Text Mining

CCSE 634 Data Mining

Professor Anita Wasilewska

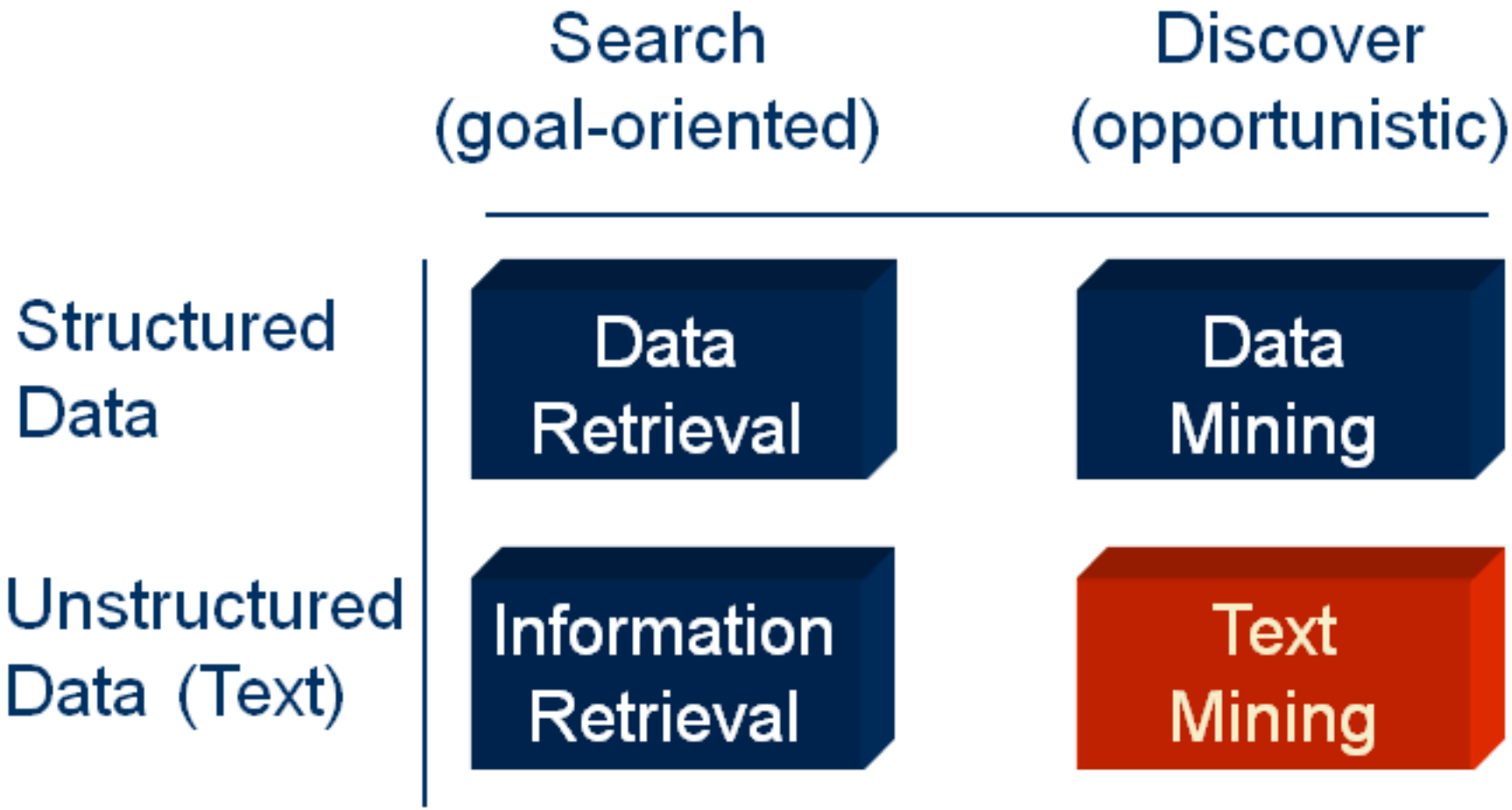
References

- Jiawei Han and Michelle Kamber. Data Mining Concept and Techniques Morgan Kaufman Publishers, 2003, 2011
- https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html
- http://stp.lingfil.uu.se/~shaooyan/textanalys17/TokeninsationSegmentation.pdf
- https://sites.duke.edu/lit80s_02_f2013_augrealities/text-visualization-see-more-than-texts/
- https://en.wikipedia.org/wiki/Tf%E2%80%93idf
- http://www.tfidf.com/
- https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html
- http://www.nltk.org/
- https://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Overview

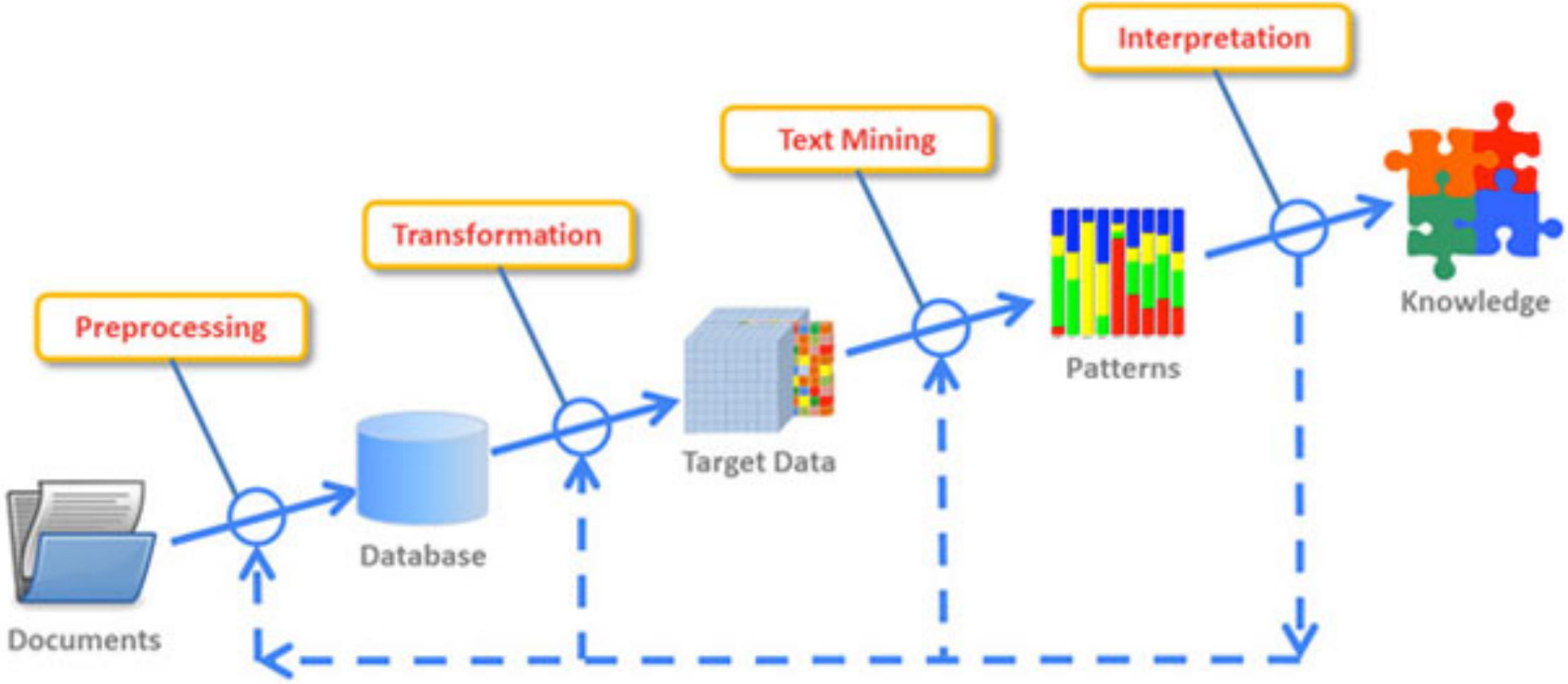
- Introduction to Text Mining
- Text Mining Process
- Visualization
- Research Paper
  - VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text
  - Sentiment Analysis Demo

What is Text Mining



Source: Avaquest Inc

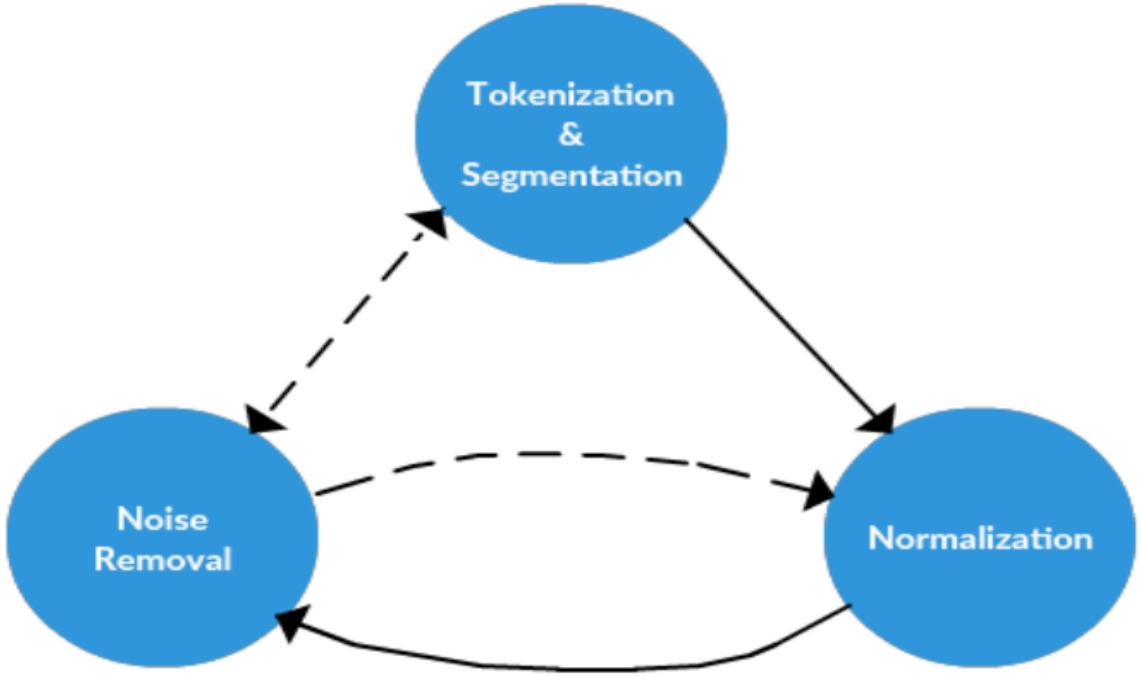
Text Mining Process



https://

TEXT PREPROCESSING

- Perform basic transformations(shown in the image below).
- Obtain data that is suitable to perform analysis on.



The text data preprocessing framework.

https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html

SEGMENTATION

- **Definition**: find sentence boundaries between words in different sentences.
- Non-trivial task as “.” is ambiguous.
  - Dr. Ford did not ask Col. Mustard the name of Mr. Smith's dog.
  - "What is all the fuss about?" asked Mr. Peters.
- A rule-based approach can be implemented for this problem(also known as sentence boundary disambiguation).
  - Consider the context of punctuation into account to create a finite state transducer (a finite state machine with 2 memory tapes).

TOKENIZATION

- **Definition** : Splitting text into linguistic units, such as words, punctuations and numbers.
- **Example** : Bob eats apples.

- Contains 4 tokens: Bob|eats|apples|.

- Non trivial task - Should we rely only on spaces to delimit the sentence?
    - New York and San Francisco.
- Can be processed using a rule-based approach.

NORMALIZATION

- **Definition**: Text normalization is the process of transforming text into a single canonical form that it might not have had before.
- 3 main tasks - Stemming, Lemmatization, Stop word removal
- Stemming - process of eliminating affixes (suffixes, prefixes, infixes, circumfixes) from a word in order to obtain a word stem.

- Example - running → run

- Lemmatization - capture canonical forms based on a word's lemma.

- Example - better → good

NORMALIZATION

- Stop word removal - filter out words that contribute little to overall meaning, given that they are generally the most common words in a language.
- Example : ~~The~~ quick brown fox jumps over ~~the~~ lazy dog.

NOISE REMOVAL

- A document scraped from the world wide web will be wrapped in HTML or XML tags.
- Remove text file headers, footers
- Remove HTML, XML markup and metadata

FEATURE GENERATION

- A text is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.
- **Example** : John also likes to watch football games.
- After pre-processing this text, the features generated by bag of words method can be represented as :

- {"John":1,"likes":1,"watch":1,"football":1,"games":1};

- Where each comma separated entry is of the form “word”:frequency in the sentence.

Feature Selection

- Term Frequency
- Document Frequency
- Inverse Document Frequency
- Term Frequency-Inverse Document Frequency

Term Frequency

- The **term frequency** or tf(*t*,*d*), which measures how frequently a term occurs in a document.
- The simplest choice is to use the *raw count* of a term in a document, i.e. the number of times that term t occurs in document *d.*

Ways to compute tf(t,d)

Variants of term frequency (TF) weight

weighting scheme	TF weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

f(t,d) is frequency of term in a document.

0<K<1

https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Document Frequency

- The document frequency df(t), defined to be the number of documents in the collection that contains a term.
- This is because in trying to discriminate between documents for the purpose of scoring it is better to use a document-level statistic (such as the number of documents containing a term) than to use a collection-wide statistic for the term

Word	cf	df
try	10422	8760
insurance	10440	3997

Inverse Document Frequency

- The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.
- While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance.

Where N is the number of documents and t is the term for which we define the inverse document frequency.

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Term frequency –Inverse Document Frequency

- Term frequency-inverse document frequency or tf-idf, and the tf-idf weight is a weight often used in information retrieval and text mining.
- This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.
- The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

where t is term and d is document.

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

https://en.wikipedia.org/wiki/Tf%E2%80%93idf

- highest when term (t) occurs many times within a small number of documents (thus lending high discriminating power to those documents);
- lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
- lowest when the term occurs in virtually all documents.

Text Mining Process

- Document Clustering
- Text Categorization
- Text Clustering
- Sentiment Analysis

Document Clustering

- Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process. Examples of document clustering include web document clustering for search users.

Application of Document Clustering

- The application of document clustering can be categorized to two types, online and offline.
- Online applications are usually constrained by efficiency problems when compared to offline applications.
- Document clustering may be used for different tasks, such as grouping similar documents (news, tweets, etc.) and the analysis of customer/employee feedback, discovering meaningful implicit subjects across all documents.

Sentiment Analysis

- **Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.**
- A common use case for this technology is to discover how people feel about a particular topic.

VISUALIZATION

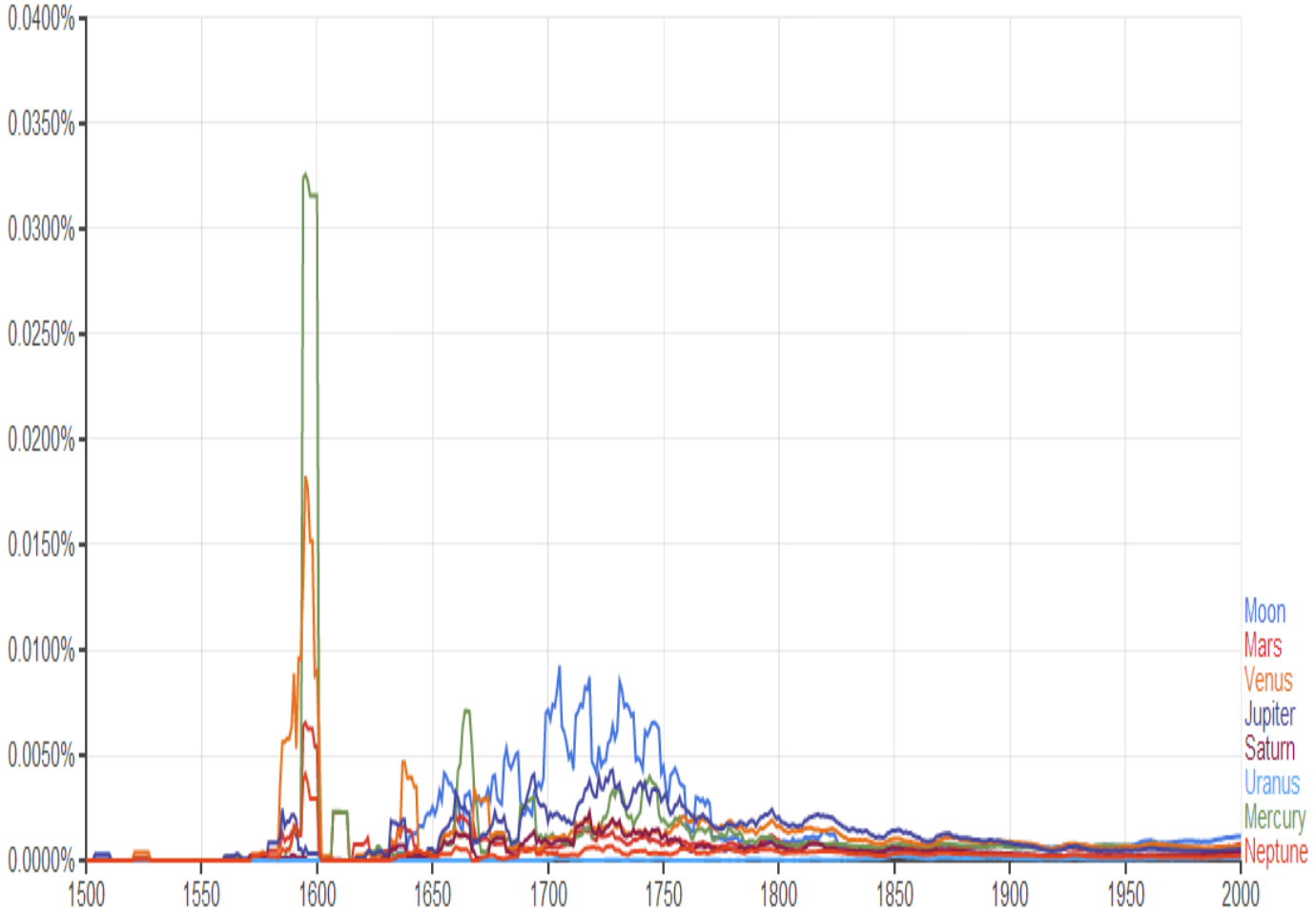
- Text can be stored in the following formats:
  - Raw data
  - Corpus
  - Document-term matrix - This is a matrix where each row represents one document (such as a book or article), each column represents one term, and each value (typically) contains the number of appearances of that term in that document.

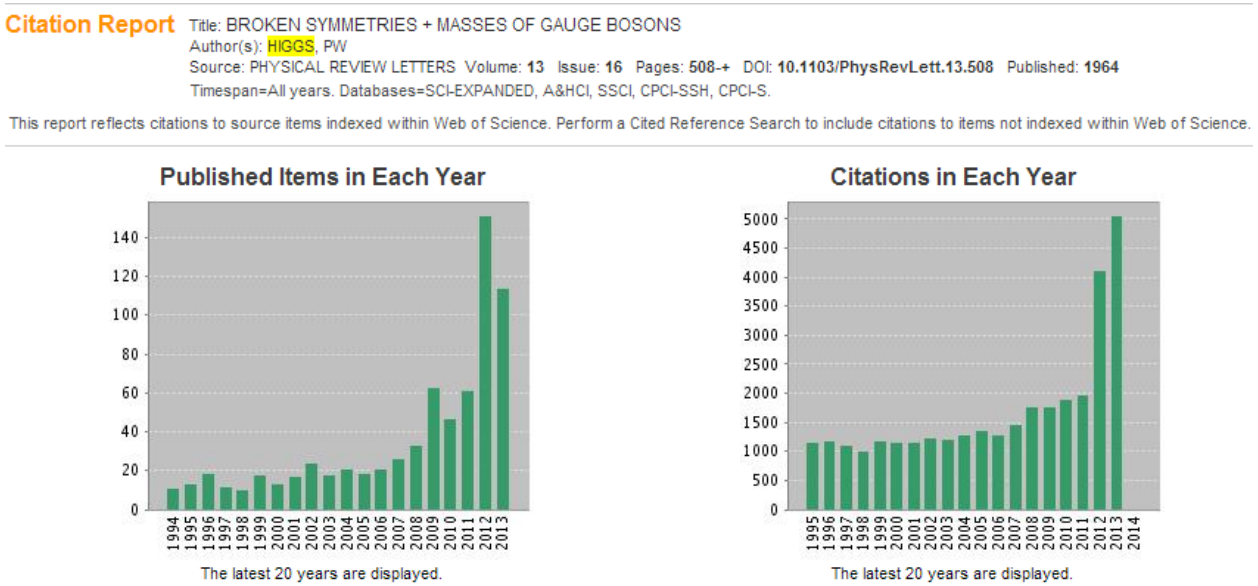
Why Visualization?

- Intuitive and interactive data visualization allows decision makers to immediately grasp what the analysis reveals. Visualization tools help companies:
  - Make sense of data
  - Analyze information in a simple and interactive way
  - Discover trends, insights and hidden relationships between concepts
  - Display and share information and quickly create reports

Types of visualizations:

- Graph - As for the use of graphs to do text visualization, the most successful example is Google Ngram Viewer. When the user types in several words or phrases, the tool will search the database to find matches and generate a graph to show the relation between different parameters. Example:



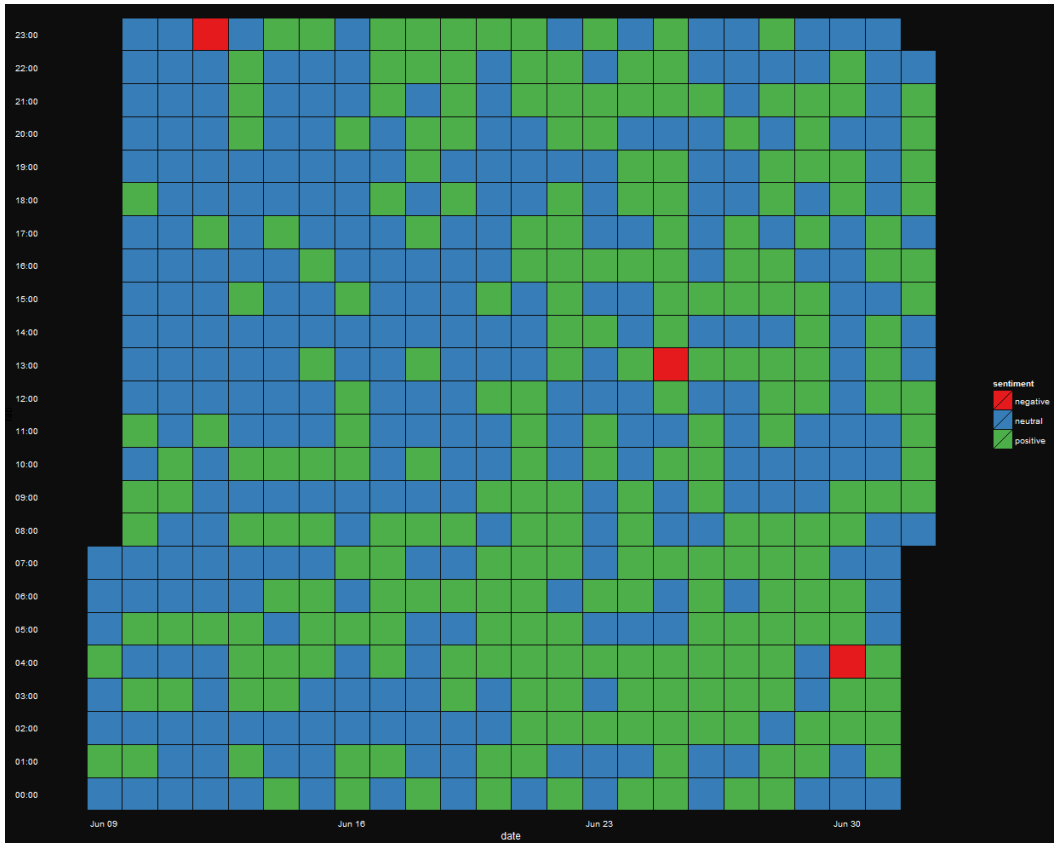


- 2. Chart - It is a very common method to do data visualization by using of charts in reports, scientific papers, blogs etc. There are several different kinds of charts such as the pie chart, bar chart, bubble chart and scattering chart for different uses. Example:

https://sites.duke.edu/lit80s_02_f2013_augrealities/text-visualization-see-more-than-texts/

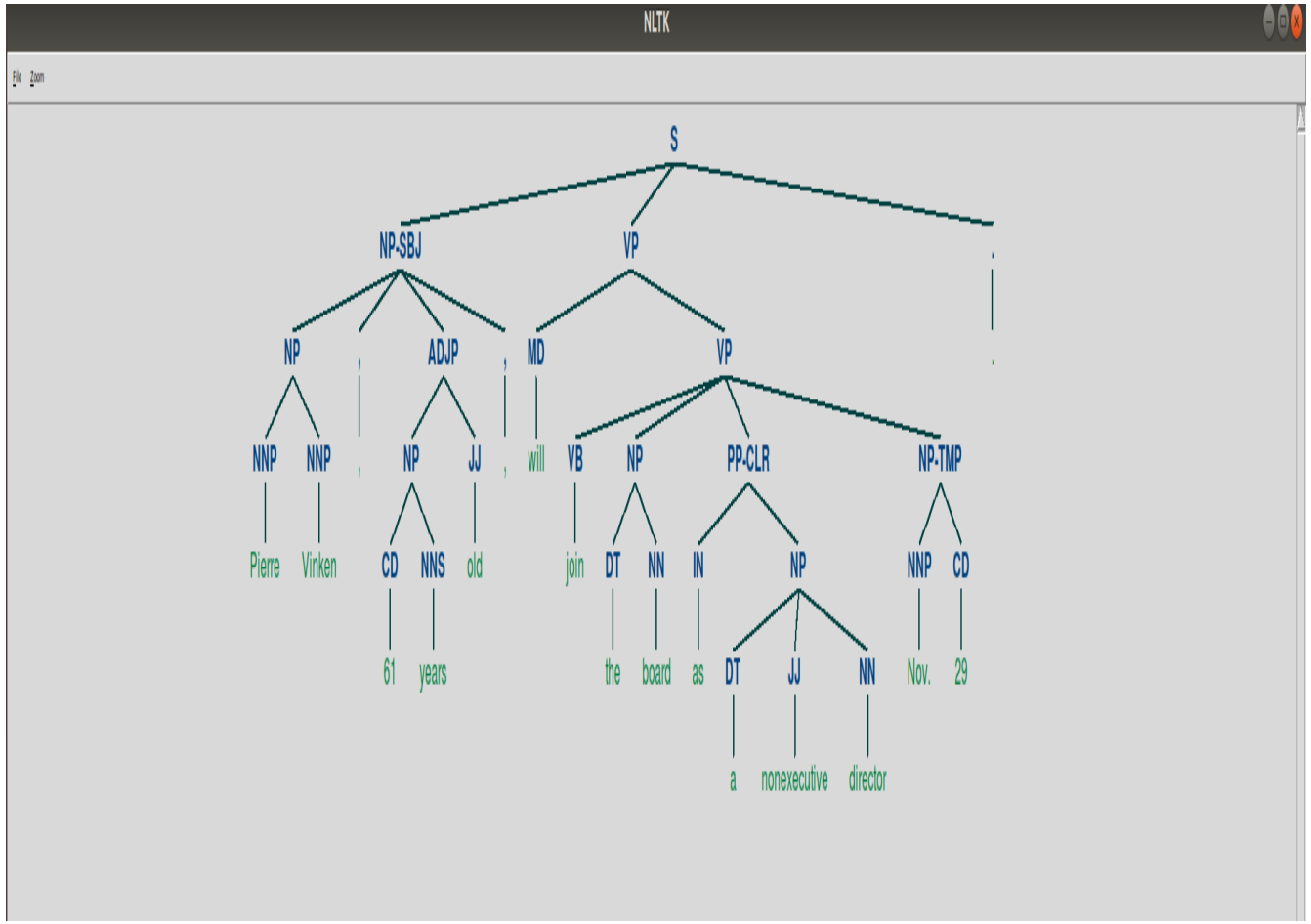
- 3. Word Cloud - By making words stand out either by means of font size or color according to their usage frequency, Word Cloud has its significance in both text analysis and digital humanities scholarship. Example:

https://sites.duke.edu/lit80s_02_f2013_augrealities/text-visualization-see-more-than-texts/



- 4. Map - Maps have always been an important tool for geoscience. There are mainly two different kinds of maps in text visualization: geographic map and abstract map.

https://socialfunction.wordpress.com/2014/08/23/heatmap/



- 5. Network - A Network is used to show the relations between different units that make up the whole network. One important form of the network is tree structure which pays more attention to the relations between leading parts and subparts.

In [45]:
```
import nltk
```

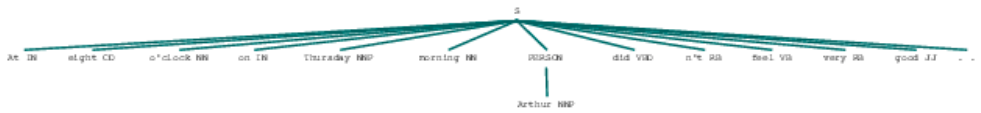
In [50]:
```
sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
tokens = nltk.word_tokenize(sentence)

tagged = nltk.pos_tag(tokens)
```

In [51]:
```
entities = nltk.chunk.ne_chunk(tagged)
```

In [52]:
```
entities
```

Out[52]:



In [53]:
```
tagged
```

Out[53]:
```
[('At', 'IN'),
 ('eight', 'CD'),
 ("o'clock", 'NN'),
 ('on', 'IN'),
 ('Thursday', 'NNP'),
 ('morning', 'NN'),
 ('Arthur', 'NNP'),
 ('did', 'VBD'),
 ("n't", 'RB'),
 ('feel', 'VB'),
 ('very', 'RB'),
 ('good', 'JJ'),
 ('.', '.')]
```

In [ ]:
```
from nltk.corpus import treebank
t = treebank.parsed_sents('wsj_0001.mrg')[0]
t.draw()
```

In [ ]:
```

```

VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text

C.J. Hutto **Eric Gilbert**

Georgia Institute of Technology, Atlanta, GA 30032

cjhutto@gatech.edu gilbert@cc.gatech.edu

8th International Conference on Weblogs and Social Media, ICWSM 2014

Why did we choose this paper?

VADER model is considered GOLD-Standard quality in Sentiment Analysis

What is Sentiment Analysis?

Extracting the emotions and opinions from a piece of text

Why is there so much boom about it?

- Understanding the demands of customers
- Understanding the feedbacks of customers

“The party is wonderful.”

&

“I hate that man.”

“The party is wonderful.” → Positive Emotion

“I hate that man.” → Negative Emotion

Why ?

Two approaches for doing Sentiment Analysis

1. Lexical Approach
2. Machine Learning Approach

Lexical Approach

By using “lexicons”

Sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative

1. Polarity Based Lexicons

Ex: LIWC (Linguistic Inquiry and Word Count)

Drawbacks:

- Unable to identify the intensity of words
- Doesn’t include acronyms, emoticons, slangs

2. Valence-based Lexicons

Ex: ANEW (Affective Norms for English Words)

Drawbacks

- Doesn’t include acronyms, emoticons, slangs
  i.e insensitive to Social media texts

Machine Learning Approach Naive Bayes, Maximum Entropy, Support Vector Machine

Drawbacks

1. Require extensive training data covering all features
2. Computationally expensive

VADER (Valence Aware Dictionary and sEntiment Reasoner)

1. Constructing and Validating a Valence-Aware Sentiment Lexicon: A Human-Centered Approach

2. Identifying Generalizable Heuristics Humans Use to Assess Sentiment Intensity in Text

3. Testing the accuracy in Multiple Domain Contexts: Social media, Movie reviews, Product reviews, News articles.

Constructing and Validating a Valence-Aware Sentiment Lexicon: A Human-Centered Approach

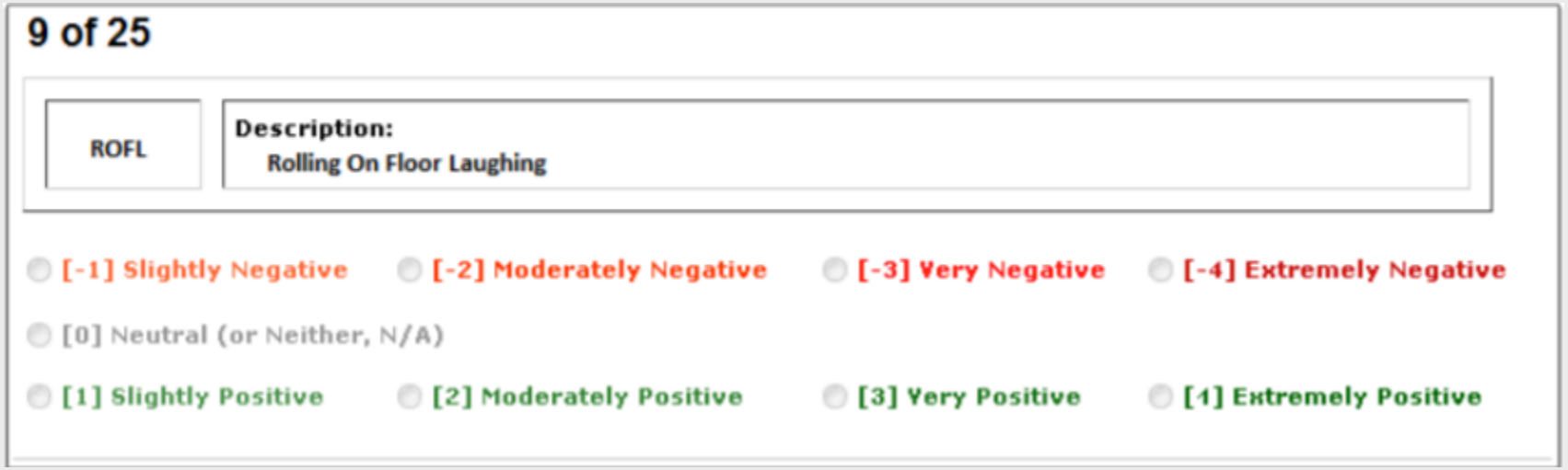
English experts from AMT(Amazon Mechanical Turk)

Rating the intensity on a scale of (-4 to +4)

1. Existing sentiment word-banks – (LEWIS, ANEW)
2. Emoticons, slangs
   Ex: "☺️", "LOL", "WTF"

Ensuring Quality of Ratings by AMT workers

1. Reading Comprehension Test
2. Sentiment rating test for pre-validated words
3. “Golden” items in each batch



Identifying Generalizable Heuristics Humans Use to Assess Sentiment Intensity in Text

QUALITATIVE ANALYSIS

Punctuation

“The food here is good!!!” ➔ More Intensive

than,

“The food here is good.”

Capitalization

“The food here is GREAT!” → More intensive

Than,

“The food here is great!”

Degree Modifiers

“The service here is extremely good” ➔ More intensive

than

“The service here is good”

Polarity Shifters

I love you, but I don't want to be with you anymore

The part after "but" is the dominant factor for deciding the sentiment here.

Polarity Negators

“The food here is great”

Negated by “Not”

“The food here is not great”

Test Condition	Example Text
Baseline	Yay. Another good phone interview.
Punctuation1	Yay! Another good phone interview!
Punctuation1 + Degree Mod.	Yay! Another extremely good phone interview!
Punctuation2	Yay!! Another good phone interview!!
Capitalization	YAY. Another GOOD phone interview.
Punct1 + Cap.	YAY! Another GOOD phone interview!
Punct2 + Cap.	YAY!! Another GOOD phone interview!!
Punct3 + Cap.	YAY!!! Another GOOD phone interview!!!
Punct3 + Cap. + Degree Mod.	YAY!!! Another EXTREMELY GOOD phone interview!!!

! → Increased intensity by 0.292

All caps → Increased intensity by 0.733

But check → sub sentence before but has its intensity reduced to 50% and sub sentence after that, intensity increased to 150%

Quantifying the Emotion of a Sentence

$$x = \sum sentiment\ scores\ of\ each\ VADER - dictionary - listed\ word\ in\ the\ sentence$$

$$\boldsymbol{Normalized\ Compound\ Score}\ (\boldsymbol{c}) = \frac{x}{\sqrt{x^2 + \alpha}}$$

if c >= 0.05 → Positive sentiment

if c > -0.05 and c < 0.05 → Neutral sentiment

if c <= -0.05 → Negative sentiment

Testing the Testing the accuracy in Multiple Domain Contexts

1. Social media text:
   includes 4,000 tweets pulled from Twitter's public timeline

2. Movie reviews:
   includes 10,605 sentence-level snippets from rotten.tomatoes.com

3. Product reviews:
   includes 3,708 sentence level snippets from 309 customer reviews from Amazon

4. news articles:
   includes 5,190 sentence-level snippets from 500 New York Times opinion editorials

Test Results

	Tweets	Movie	Amazon	NYT
VADER	**0.96**	0.61	**0.63**	**0.55**
NB (tweets)	0.84	0.53	0.53	0.42
ME (tweets)	0.83	0.56	0.58	0.45
SVM-C (tweets)	0.83	0.56	0.55	0.46
SVM-R (tweets)	0.65	0.49	0.51	0.46
NB (movie)	0.56	**0.75**	0.49	0.44
ME (movie)	0.56	**0.75**	0.51	0.45
NB (amazon)	0.69	0.55	0.61	0.48
ME (amazon)	0.67	0.55	0.60	0.43
SVM-C (amazon)	0.64	0.55	0.58	0.42
SVM-R (amazon)	0.54	0.49	0.48	0.44
NB (nyt)	0.59	0.56	0.51	0.49
ME (nyt)	0.58	0.55	0.51	0.50

Performance Comparison

- SVM (without pre-trained) → Hours
- SVM (pre-trained model)→ 10 minutes
- VADER → Fraction of a second

Application:

Twitter Sentiment Analysis

https://datamining-demo.herokuapp.com/