



TEXT MINING

2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

DR. JING TANG

WHY TEXT IS IMPORTANT?

- Social networks (e.g., Facebook, Twitter, Instagram)
- Documents (e.g., email, SMS, reports)
- News (e.g., e-newspapers, CNN, Google News)

Leave Comments

10 comments

[Add a comment](#)



Jennifer Sable Lopez · Community Manager at SEOmoz

Looks like I'll be giving it another shot. :)

[Like](#) · [Reply](#) · [Subscribe](#) · 13 minutes ago



Rudy Lopez · Seattle, Washington

It looks like the iPhone app gets the upgrades first. The new desktop app is now just like it.

[Like](#) · [Reply](#) · 4 minutes ago

Jamie Steven · VP Marketing at SEOmoz

I've come to really love the client. Just wish they'd change the ugly icon!

[Like](#) · [Reply](#) · 59 seconds ago



[Tech](#) > [Computer](#)

Appeal court revives Oracle-Google copyright battle

Published: 10 May 2014 at 08:49 | Viewed: 480 | Comments: 0

Online news: [Computer](#)

Writer: [AFP](#)

1 0
[Tweet](#) [g+1](#)

[Print](#) [Email](#) [Share](#)

An appeals court has breathed new life into Oracle's big-money lawsuit against Google by ruling that software commands can be copyrighted just like classic books.



An appeals court has breathed new life into Oracle's big-money lawsuit against Google by ruling that

The case stems from 2012 trial, in which Oracle claimed Google owed them billions in damages for using parts of the Java programming language in its Android smartphone operating system.

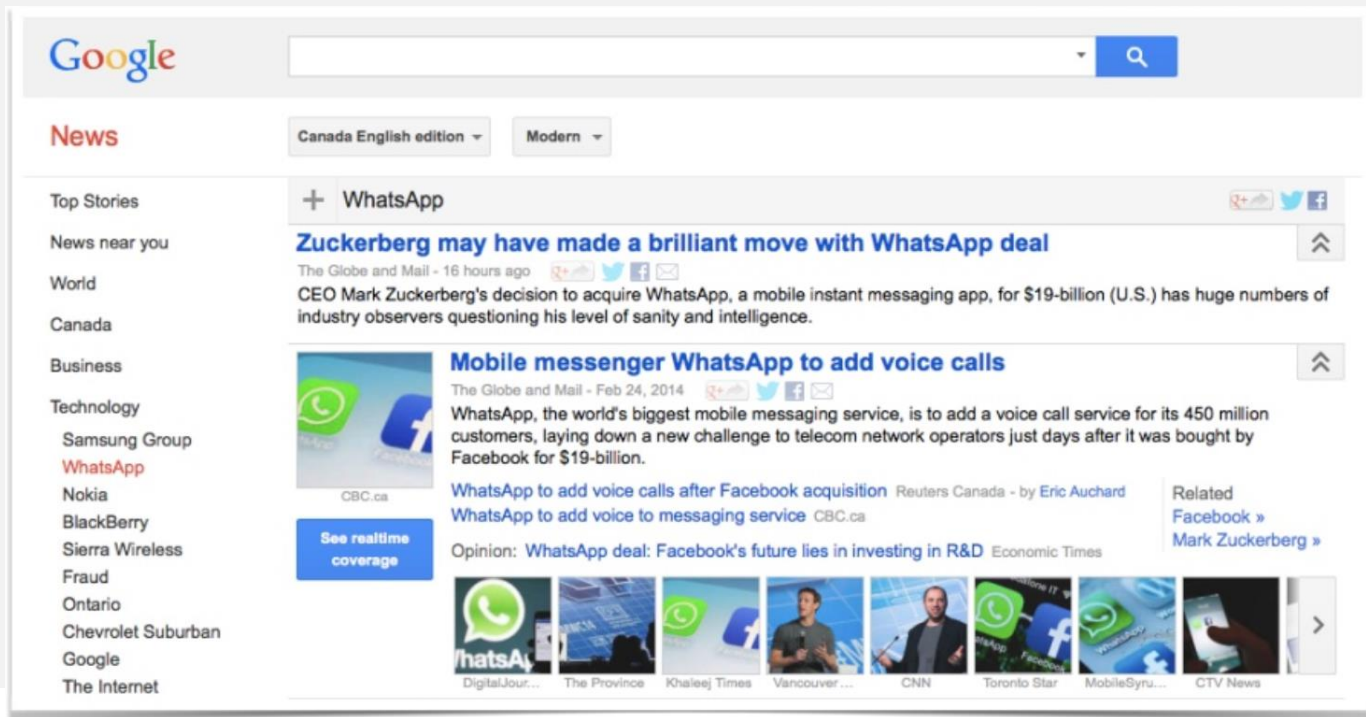
The case is being closely watched in Silicon Valley, where some champions of Internet freedom worry that extending copyright protection to these bits of code, called application programming interfaces, or APIs, would threaten innovation.

A panel of three judges in the US Federal Circuit Court of Appeals concluded that the trial court in 2012 erred and that it is bound to afford APIs protection under copyright laws "until either the Supreme Court or Congress tells us otherwise."

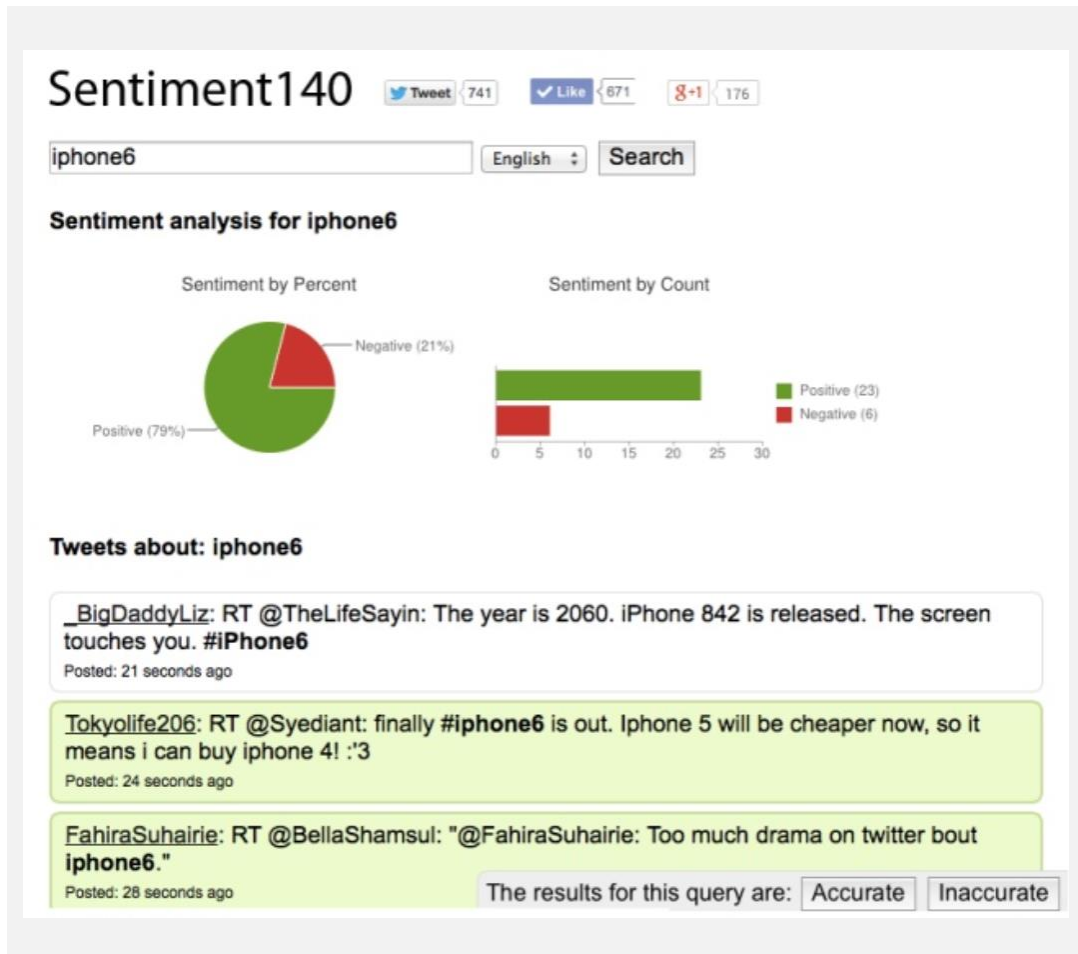
"We're disappointed -- and worried," the Electronic Frontier Foundation (EFF) said in a blog post about the appeals court

APPLICATIONS

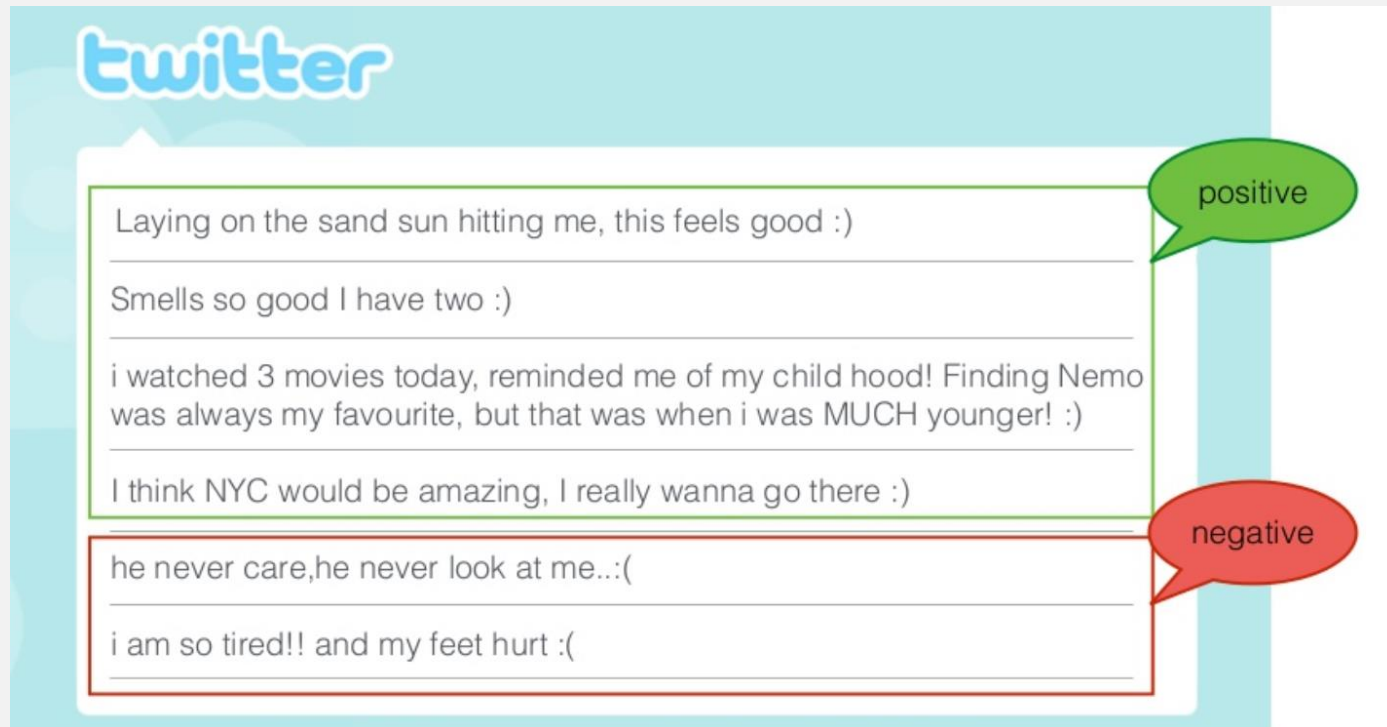
- Find webpage or news that have similar content to given keywords



APPLICATIONS: SENTIMENT ANALYSIS



TEXT REPRESENTATION: CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA



CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA (1)

- Frequency of words in texts

ID	Tweet	Sentiment	Term	Count
1	Laying on the sand sun hitting me, this feels good.	positive	Finding	1
2	Smells so good I have two.	positive		
3	i watched 3 movies today, reminded me of my child hood! Finding Nemo was always my favourite, but that was when i was MUCH younger!	positive		
4	I think NYC would be amazing, I really wanna go there.	positive		
5	he never care,he never look at me.	negative		
6	i am so tired!! and my feet hurt.	negative		

CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA (2)

- Frequency of words in texts

ID	Tweet	Sentiment	Term	Count
1	Laying on the sand sun hitting me, this feels good.	positive	Finding	1
2	Smells so good I have two.	positive	I	3
3	i watched 3 movies today, reminded me of my childhood! Finding Nemo was always my favourite, but that was when i was MUCH younger!	positive		
4	I think NYC would be amazing, I really wanna go there.	positive		
5	he never care,he never look at me.	negative		
6	i am so tired!! and my feet hurt.	negative		

CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA (3)

- Frequency of words in texts

ID	Tweet	Sentiment
1	Laying on the sand sun hitting me, this feels good.	positive
2	Smells so good I have two.	positive
3	i watched 3 movies today, reminded me of my child hood! Finding Nemo was always my favourite, but that was when i was MUCH younger!	positive
4	I think NYC would be amazing, I really wanna go there.	positive
5	he never care,he never look at me.	negative
6	i am so tired!! and my feet hurt.	negative



Bag of words

Term	Count	Term	Count
Finding	1	child	1
I	3	favourite	1
Laying	1	feels	1
MUCH	1	feet	1
NYC	1	go	1
Nemo	1	good	2
Smells	1	have	1
always	1	he	2
am	1	hitting	1
amazing	1	hood	1
and	1	hurt	1
at	1	i	3
be	1	look	1
but	1	me	1
care	1	...	1

CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA (4)

- Convert the words into roots (e.g., finding → find)

Bag of words

Term	Count	Term	Count
Finding	1	find	1
I	3	i	6
Laying	1	lai	1
MUCH	1	much	1
NYC	1	nyc	1
Nemo	1	nemo	1
Smells	1	smell	1
always	1	alwai	1
am	1	am	1
amazing	1	amaz	1
and	1	and	1
at	1	at	1
be	1	be	1
but	1	but	1
care	1	care	1

Bag of words

Term	Count	Term	Count
child	1	child	1
favourite	3	favourit	3
feels	1	feel	1
feet	1	feet	1
go	1	go	1
good	2	good	2
have	1	have	1
he	2	he	2
hitting	1	hit	1
hood	1	hood	1
hurt	1	hurt	1
i	3	i	3
look	1	look	1
me	1	me	1
...	1	...	1

CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA (5)

- Remove stop words (common words)

Term	Count	Term	Count
find	1	child	1
i	6	favourit	3
lai	1	feel	1
much	1	feet	1
nyc	1	go	1
nemo	1	good	2
smell	1	have	1
alwai	1	he	2
am	1	hit	1
amaz	1	hood	1
and	1	hurt	1
at	1		
be	1	look	1
but	1	me	1
care	1	...	1



Term	Count	Term	Count
find	1	hood	1
i	6	hurt	1
lai	1	look	1
nyc	1	care	1
nemo	1	movi	1
smell	1	reali	1
alwai	1	remind	1
amaz	1	sand	1
child	1	sun	1
favorit	1	thi	1
feel	1	think	1
feet	1	tire	1
go	1	today	1
good	2	wa	3
hit	1	watch	1

CONVERT UNSTRUCTURED DATA TO STRUCTURED DATA (6)

- Binary occurrence

ID	Tweet	Sentiment
1	Laying on the sand sun hitting me, this feels good.	positive
2	Smells so good I have two.	positive
3	i watched 3 movies today, reminded me of my child hood! Finding Nemo was always my favourite, but that was when i was MUCH younger!	positive
4	I think NYC would be amazing, I really wanna go there.	positive
5	he never care,he never look at me.	negative
6	i am so tired!! and my feet hurt.	negative



← attribute →									← label →
ID	find	I	lai	nyc	nemo	smell	alwai	...	Sentiment
1	0	0	1	0	0	0	0	...	positive
2	0	1	0	0	0	1	0	...	positive
3	1	1	0	0	1	0	1	...	positive
4	0	1	0	1	0	0	0	...	positive
5	0	0	0	0	0	0	0	...	negative
6	0	1	0	0	0	0	0	...	negative

WORD IMPORTANT MEASURES (1)

- Term frequency (TF)**

- TF(t) = word t frequency / all words in a message

$$1/6 = 0.17$$

ID	Tweet	Sentiment
1	Laying on the sand sun hitting me, this feels good.	positive
2	Smells so good I have two.	positive
3	i watched 3 movies today, reminded me of my child hood! Finding Nemo was always my favourite, but that was when i was MUCH younger!	positive
4	I think NYC would be amazing, I really wanna go there.	positive
5	he never care,he never look at me.	negative
6	i am so tired!! and my feet hurt.	negative



attribute							label
ID	find	I	lai	nyc	nemo	...	Sentiment
1	0	0	0.17	0	0	...	positive
2	0	0.17	0	0	0	...	positive
3	0.17	0.33	0	0	0.17	...	positive
4	0	0.33	0	0.17	0	...	positive
5	0	0	0	0	0	...	negative
6	0	0.17	0	0	0	...	negative

WORD IMPORTANT MEASURES (2)

- Inverse document frequency (IDF)**

- Measure how important a term is $IDF(t) = \log_{10} \frac{N}{n}$

attribute									label
ID	find	I	lai	nyc	nemo	smell	alwai	...	Sentiment
1	0	0	1	0	0	0	0	...	positive
2	0	1	0	0	0	1	0	...	positive
3	1	1	0	0	1	0	1	...	positive
4	0	1	0	1	0	0	0	...	positive
5	0	0	0	0	0	0	0	...	negative
6	0	1	0	0	0	0	0	...	negative

N = Total number of documents

n = Total number of documents with term t

$$IDF(\text{find}) = \log_{10} \left(\frac{6}{1} \right)$$

$$= \log_{10} 6 = 0.78$$

term	find	I	lai	nyc	nemo	smell	alwai
IDF	0.78	0.18	0.78	0.78	0.78	0.78	0.78

WORD IMPORTANT MEASURES (3)

- **Term frequency - Inverse document frequency (TF-IDF)**
 - A measure of how much information the word provides

TF table

attribute							label
ID	find	I	lai	nyc	nemo	...	Sentiment
1	0	0	0.17	0	0	...	positive
2	0	0.17	0	0	0	...	positive
3	0.17	0.33	0	0	0.17	...	positive
4	0	0.33	0	0.17	0	...	positive
5	0	0	0	0	0	...	negative
6	0	0.17	0	0	0	...	negative



term	find	I	lai	nyc	nemo
TF-IDF	0.000	0.000	0.133	0.000	0.000
	0.000	0.031	0.000	0.000	0.000
	0.133	0.059	0.000	0.000	0.133
	0.000	0.059	0.000	0.133	0.000
	0.000	0.000	0.000	0.000	0.000
	0.000	0.031	0.000	0.000	0.000

term	find	I	lai	nyc	nemo	smell	alwai
IDF	0.78	0.18	0.78	0.78	0.78	0.78	0.78

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

$$\begin{aligned} \text{TF-IDF}(\text{find in doc. 3}) &= 0.17 \times 0.78 \\ &= 0.133 \end{aligned}$$

EXAMPLE 1:

- When a 100-word document A contains the term “cat” 12 times. When a 100-word document B contains the term “cat” 6 times. When a 100-word document C contains the term “cat” 0 times. When a 100-word document D contains the term “cat” 0 times. Please calculate the TF-IDF of the term “cat” in each document.

TF-IDF

- **High** TF-IDF refers to a **high rarity** of the term.
- If TF-IDF applied to a search engine, it brings benefits as:
 - stop worrying about using the **stop-words**
 - Stop-words: the most common used but unimportant words
 - successfully hunt words with **higher search volumes** and **lower competition**
 - be sure to have words that make your content **unique and relevant** to the user, etc

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

WORD IMPORTANT MEASURES (4)

- Analysis of sequential words
 - Unigram (1 words)
 - Bi-gram (2 words)
 - Tri-gram (3 words)

The diagram illustrates the process of extracting sequential word features from a sentence. On the left, an orange box contains the sentence "Smells so good I have two." with the word "Smells" highlighted by a blue box. A blue arrow points from this box to the first row of the table. A red arrow points from the sentence box to the table. The table on the right has three columns: "unigram" (blue), "bi-gram" (red), and "tri-gram" (green). The rows show the extraction of 1-word, 2-word, and 3-word sequences from the sentence.

unigram	bi-gram	tri-gram
smells	smells so	smells so good
so	so good	so good I
good	good I	good I have
I	I have	I have two
have	have two	
two		

WORD IMPORTANT MEASURES (5)

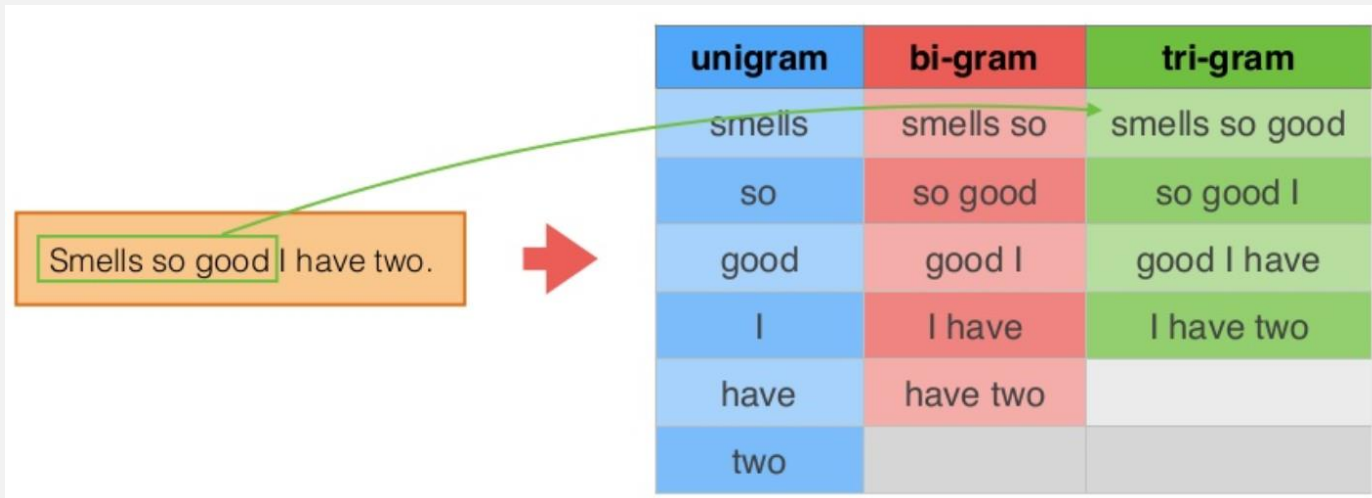
- Analysis of sequential words
 - Unigram (1 words)
 - Bi-gram (2 words)
 - Tri-gram (3 words)

The diagram illustrates the process of extracting sequential word features from a sentence. On the left, the sentence "Smells so good I have two." is shown in an orange box. A red arrow points from the words "Smells so" in this sentence to the first row of the table on the right. The table is organized into three columns: unigram (blue), bi-gram (red), and tri-gram (green). The rows represent the sequential pairs of words in the sentence, with the last two rows having empty cells for bi-grams and tri-grams as they do not have enough subsequent words.

unigram	bi-gram	tri-gram
smells	smells so	smells so good
so	so good	so good I
good	good I	good I have
I	I have	I have two
have	have two	
two		

WORD IMPORTANT MEASURES (6)

- Analysis of sequential words
 - Unigram (1 words)
 - Bi-gram (2 words)
 - Tri-gram (3 words)





N-GRAMS

- Identify words and phrases
- Reveal relationships between words

EXAMPLE: SENTIMENT ANALYSIS OF MOVIE REVIEWS

- 400 Reviews (community samples → community dataset → entertainment → sentiments of 4000 movie reviews)

Open in  Turbo Prep  Auto Model Filter (400 / 400 examples): all

Row No.	id	label	text
1	cv000_29590	pos	films adapte...
2	cv001_18431	pos	ever
3	cv002_15918	pos	you
4	cv003_11664	pos	" ja
5	cv004_11636	pos	mov
6	cv005_29443	pos	on j
7	cv006_15448	pos	app
8	cv007_4968	pos	one
9	cv008_29435	pos	afte
10	cv009_29592	pos	the
11	cv010_29198	pos	afte
12	cv011_12166	pos	i've
13	cv012_29576	pos	sync
14	cv013_10159	pos	sync
15	cv014_13924	pos	the police n...
16	cv015_29439	pos	plot : a youn...
17	cv016_4659	pos	carry on ma...
18	cv017_22464	pos	the ultimate ...

ExampleSet (400 examples, 2 special attributes, 1 regular attribute)

ANALYSIS STEPS

1. Data Preparation & Cleaning

- Import data
- Split data
- TF_IDF Calculation (Text Vectorization)

2. Data Visualization & Analysis

- None

3. Deep Learning Training:

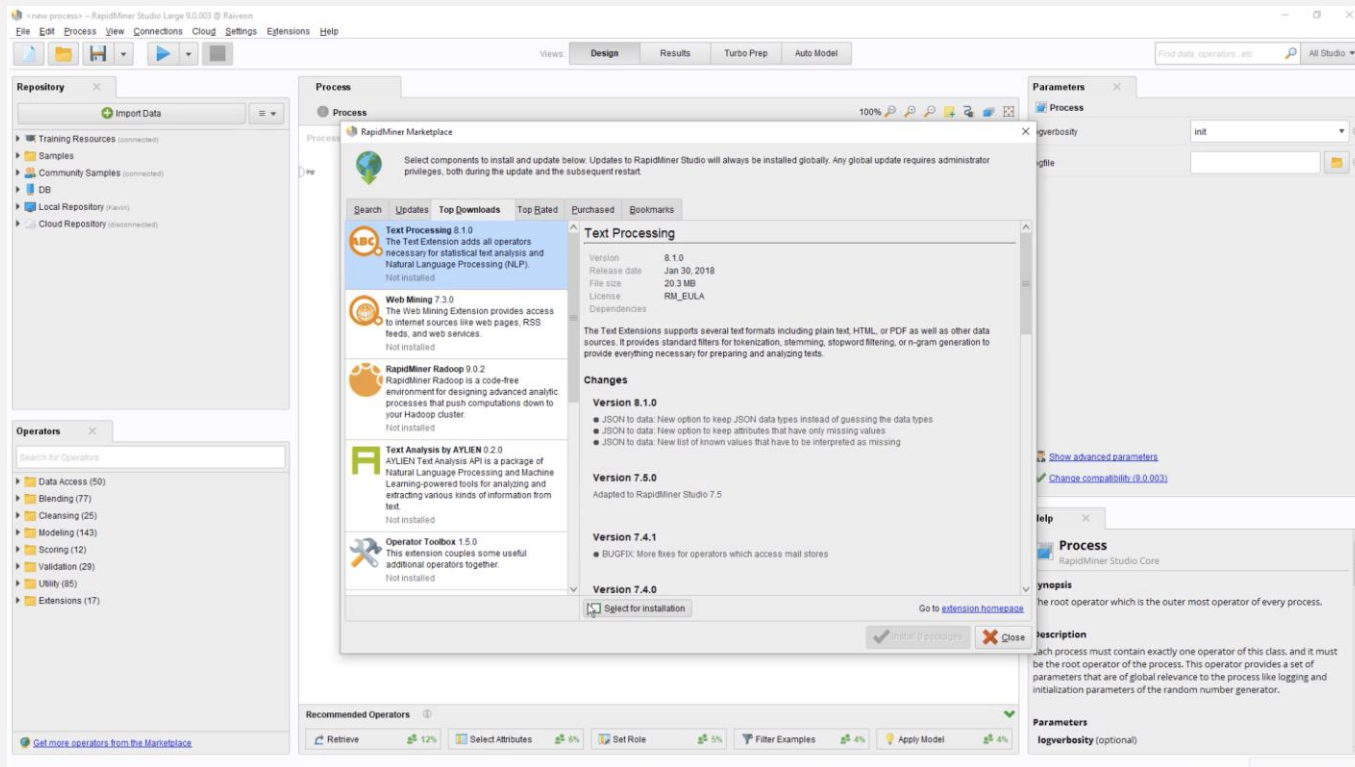
- SVM (any type of classification)

4. Testing & Evaluation:

Confusion Matrix, Accuracy

TEXT PROCESSING EXTENSION IN RAPIDMINER

- <https://academy.rapidminer.com/learn/video/loading-text-into-rapidminer>



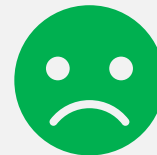
DATA PREPROCESSING

- Transform cases
- Tokenize
- Filter Tokens
- Filter Stopwords
- Stemming (Porter)

HW_14: SENTIMENT PREDICTION BASED ON US AIRLINE TWEETS

- A sentiment analysis of 14,427 tweets about US airline

1	Tweet_id	5	Negativereason_confidence
2	Airline_sentiment	6	Airline
3	Airline_sentiment_confidence	7	Test
4	Negativereason		



ANALYSIS STEPS

1. Data Preparation & Cleaning

- Import data
- Select attributes (id, sentiment, and text)
- Split data
- Text Preprocessing (text vectorization, or following page 24)

2. Data Visualization & Analysis

- Airline Ranking based on Num. of positive and negative tweets, etc.

3. Deep Learning Training:

- SVM (any type of classification)

4. Testing & Evaluation:

Confusion Matrix, Accuracy