POS_SCORE AS WEIGHT

IF POS_SCORE<NEG_SCORE, USE -NEG_SCORE AS WEIGHT

{'THE', 'WORST', 'THING', 'IN', 'THE', 'WORLD'} VOCABULARY

{'THE', 'WORST', 'THING'} TWEET

YOU COULD ALSO USE THE AVERAGE OF THE SCORES FOR ALL SYNSETS

(0, -1, 0, 0, 0, 0) FEATURE VECTOR

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

WE'LL DO THE CLASSIFICATION IN 2 WAYS :

NAIVE BAYES CLASSIFICATION

SUPPORT VECTOR MACHINES

THE FEATURE VECTOR IS SLIGHTLY DIFFERENT IN BOTH

2. FOR EACH OF THESE TWEETS, USE A MACHINE LEARNING CLASSIFIER AND CLASSIFY IT AS POSITIVE/NEGATIVE

1. DOWNLOAD A CORPUS TO USE AS TRAINING DATA

PREPROCESS THE TWEETS BEFORE STEP 2

2. EXTRACT FEATURES FROM BOTH THE TEST DATA (THE 100 TWEETS TO BE CLASSIFIED) AND THE TRAINING DATA

3. TRAIN A CLASSIFIER ON THE TRAINING DATA

4. USE THE CLASSIFIER TO CLASSIFY THE PROBLEM INSTANCES