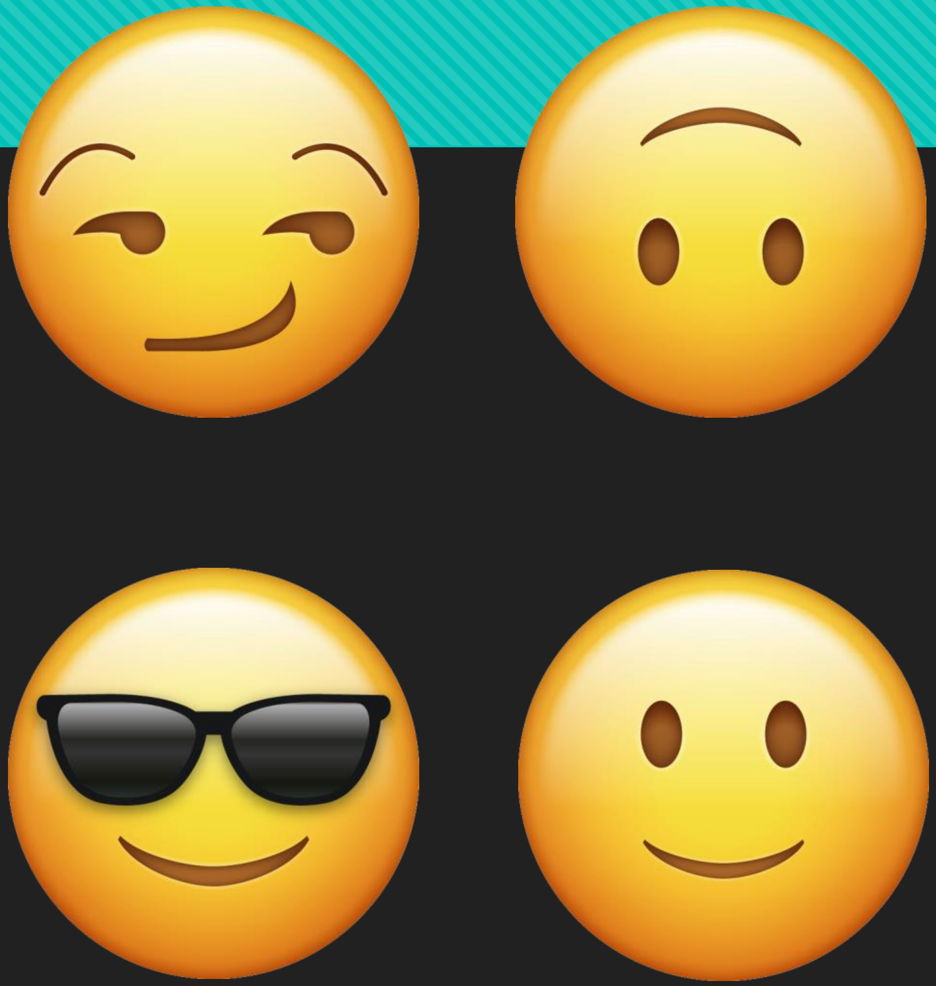
Emoji Prediction: A Survey of Classification Algorithms 🙃

Alexandra Gamez

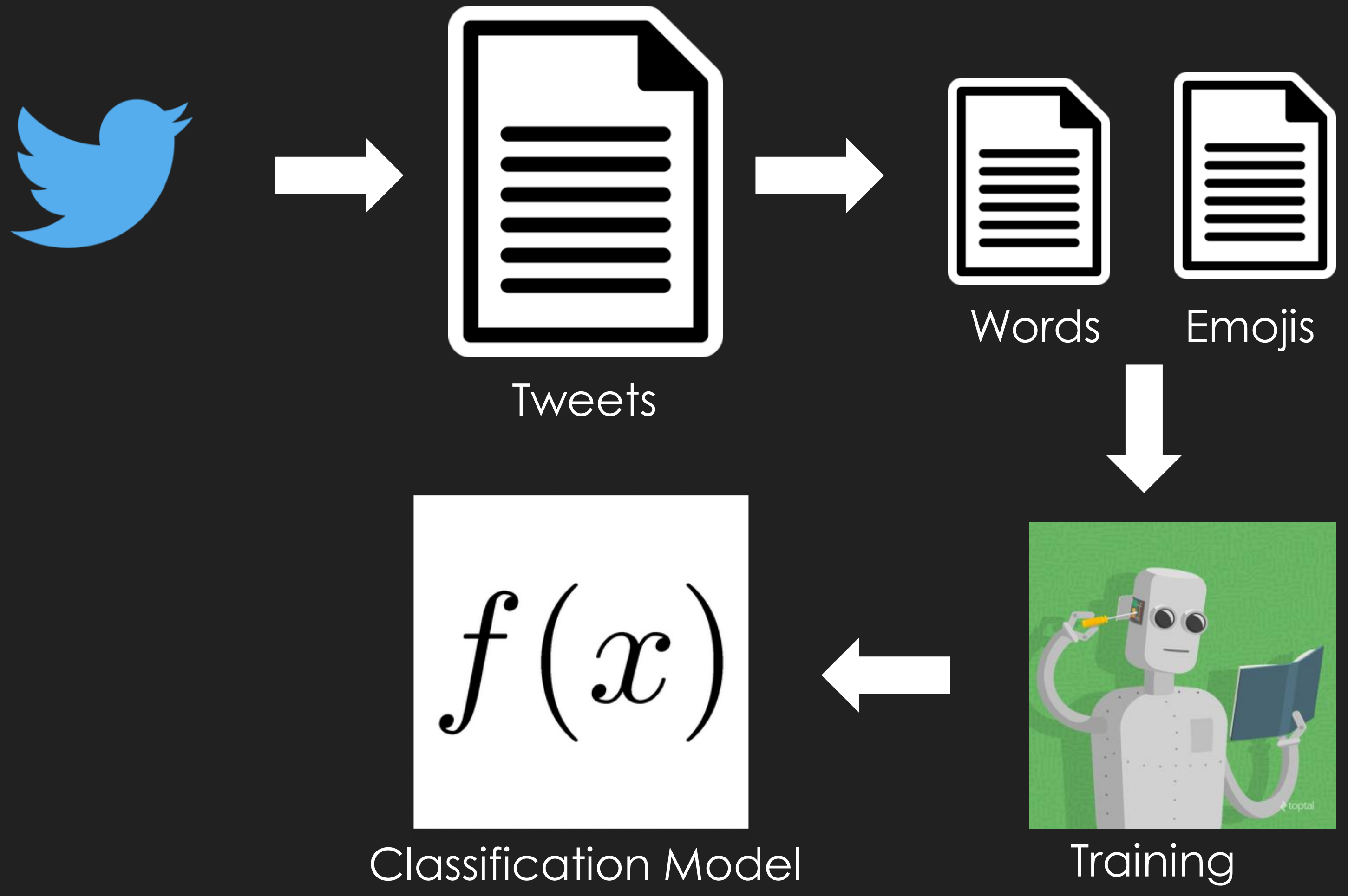
Background

- Twitter is a place where valuable information such as sentiments, popularity, and opinions of various topics are located
- A perfect place for Natural Language Processing
- Emoji prediction is a classification problem



Problem & General Approach

- Task
  - Emoji prediction
- Approach
  - Get data
  - Prepare data
  - Train data
  - Test



Dataset Example

Tweet content

75	PS I U @ Beaver Stadium
76	Get ready for a bunch of annoying pictures of Dallas @ Dallas, Texas
77	@user aww love ya laura!!!
78	Hoes never get cold @ Downtown Los Angeles
79	National Siblings Day #WeAreFamily #HappyNationalSiblingsDay #SistersLikeUs @ Time Square…
80	"Don't you hate working holidays?"... they asked. #roseparade2016 #ilovemyjob #colorfulfloats…
81	#taromilkteaboba for this hot LA day. @ McDonald's at 2810 South…
82	S N ~ They're saying it's the hottest day of the year today. So we're hiding in a cabana by the…
83	Our fierce party crew! @ Blarney Stone Pub
84	Got to visit my grandpa K today :) @ Greenwood, South Carolina
85	"what's in the ice box, if you don't mind me asking sir?".....Our Hearts …

Emoji Label

75	8
76	5
77	0
78	7
79	9
80	0
81	4
82	12
83	6
84	3
85	8

Emoji Mapping

0	❤	_red_heart_
1	😍	_smiling_face_with_hearteyes_
2	😂	_face_with_tears_of_joy_
3	💕	_two_hearts_
4	🔥	_fire_
5	😊	_smiling_face_with_smiling_eyes_
6	😎	_smiling_face_with_sunglasses_
7	✨	_sparkles_
8	💙	_blue_heart_
9	😘	_face_blowing_a_kiss_
10	📷	_camera_
11	US	_United_States_
12	☀	_sun_
13	💜	_purple_heart_
14	😉	_winking_face_
15	💯	_hundred_points_
16	😁	_beaming_face_with_smiling_eyes_
17	🎄	_Christmas_tree_
18	📸	_camera_with_flash_
19	😜	_winking_face_with_tongue_

Algorithms Used

- Naïve Bayes (Multinomial Naïve Bayes)
- Stochastic Gradient Descent
- Support Vector Machines
- Logistic Regression
- K Nearest Neighbors
- Decision Tree
- Neural Networks
- My Naïve Bayes ***

Naïve Bayes

Likelihood

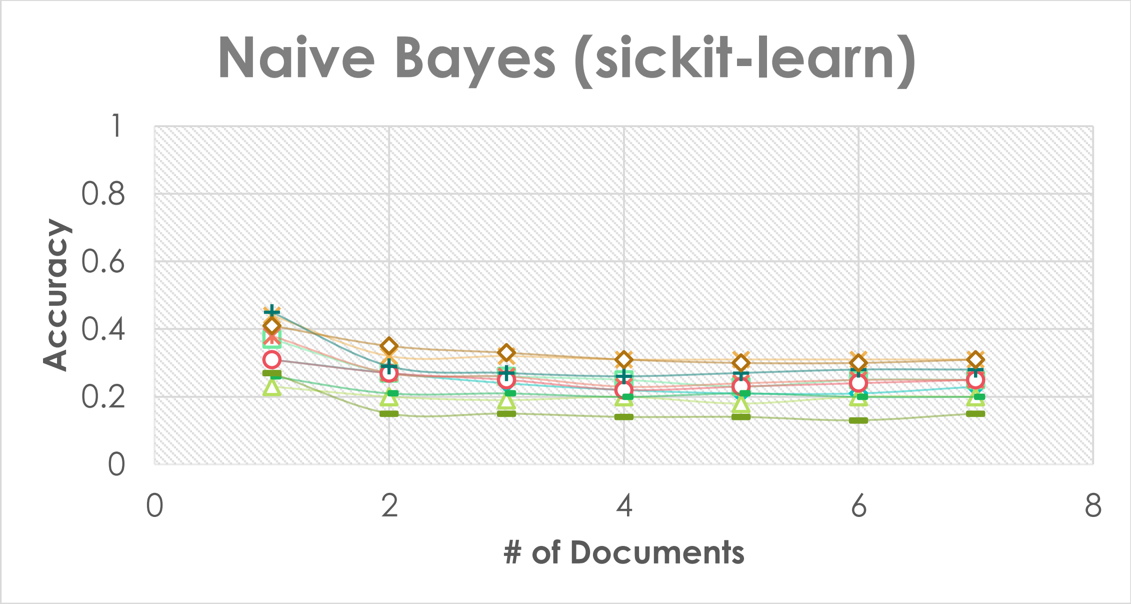
Class Prior Probability

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Predicted prior probability

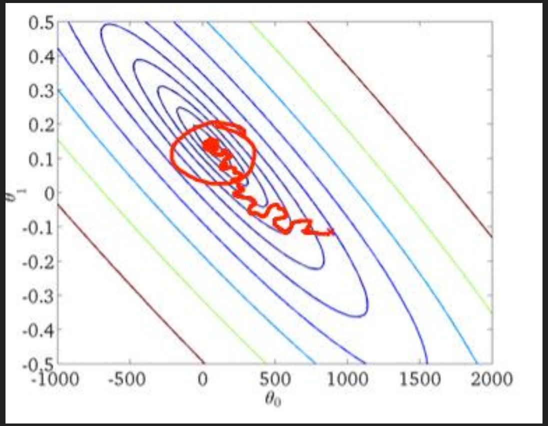
- Fast training
- Fast classification
- Terrible results

Naïve Bayes	Documents						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	0.31	0.27	0.24	0.22	0.21	0.21	0.23
100-200	0.37	0.27	0.26	0.25	0.23	0.25	0.25
200-300	0.23	0.2	0.19	0.2	0.18	0.2	0.2
300-400	0.44	0.32	0.32	0.31	0.31	0.31	0.31
400-500	0.38	0.27	0.26	0.23	0.24	0.25	0.25
500-600	0.31	0.27	0.25	0.22	0.23	0.24	0.25
600-700	0.45	0.29	0.27	0.26	0.27	0.28	0.28
700-800	0.26	0.21	0.21	0.2	0.21	0.2	0.2
800-900	0.27	0.15	0.15	0.14	0.14	0.13	0.15
900-100	0.41	0.35	0.33	0.31	0.3	0.3	0.31
Training Time	0.015	0.046	0.071	0.125	0.266	0.33	0.224
Elaspsed time	0.171	0.171	0.188	0.205	0.161	0.63	0.268



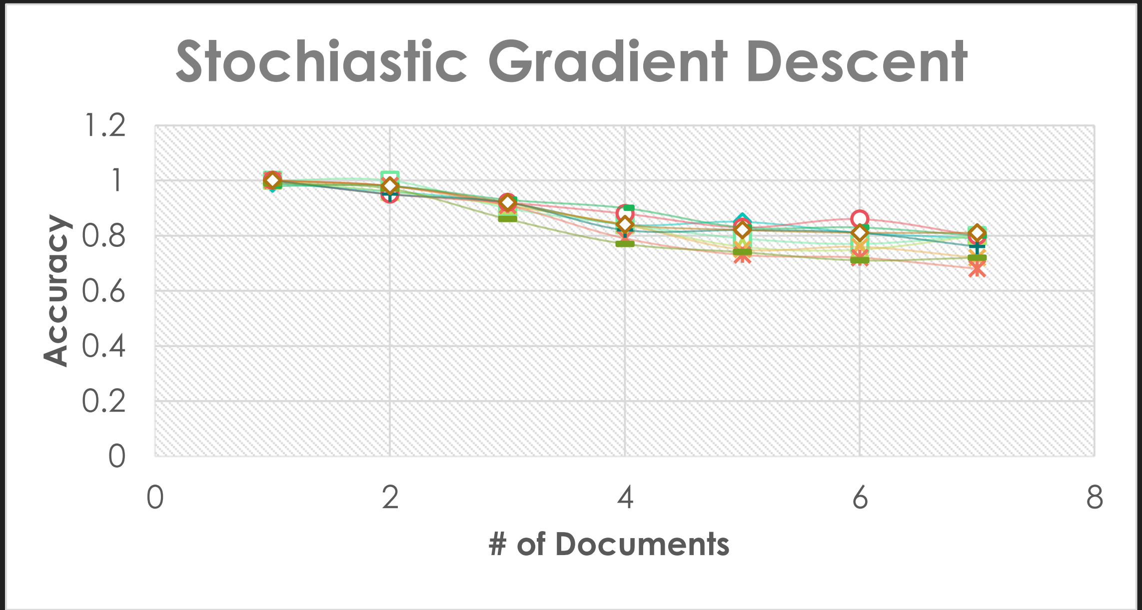
Stochastic Gradient Descent

- Trained by assigning weights and then updating iteratively until convergence at a maximum



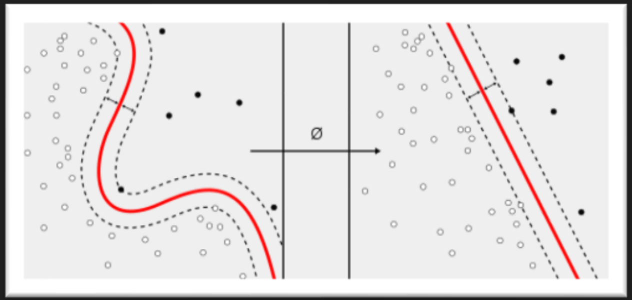
- Fast training
- Fast classification
- Sufficiently accurate

Stochiastic Gradient Descent	Accuracy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	0.99	0.96	0.92	0.84	0.85	0.81	0.79
100-200	1	1	0.9	0.84	0.79	0.77	0.8
200-300	1	0.96	0.91	0.84	0.76	0.75	0.8
300-400	1	0.98	0.91	0.84	0.75	0.76	0.72
400-500	1	0.98	0.91	0.79	0.73	0.72	0.68
500-600	1	0.95	0.92	0.88	0.83	0.86	0.8
600-700	1	0.95	0.92	0.82	0.82	0.81	0.76
700-800	0.98	0.98	0.93	0.9	0.83	0.83	0.8
800-900	0.99	0.97	0.86	0.77	0.74	0.71	0.72
900-100	1	0.98	0.92	0.84	0.82	0.81	0.81
Training Time	0.032	0.141	0.328	0.702	1.105	1.569	1.827
Elaspsed time	0.017	0.174	0.213	0.233	0.253	0.266	0.296



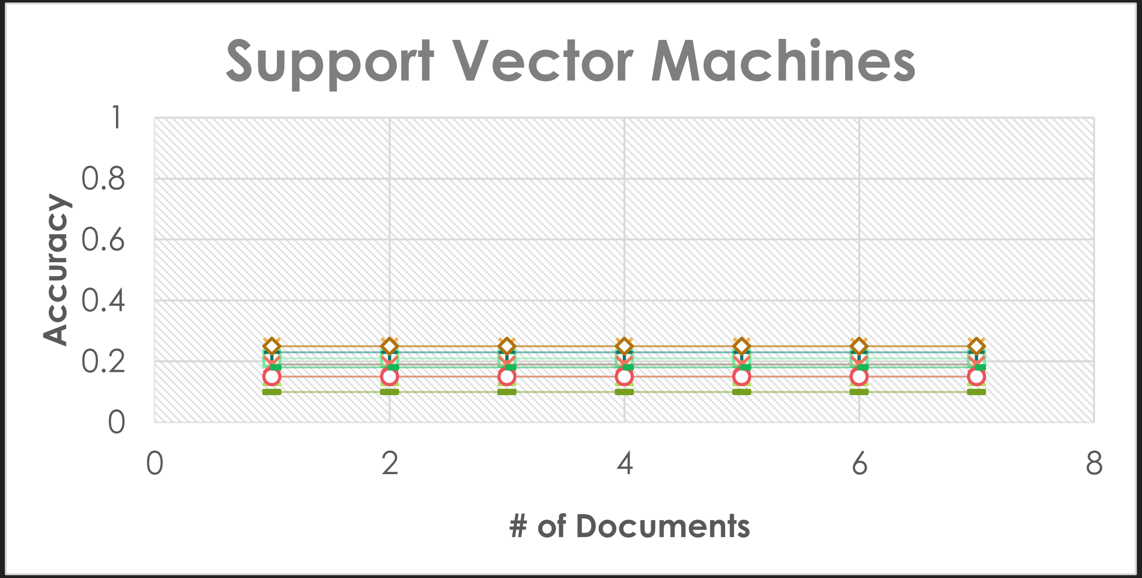
Support Vector Machines

- Points in space where categories are separated by gaps



- Slow training
- Slow classification
- Miserable results

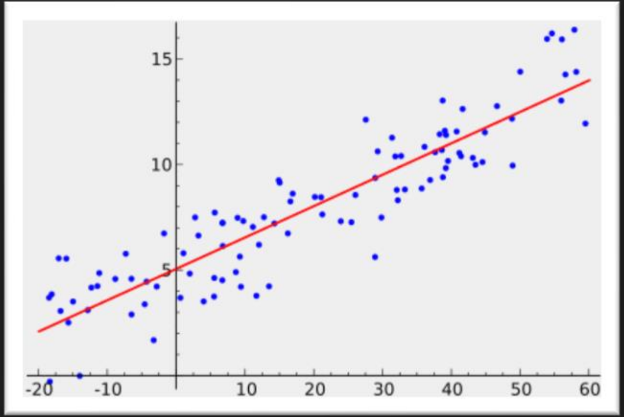
Vector Machines	Accuracy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	0.19	0.19	0.19	0.19	0.19	0.19	0.19
100-200	0.21	0.21	0.21	0.21	0.21	0.21	0.21
200-300	0.15	0.15	0.15	0.15	0.15	0.15	0.15
300-400	0.25	0.25	0.25	0.25	0.25	0.25	0.25
400-500	0.19	0.19	0.19	0.19	0.19	0.19	0.19
500-600	0.15	0.15	0.15	0.15	0.15	0.15	0.15
600-700	0.23	0.23	0.23	0.23	0.23	0.23	0.23
700-800	0.18	0.18	0.18	0.18	0.18	0.18	0.18
800-900	0.1	0.1	0.1	0.1	0.1	0.1	0.1
900-100	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Training Time	0.674	14.68	58.409	253.598	572.56	1041.821	1344.291
Elaspsed time	0.673	2.44	5.304	9.736	18	19.182	21.89



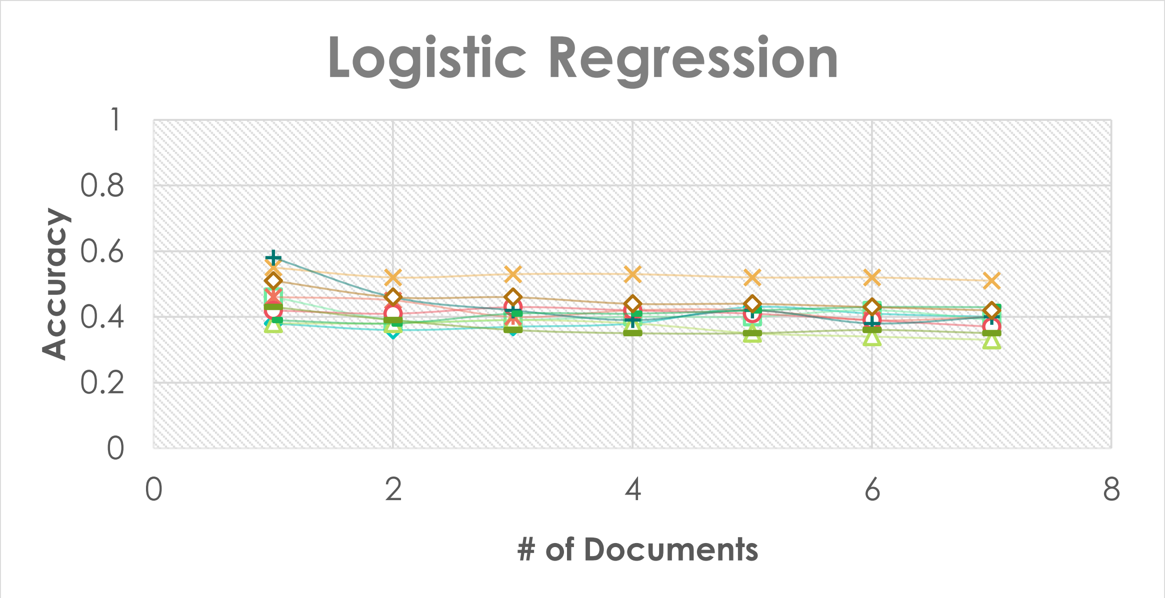
Logistic Regression

Logistic Regression	Accuracy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	0.38	0.36	0.37	0.38	0.43	0.41	0.4
100-200	0.46	0.39	0.39	0.4	0.4	0.42	0.39
200-300	0.38	0.38	0.39	0.38	0.35	0.34	0.33
300-400	0.55	0.52	0.53	0.53	0.52	0.52	0.51
400-500	0.46	0.45	0.4	0.42	0.42	0.39	0.4
500-600	0.42	0.41	0.43	0.42	0.41	0.39	0.37
600-700	0.58	0.46	0.42	0.39	0.42	0.38	0.4
700-800	0.39	0.38	0.41	0.41	0.42	0.43	0.43
800-900	0.43	0.39	0.36	0.35	0.35	0.36	0.35
900-100	0.51	0.46	0.46	0.44	0.44	0.43	0.42
Training Time	0.212	1.12	2.4	6.08	13.339	18.722	20.227
Elaspsed time	0.085	0.078	0.078	0.094	0.309	0.107	0.082

- Multinomial logistic regression also known as MaxEnt
- Features, scores, weights

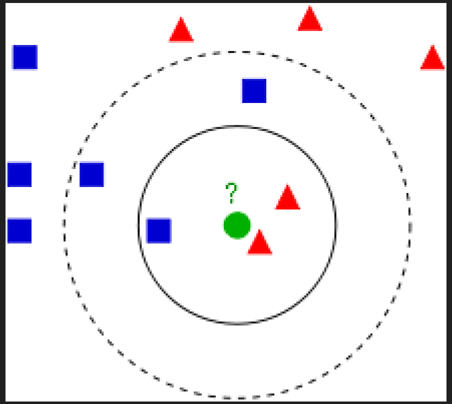


- Fast training
- Fast classification
- Subpar results



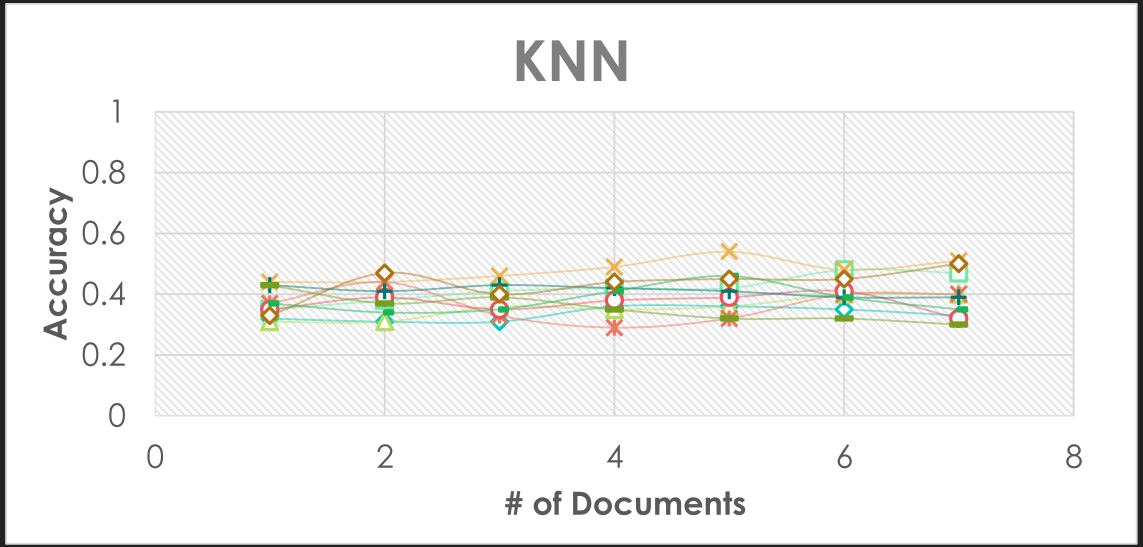
K Nearest Neighbors

- Classified by a majority vote of its neighbors n nearest neighbors



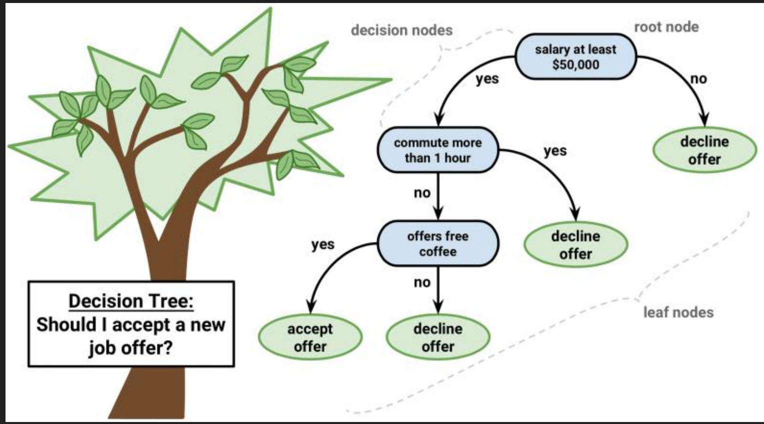
- Really fast training
- Relatively fast classification
- Pretty bad results

K Nearsest Neighbors	Accura cy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	0.32	0.31	0.31	0.36	0.36	0.35	0.33
100-200	0.35	0.38	0.41	0.42	0.42	0.48	0.47
200-300	0.31	0.31	0.36	0.35	0.36	0.4	0.4
300-400	0.44	0.44	0.46	0.49	0.54	0.48	0.51
400-500	0.37	0.44	0.33	0.29	0.32	0.4	0.4
500-600	0.35	0.39	0.35	0.38	0.39	0.41	0.32
600-700	0.43	0.41	0.43	0.42	0.41	0.39	0.39
700-800	0.37	0.34	0.35	0.41	0.46	0.39	0.35
800-900	0.43	0.37	0.39	0.35	0.32	0.32	0.3
900-100	0.33	0.47	0.4	0.44	0.45	0.45	0.5
Training Time	0	0	0	0.016	0.019	0.031	0.031
Elaspsed time	0.155	0.625	1.078	1.983	3.31	4.045	4.59



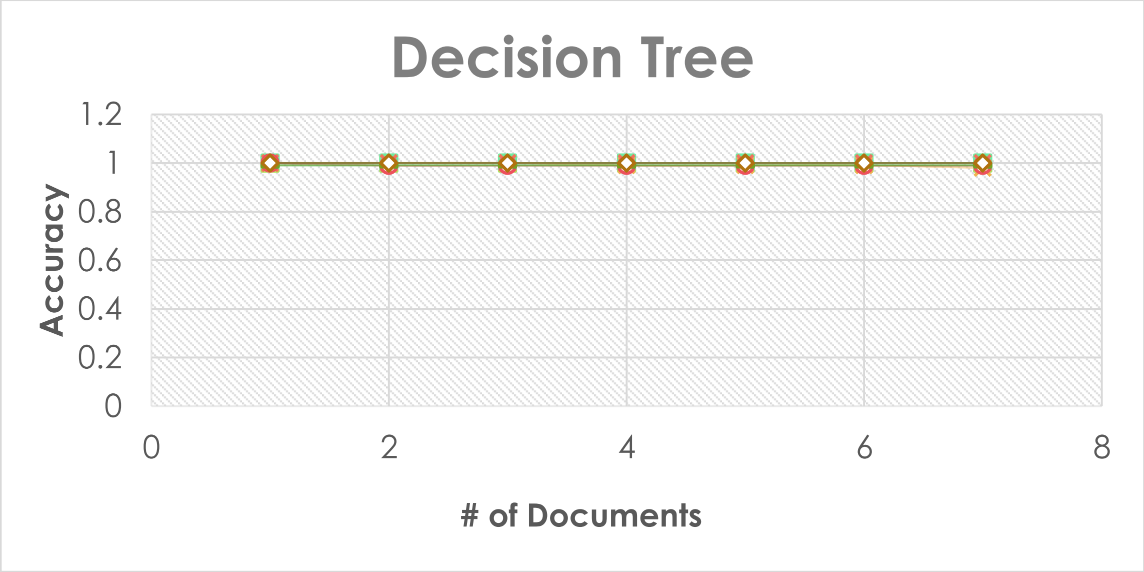
Decision Tree

- Data into subsets
- Decision nodes



- Slowish training
- Fast classification
- Really good results

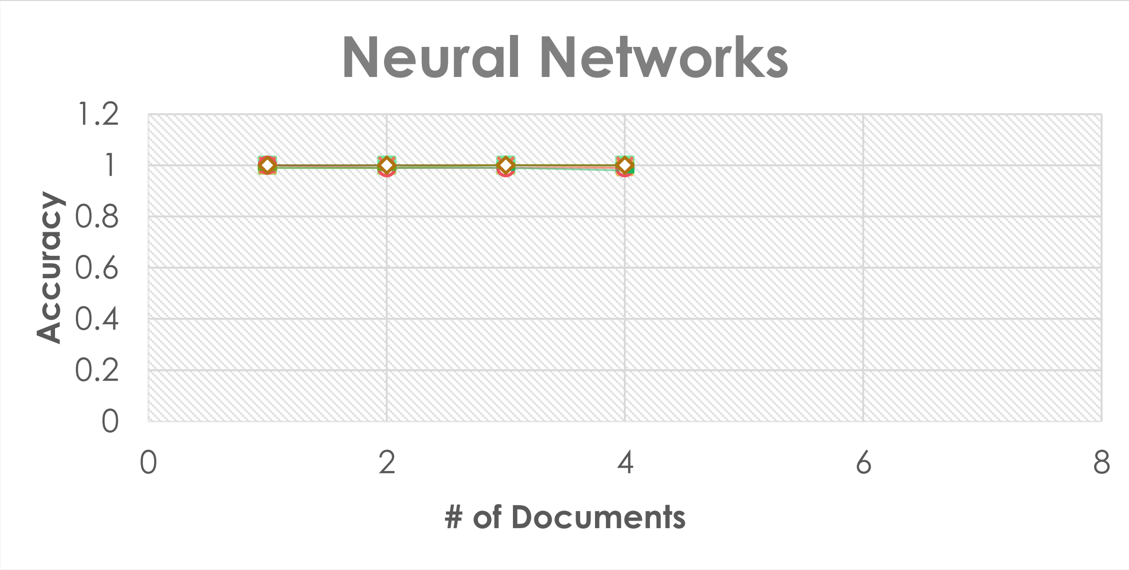
Decision Tree	Accuracy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	1	1	1	1	1	1	1
100-200	1	1	1	1	1	1	1
200-300	1	1	1	1	1	1	1
300-400	1	1	1	0.99	0.99	0.99	0.98
400-500	1	1	1	1	1	1	1
500-600	1	0.99	0.99	0.99	0.99	0.99	0.99
600-700	1	1	1	1	1	1	1
700-800	0.99	0.99	0.99	0.99	0.99	0.99	0.99
800-900	0.99	0.99	0.99	0.99	0.99	0.99	0.99
900-100	1	1	1	1	1	1	1
Training Time	0.806	7.974	23.993	56.48	99.828	159.71	198.07
Elaspsed time	0.075	0.092	0.154	0.078	0.102	0.094	0.0899



Neural Networks

- Process samples one by one
- Compare result to actual label
- Errors are from classification are used to make modifications
- Backwards prorogation, tuning

Neural Networks	Accuracy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	1	1	1	1			
100-200	1	1	1	1			
200-300	1	1	1	1			
300-400	1	1	1	0.99			
400-500	1	1	1	1			
500-600	1	0.99	0.99	0.99			
600-700	1	1	1	1			
700-800	0.99	0.99	0.99	0.98			
800-900	0.99	0.99	1	1			
900-100	1	1	1	1			
Training Time	56.315	435.316	1506.819	10349.03			
Elaspsed time	0.026	0.105	0.25	0.15			



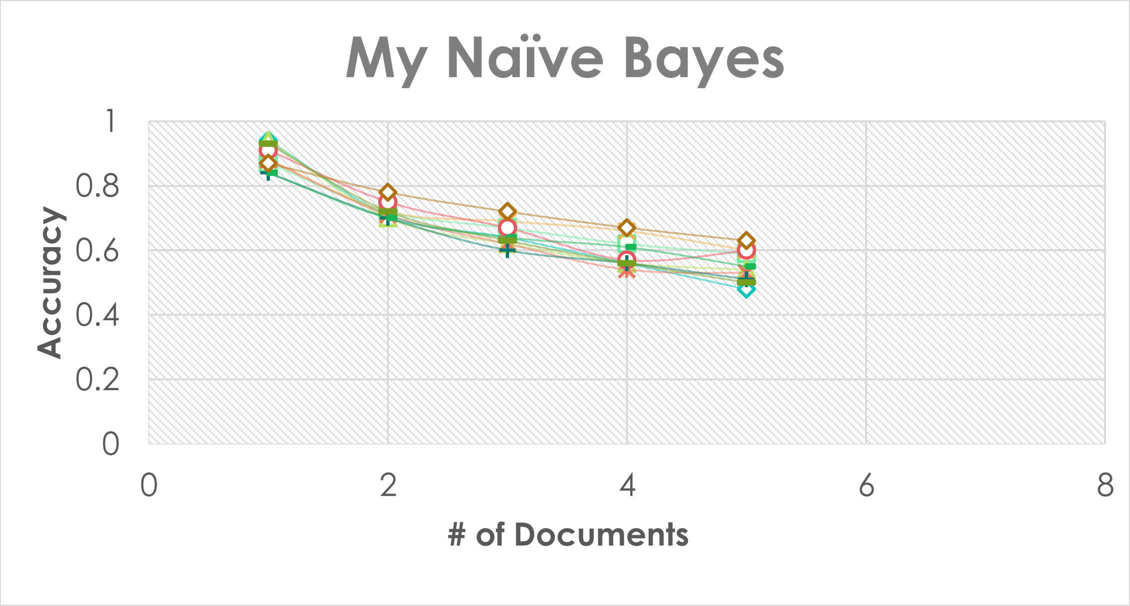
- Slowest training I ever did see
- Fast classification
- Great results

My Naïve Bayes

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

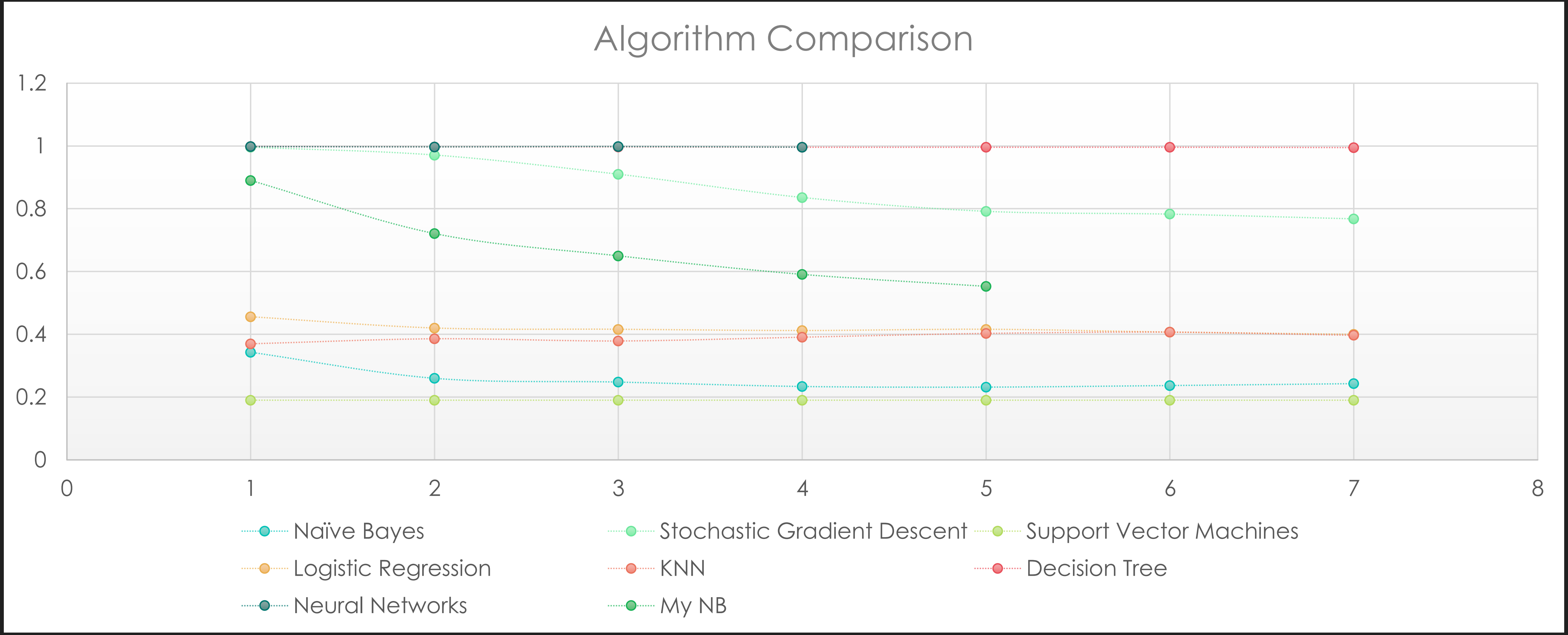
- Made by me
- Bag of words method

MyNB	Accuracy						
Training Set Size	1k	5k	10k	20k	30k	40k	50k
0-100	0.94	0.71	0.64	0.56	0.48		
100-200	0.87	0.72	0.67	0.62	0.59		
200-300	0.94	0.7	0.62	0.56	0.54		
300-400	0.88	0.72	0.69	0.66	0.6		
400-500	0.88	0.71	0.62	0.54	0.53		
500-600	0.91	0.75	0.67	0.57	0.6		
600-700	0.84	0.7	0.6	0.56	0.51		
700-800	0.84	0.7	0.64	0.61	0.55		
800-900	0.93	0.72	0.63	0.56	0.5		
900-100	0.87	0.78	0.72	0.67	0.63		
Training Time	0.16	0.944	2.713	14.99	24.331		
Elaspsed time	132.059	877.13	2418.075	3427.908	41635.77		



- Fast training
- SLOW classification
- Good then terrible results

Algorithm Comparison



My NB In Action

```
merry christmans ==> (u'17', 1.6138699208126408e-09)
hapy 4th of july ==> (u'11', 2.848003624086619e-18)
i hate you ==> (u'2', 1.1685893961947471e-14)
i love you ==> (u'0', 7.436597389420465e-13)
```

🎄
🇺🇸
😂
❤️

Emoji Mapping

0	❤	_red_heart_
1	😍	_smiling_face_with_hearteyes_
2	😂	_face_with_tears_of_joy_
3	💕	_two_hearts_
4	🔥	_fire_
5	😊	_smiling_face_with_smiling_eyes_
6	😎	_smiling_face_with_sunglasses_
7	✨	_sparkles_
8	💙	_blue_heart_
9	😘	_face_blowing_a_kiss_
10	📷	_camera_
11	US	_United_States_
12	☀	_sun_
13	💜	_purple_heart_
14	😉	_winking_face_
15	💯	_hundred_points_
16	😁	_beaming_face_with_smiling_eyes_
17	🎄	_Christmas_tree_
18	📸	_camera_with_flash_
19	😜	_winking_face_with_tongue_

Extraction of Mathematical Expressions from Natural Language Statement

Avinash Saxena and Bhavesh Munot

Prof. Ruihong Huang

CSCE 489 Natural Language Processing

Motivation

- A professor from Mechanical Department of TAMU needed this for his research in autonomous driving vehicles
- Mathematical form is the most unambiguous form of expressing relation between entities

Approach

- Autocorrect in the statement
- Case Insensitive matching
- Stopwords removal
- Unworthy words removal
- Operators extraction
- Operand extraction
- Appropriate structuring of the entities & operators
- Complex expressions handling

Autocorrect

- 1st approach : Edit-distance
- 2nd approach : Python library “autocorrect” API

Sentence Pre-processing

- Case does not matter
- Articles are not useful
- Helping verbs are not really helpful

Entities Extraction

- Predefined set of operators for now:
    - +, -, *, /, <, <=, >, >=, =
- Remaining parts of the sentence are operands with some exceptions
- Place operators and operands in proper relative order
- Based on the sentence, change the operators.

  e.g. A is less than B by 10 ---> A = B - 10

Variables and Operands matching

- 1st approach : substring matching
  - Issue - Not intelligent, very hard wired
- 2nd approach : little better substring matching
  - Matching irrespective of word position

Variables and Operands matching

- 3rd approach : string matching with fixed window size

- 4th approach : Cosine similarity with fixed window size of 4
  - Issue : False positives, operands order mismatch

Variables and Operands matching

- 5th approach : Cosine similarity with trigram model

- 6th approach : Cosine similarity without window size
  - Best so far
  - Extract the other parts and then pass the remaining parts of sentences separately
  - $$similarity(x, y) = cos(\theta) = \frac{x \cdot y}{||x|| * ||y||}$$

Other approaches tried during the journey

- At first we didn’t extract the operators, operands separately and kept the fixed window size.
- It was not working for misspelled words, synonymous words
  - Applied DFS to generate all the possible combinations of sentences with synonymous words
- Tried multiple similarity detection approaches
- Tagged nouns in the sentence so that we don’t miss those

Progress so far...

- All simple straightforward sentences work without any issues

- between, range is also supported.
  e.g. a plus b is between 10 and 20 ⟶ a + b in [10, 20]

- a is less than b ⟶ a < b
  a is less than b by 20 ⟶ a = b - 20 // No ‘<’ operator in expression

- Units of quantities remain as it is.
  E.g. Vehicle distance should be less than 10 m // vehicle_distance < 10 m

Future Scope

- Parenthesization
- Logical operators & operations support
- Trigonometric functions
- Raised to, power, square root etc.

Examples

1. Vehicle Y is ahead of vehicle X by 20 meter
2. The speed of the vehicle should be less than 50 kph
3. Vehicle distance is less than vehicle acceleration
4. Vehicle distance is less than vehicle acceleration by 20
5. The product of 5 and 10 is equal to 50

Future Work

1. Generating dictionary of operands and operators from the e-books in Mechanical Engineering.

2. Generating most frequent synonyms relevant to mechanical engineering using word frequency computation across huge set of text from 1000s of books in mechanical engineering.

Thank you!

CSCE 489 Natural Language Processing

Text Summarization

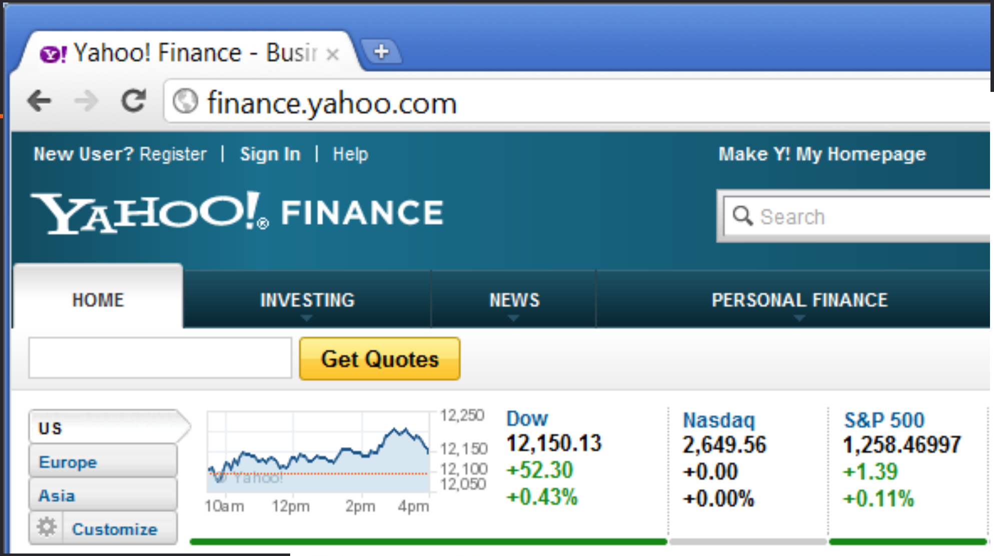
Justin Jin

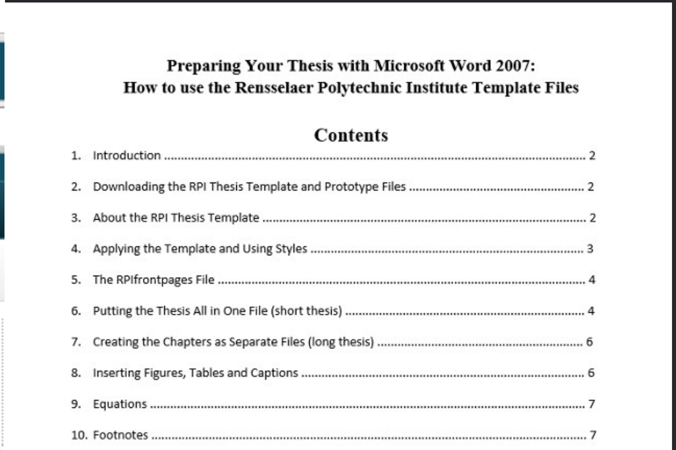
1.
Automatic Summarization

Information about automatic summarization

What is a "Summary"?

- Project Abstract
- News
- Finance
- Table of Contents





Preparing Your Thesis with Microsoft Word 2007:
How to use the Rensselaer Polytechnic Institute Template Files

Contents

What is Automatic Summarization?

- Process of shortening a text document or paragraph using software in order to create a coherent summary
- Part of Machine Learning and Data Mining
- Subset of data which contains information of entire set

Widely used in popular industry such as search engines, image collecting, and videos searching!

Two Approaches

Extraction

- Use existing data to build phrases or sentences from the original text
- Do not modify the original objects

Abstraction

- Build internal semantic representation then use natural language techniques to create a summary
- More of a human representation

Types of Summarization

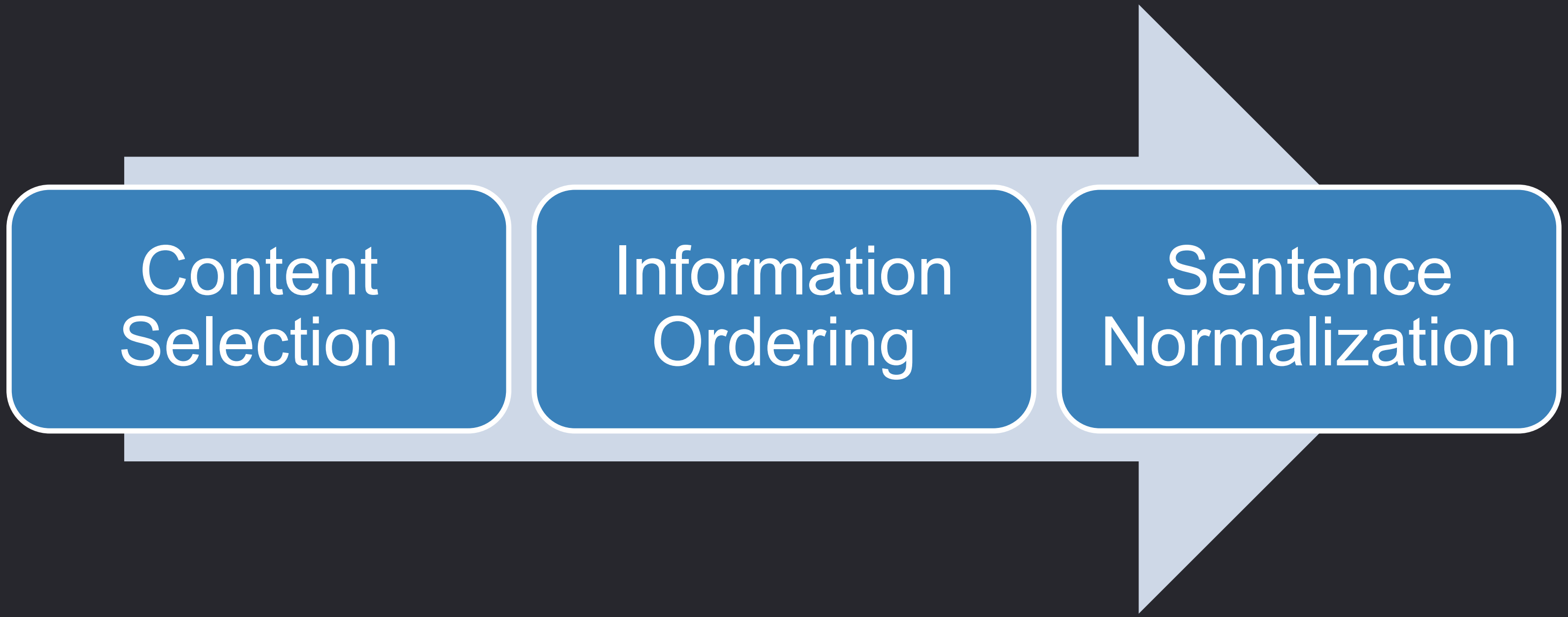
Query-Focused Summarization

- Summarize a document using a specific user query
- Answers questions by summarizing a document using outside information to construct answers

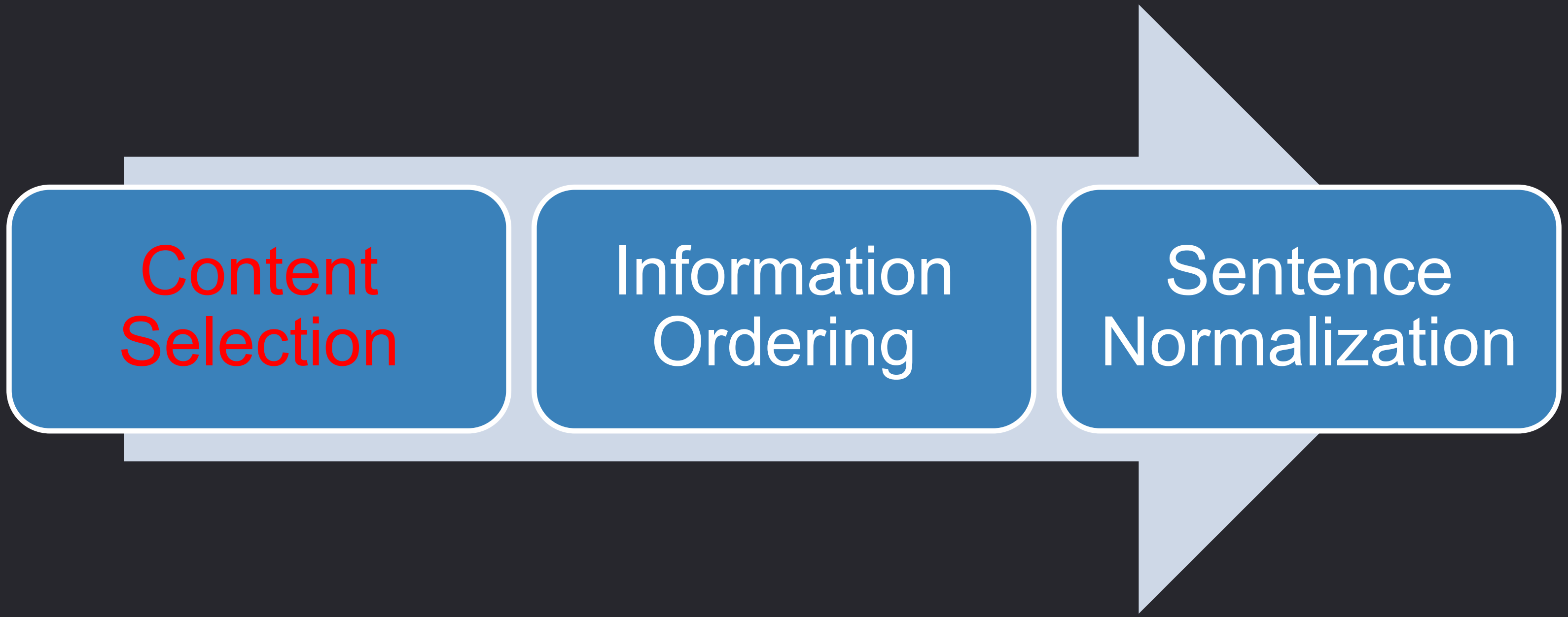
Generic Summarization

- Summarize the base content of a document

Stages of Summarization



Stages of Summarization



Content Selection

- How do I choose which sentences to extract from the document?
- Supervised Content Selection
- Unsupervised Content Selection

Supervised Content Selection

- Given a training set of data (good summaries) in each document
- Correlate each sentence in the document with sentences in the summary
- Certain Word Features to look for: Position, Length of Sentence, Cohesion
- Binary classifier – Should sentence be included in the summary?

Supervised Summaries

- ROUGE – Recall Oriented Understudy for Gisting Evaluation
- Internal metric for evaluating summaries
- Given a Document D and a automatically generated summary S
- Have reference summaries made beforehand from humans and compare these two models
- Calculate percentage of bigrams from reference summaries that also appear in automatic summary S

$$ROUGE-2 = \frac{\sum_{s \in \{\text{RefSummaries}\}} \sum_{\text{bigrams } i \in S} \min(count(i,X), count(i,S))}{\sum_{s \in \{\text{RefSummaries}\}} \sum_{\text{bigrams } i \in S} count(i,S)}$$

Example of ROUGE

Example

- Human 1: water spinach is a green leafy vegetable grown in the tropics.
- Human 2: water spinach is a semi-aquatic tropical plant grown as a vegetable.
- Human 3: water spinach is a commonly eaten leaf vegetable of Asia.

- System: water spinach is a leaf vegetable commonly eaten in tropical areas of Asia.

- ROUGE -2= $\frac{3+3+6}{10+9+9} = 12/28 = 0.43$

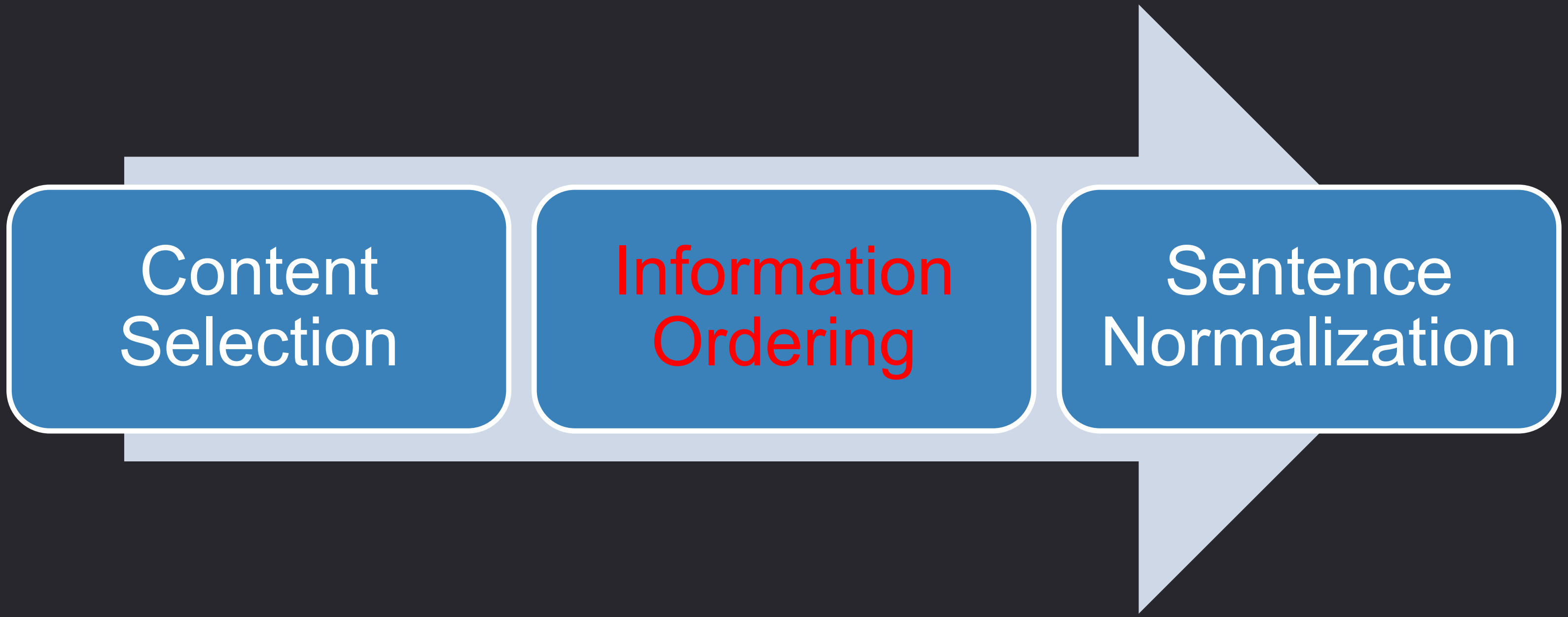
Unsupervised Content Selection

- Choose sentences that are distinguished or are informative based on the weight
- TF-IDF measure is used to determine which words are informative

$$w_{t,d} = \mathrm{tf}_t \cdot \log \frac{|D|}{|\{t \in d\}|}$$

- Log-Likelihood Ratio – Statistical Test used to compare the goodness of fit between two models

Stages of Summarization



Information Ordering

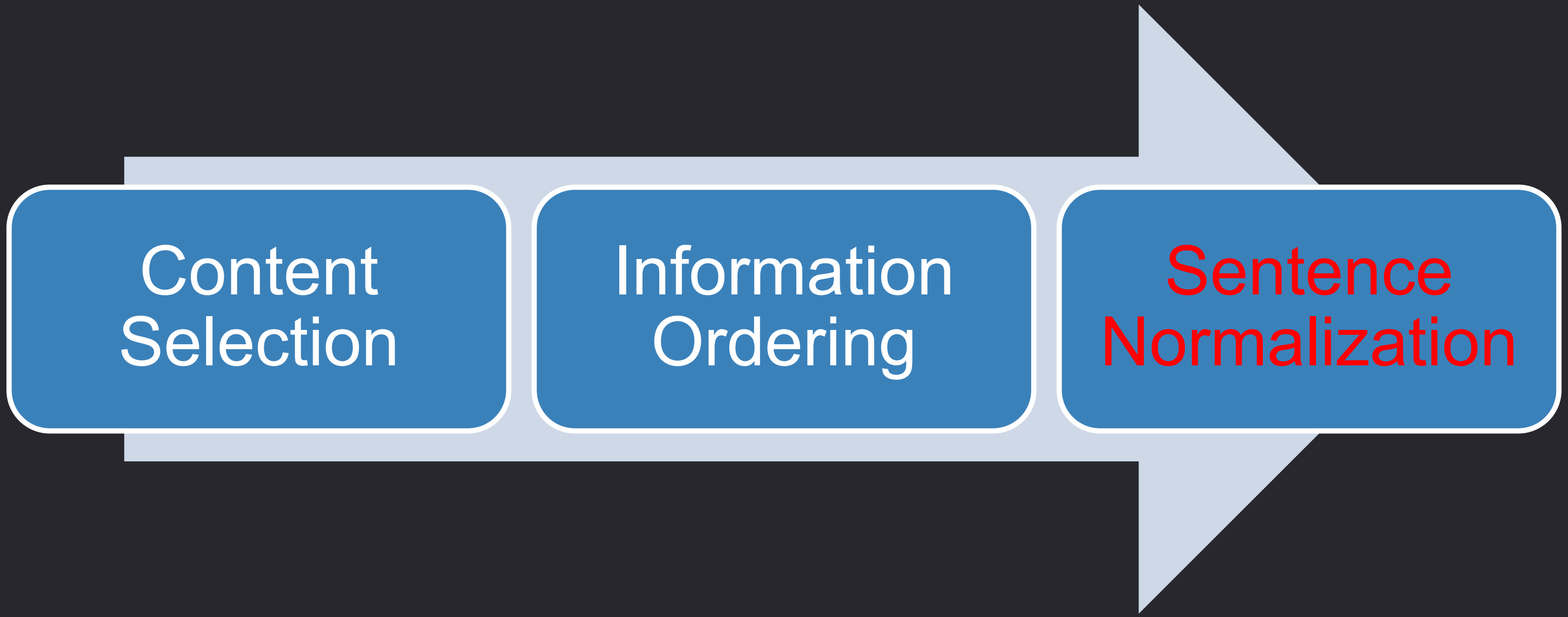
- How do I order these sentences in the way that makes the most sense?

- Coherence
- Chronological Ordering
- Topics

Information Ordering

Coherence	• Choose sentence ordering based on mathematical calculations (cosine similarity) • Ordering based on which sentences are discussing the same entity.
Chronological Ordering	• Choose sentence ordering based on document date or time differences within the document
Topics	• Discover topic ordering from source document

Stages of Summarization



Sentence Normalization

- How do I clean up this sentence to make it presentable and easy to read?

- Parse the sentence and extract parts of speech tags
- Changing characters from upper to lower case and vice versa
- Removing stopwords
- Expanding abbreviations
- Stemming and Lemmatization

2.
Project Rundown

Information about what I did (or attempted to do)

Overview

- Implemented Extractive Text Summarization

- Count occurrences of each word in a paragraph
- Calculate frequency of the word
- Add each frequency of the word in each sentence
- The sentence with the highest frequency is used as the "summary" of the paragraph

Some Regex Used

```python
def word_parse(self, sentence):

    word_list = []
    stop_words = ["the","a","were","so","in","on", "an","to", "it", "of", "is","not","are", "all","as","by"]

    # This particular regex is used to finds words in sentences
    word_regex = r"([a-zA-Z]+)"
```

```python
def sentence_token(self, paragraph):

    # This particular regex is used to find sentences in paragraphs
    sentence_regex = r"([\"]*[a-zA-z][a-z0-9A-Z, \-\'\"!<>\]\[]+[.][\"]*)"
```

```
C:\Users\Justin\PycharmProjects\Final Project NLP>python TextSummarizer.py
The FrequencySummarizer tokenizes the input into sentences then computes the term frequency map of the words. Then, the frequency map is filtered in
 order to ignore very low frequency and highly frequent words, this way it is able to discard the noisy words such as determiners, that are very fre
quent but don't contain much information, or words that occur only few times. And finally, the sentences are ranked according to the frequency of th
e words they contain and the top sentences are selected for the final summary.

******Sentences: This list is the sentences of the paragraph, each matched with a specific index used in the computation of the frequency***********
*
{0: 'The FrequencySummarizer tokenizes the input into sentences then computes the term frequency map of the words.',
 1: "Then, the frequency map is filtered in order to ignore very low frequency and highly frequent words, this way it is able to discard the noisy w
ords such as determiners, that are very frequent but don't contain much information, or words that occur only few times.",
 2: 'And finally, the sentences are ranked according to the frequency of the words they contain and the top sentences are selected for the final sum
mary.'}

******Word List: Each seperate token of each Word******
['frequencysummarizer', 'tokenizes', 'input', 'into', 'sentences', 'then', 'computes', 'term', 'frequency', 'map', 'words', 'then', 'frequency', 'ma
p', 'filtered', 'order', 'ignore', 'very', 'low', 'frequency', 'and', 'highly', 'frequent', 'words', 'this', 'way', 'able', 'discard', 'noisy', 'wor
ds', 'such', 'determiners', 'that', 'very', 'frequent', 'but', 'don', 't', 'contain', 'much', 'information', 'or', 'words', 'that', 'occur', 'only',
 'few', 'times', 'and', 'finally', 'sentences', 'ranked', 'according', 'frequency', 'words', 'they', 'contain', 'and', 'top', 'sentences', 'selected
', 'for', 'final', 'summary']

******Word List with Frequency: Token with the frequency of it occuring in paragraph******
{'and': 0.6, 'they': 0.2, 'tokenizes': 0.2, 'into': 0.2, 'highly': 0.2, 'computes': 0.2, 'frequency': 0.8, 'sentences': 0.6, 'times': 0.2, 'occur':
0.2, 'information': 0.2, 'for': 0.2, 'top': 0.2, 'selected': 0.2, 'able': 0.2, 'finally': 0.2, 'few': 0.2, 'only': 0.2, 'much': 0.2, 'low': 0.2, 'wa
y': 0.2, 'filtered': 0.2, 'final': 0.2, 't': 0.2, 'then': 0.4, 'that': 0.4, 'very': 0.4, 'frequencysummarizer': 0.2, 'ranked': 0.2, 'but': 0.2, 'wor
ds': 1.0, 'input': 0.2, 'such': 0.2, 'noisy': 0.2, 'term': 0.2, 'don': 0.2, 'this': 0.2, 'summary': 0.2, 'according': 0.2, 'or': 0.2, 'ignore': 0.2,
 'map': 0.4, 'determiners': 0.2, 'contain': 0.4, 'discard': 0.2, 'order': 0.2, 'frequent': 0.4}
[(7.599999999999999, 0), (13.999999999999995, 2), (17.399999999999995, 1)]


Then, the frequency map is filtered in order to ignore very low frequency and highly frequent words, this way it is able to discard the noisy words
such as determiners, that are very frequent but don't contain much information, or words that occur only few times.
```

```
C:\Users\Justin\PycharmProjects\Final Project NLP>python TextSummarizer.py
Virgin Group founder Richard Branson rode out Hurricane Irma in his wine cellar on his private island in the British Virgin Islands. Tuesday, as man
y of the same islands braced for the impact of Hurricane Maria, he appeared on CNN's "New Day" with a message: "Climate change is real." "Look, you
can never be 100% sure about links," Branson said after anchor John Berman asked if he saw a correlation between the recent hurricanes and climate c
hange. "But scientists have said the storms are going to get more and more and more intense and more and more often. We've had four storms within a
month, all far greater than that have ever, ever, ever happened in history." "Sadly," he continued, "I think this is the start of things to come."

******Sentences: This list is the sentences of the paragraph, each matched with a specific index used in the computation of the frequency***********
*
{0: 'Virgin Group founder Richard Branson rode out Hurricane Irma in his wine cellar on his private island in the British Virgin Islands.',
 1: '"Climate change is real."',
 2: 'sure about links," Branson said after anchor John Berman asked if he saw a correlation between the recent hurricanes and climate change.',
 3: '"But scientists have said the storms are going to get more and more and more intense and more and more often.',
 4: 'We\'ve had four storms within a month, all far greater than that have ever, ever, ever happened in history."',
 5: '"Sadly," he continued, "I think this is the start of things to come."'}

******Word List: Each seperate token of each Word******
['virgin', 'group', 'founder', 'richard', 'branson', 'rode', 'out', 'hurricane', 'irma', 'his', 'wine', 'cellar', 'his', 'private', 'island', 'briti
sh', 'virgin', 'islands', 'climate', 'change', 'real', 'sure', 'about', 'links', 'branson', 'said', 'after', 'anchor', 'john', 'berman', 'asked', 'i
f', 'he', 'saw', 'correlation', 'between', 'recent', 'hurricanes', 'and', 'climate', 'change', 'but', 'scientists', 'have', 'said', 'storms', 'going
', 'get', 'more', 'and', 'more', 'and', 'more', 'intense', 'and', 'more', 'and', 'more', 'often', 'we', 've', 'had', 'four', 'storms', 'within', 'mo
nth', 'far', 'greater', 'than', 'that', 'have', 'ever', 'ever', 'ever', 'happened', 'history', 'sadly', 'he', 'continued', 'i', 'think', 'this', 'st
art', 'things', 'come']

******Word List with Frequency: Token with the frequency of it occuring in paragraph******
{'and': 1.0, 'things': 0.2, 'think': 0.2, 'hurricanes': 0.2, 'founder': 0.2, 'correlation': 0.2, 've': 0.2, 'storms': 0.4, 'within': 0.2, 'rode': 0.
2, 'private': 0.2, 'month': 0.2, 'four': 0.2, 'sure': 0.2, 'have': 0.4, 'branson': 0.4, 'cellar': 0.2, 'out': 0.2, 'said': 0.4, 'group': 0.2, 'links
': 0.2, 'richard': 0.2, 'start': 0.2, 'saw': 0.2, 'get': 0.2, 'hurricane': 0.2, 'than': 0.2, 'virgin': 0.4, 'that': 0.2, 'going': 0.2, 'come': 0.2,
'scientists': 0.2, 'between': 0.2, 'john': 0.2, 'ever': 0.6, 'more': 1.0, 'real': 0.2, 'we': 0.2, 'his': 0.4, 'greater': 0.2, 'irma': 0.2, 'far': 0.
2, 'after': 0.2, 'about': 0.2, 'but': 0.2, 'often': 0.2, 'if': 0.2, 'recent': 0.2, 'sadly': 0.2, 'continued': 0.2, 'intense': 0.2, 'change': 0.4, 'h
e': 0.4, 'climate': 0.4, 'i': 0.2, 'island': 0.2, 'british': 0.2, 'asked': 0.2, 'this': 0.2, 'history': 0.2, 'berman': 0.2, 'islands': 0.2, 'had': 0
.2, 'happened': 0.2, 'anchor': 0.2, 'wine': 0.2}
[(0.8, 1), (1.5999999999999999, 5), (2.1999999999999997, 0), (5.0, 4), (9.600000000000001, 2), (13.2, 3)]


"But scientists have said the storms are going to get more and more and more intense and more and more often.
sure about links," Branson said after anchor John Berman asked if he saw a correlation between the recent hurricanes and climate change.
```

Review

- So how does this relate to automatic summarization?

- Content Selection
- Information Ordering
- Sentence Normalization

THANKS!

ANY QUESTIONS?

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival]
- Photographs by [Unsplash]

- https://glowingpython.blogspot.com/2014/09/text-summarization-with-nltk.html

- https://en.wikipedia.org/wiki/Likelihood-ratio_test

- https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

- Stanford NLP Course

- https://en.wikipedia.org/wiki/Automatic_summarization

Text Generation using Recurrent Neural Networks

LIAM MORAN

Overview

Learn: Construct a language model from a set of text documents

- Use recurrent neural networks

Generate: Input state is sequence of words (<eos> characters) t documents

- Allow model to predict next word

The Data

Penn Tree Bank (PTB) Dataset

- One million words of 1989 Wall Street Journal in Treebank II style

```
 a <unk> <unk> said this is an old story  we 're
talking about years ago before anyone heard of
asbestos having any questionable properties
there is no asbestos in our products now  neither
<unk> nor the researchers who studied the workers
were aware of any research on smokers of the kent
cigarettes  we have no useful information on
whether users are at risk said james a. <unk> of
boston 's <unk> cancer institute  dr. <unk> led a
team of researchers from the national cancer
institute and the medical schools of harvard
university and boston university  the <unk>
spokeswoman said asbestos was used in very modest
amounts in making paper for the filters in the
early 1950s and replaced with a different type of
<unk> in N  from N to N N billion kent cigarettes
with the filters were sold the company said
```

What are RNNs?

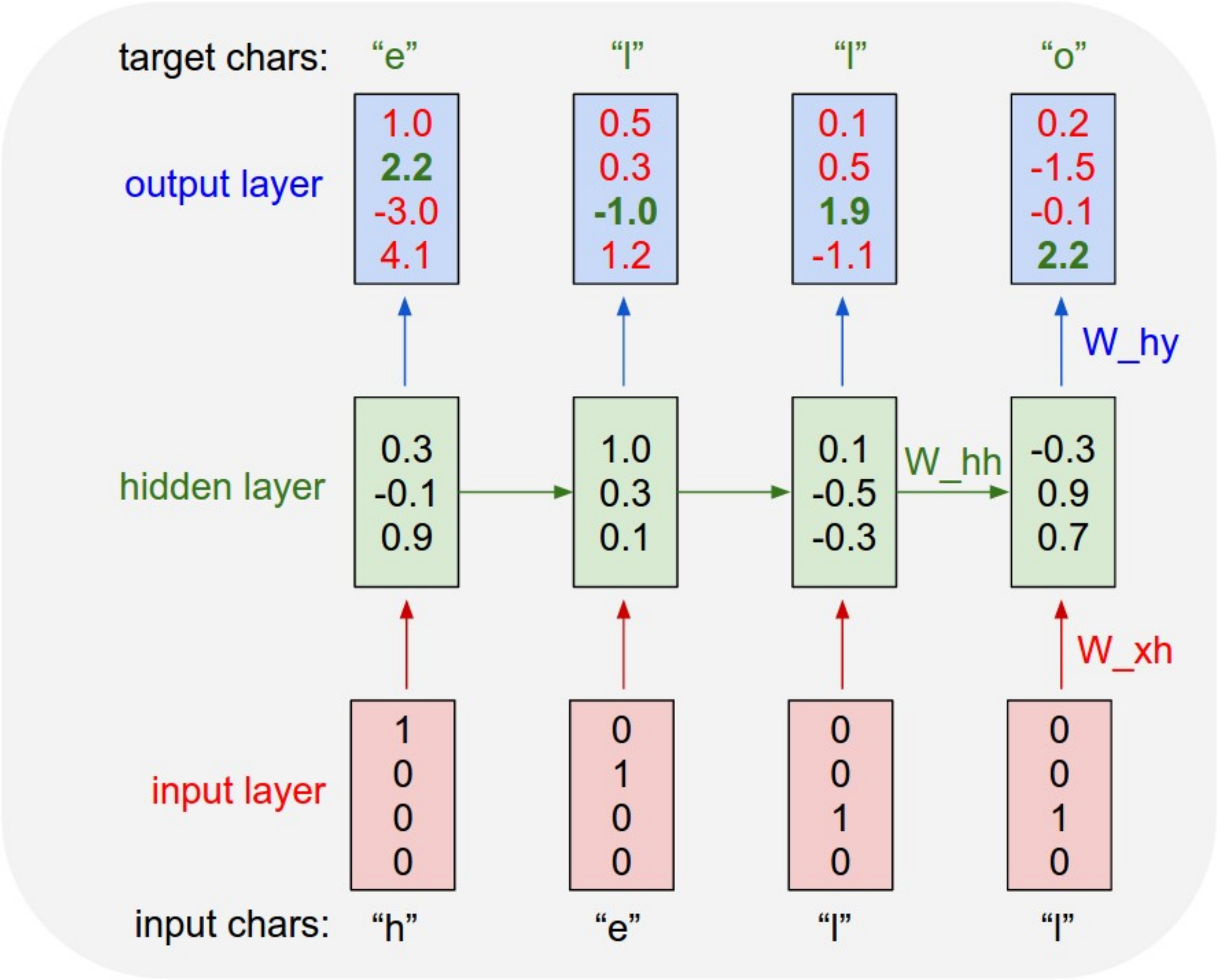
Sequences of tensors

Generates sentences from sentences while training on the word-level

Continuously apply input tensors to the state tensors

Good for data in the form of sequences (in this case, sequences of sentences)

They maintain this history of all their inputs, over the sequnces


target chars: "e" "l" "l" "o"
output layer
1.0
2.2
-3.0
4.1
0.5
0.3
-1.0
1.2
0.1
0.5
1.9
-1.1
0.2
-1.5
-0.1
2.2
W_hy
hidden layer
0.3
-0.1
0.9
1.0
0.3
0.1
0.1
-0.5
-0.3
W_hh
-0.3
0.9
0.7
W_xh
input layer
1
0
0
0
0
1
0
0
0
0
1
0
0
0
1
0
input chars: "h" "e" "l" "l"

RNN Example

Why RRNs?

Good for sequential context-based problems

Specifically LSTMs (long short-term memory) for extra context, but not too much context

LSTM cells hold words and maintain state across sequences

- Short-term to prevent overfitting
- Long to learn over sequences

Context is important

"I grew up in France… I speak fluent *French*."

Test Results

retirement the straight health continued high irs at pittsburgh treasury commerce contra federal sense coca-cola government in single dreams former columbia home of corp leave of at 's regulations of benefit be because N it board business to resolution newspaper make dropped run before u.s. is as year reasons is thinks a add relations companies.

Results

audio wo when gives can qintex is construction chapter measurements less $ the face its headquarters west that acceptable others connection following earlier fled the.

Results

anticipated N community different although administration said a your think has america completed it order or home addition the monday at growth has concern plan tvs offer not and and met share all a inc virginia plans a into in as for bank.

Results

wage more being market the as a wide to kellogg they of from shut N more navigation growth shares been election very acquire investment makes home continue he interview N being guber-peters term strategy launched fewer people reported and nikkei seem dorrance nearby founded ventures available disabled the cnbc reluctant that foster him in contends third-quarter withdrawn coupled or as adapted sharp improve largest street say in any vice for of about that ' be had battled n't has seems with bank chemical move.

Results

market and u.s. a expect where case anticipate order about rates or handful N unemployment the write suffer due the N regulators a the schaeffer said is early goes tennis posted year machines certain has at.

Results

into hall street american focused of days will they.

Results

transaction were is of a beyond the to for pertussis wedtech with credit a technology american.

Why the bad performance?

Model did not have enough time to train (running it with 13 recurrent steps takes roughly 4 hours on my laptop)

Input was just whatever was left over from the modeling phase (not end of sentence character)

Future Improvements

Clean the data a bit more (remove unnecessary words)

Train the model for longer

Optimize code to run better (currently no use of GPU or parallelization)

Tools Used

- Python
- TensorFlow

Sources

- https://www.tensorflow.org/tutorials/recurrent#lstm
- https://karpathy.github.io/2015/05/21/rnn-effectiveness/

The Professional Rhyming Assistant (PRA)

By: Larry L. Harris

How it works

Make synonyms.

Make rhymes.

Pick best rhyme.

Rearrange sentence.

Compute probability of sentence actually occuring.

Why rhyme anything at all?

A common way of writing poety, music, or many other forms of creative writing use rhyming as a way to catch a reader's attention.
I developed the PRA system with the hopes of inputting any document and getting an output of rhyming sentences that could potentially give a response the user would enjoy more than the original, or just suggest something the writer never thought of

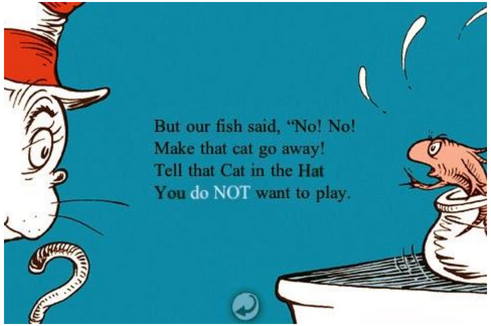
Common Rhyming Schemes:

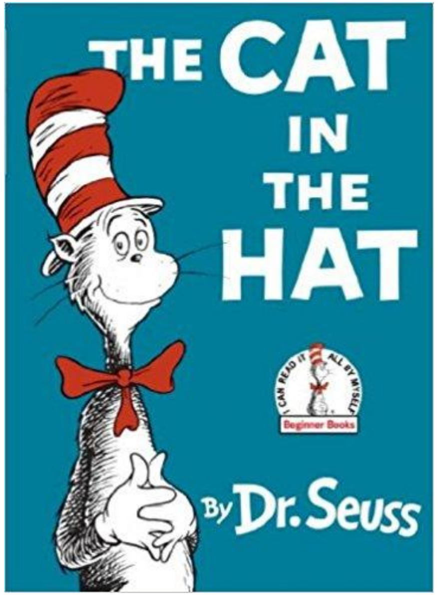
A, A, B, B

A, B, A, B

A,B,A,B,B

A,A,B,C,C





Creating synonyms

PRINCETON UNIVERSITY

WordNet
A lexical database for English

WordNet:

"WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept."

```
for syn in wn.synsets("darkness"):
    for l in syn.lemmas():
        synonyms.append(l.name())
        if l.antonyms():
            antonyms.append(l.antonyms()[0].name())
print(set(synonyms))
#print(set(antonyms))
```

```
Type "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:\Users\Larry Luke Harris\Desktop\489\project\testing2.py ====
{'helium', 'atomic_number_2', 'He', 'he'}
>>> 
==== RESTART: C:\Users\Larry Luke Harris\Desktop\489\project\testing2.py ====
{'wickedness', 'dark', 'darkness', 'swarthiness', 'shadow', 'duskiness', 'iniqui
ty'}
>>> 
```

Making a set of rhymes

The CMU Pronouncing Dictionary

PRA uses a python package called pronouncing to determine if two words have similar phonetic information.

```
>>> import pronouncing
>>> pronouncing.rhymes("climbing")
['diming', 'liming', 'priming', 'rhyming', 'timing']
```

```
>>> import pronouncing
>>> "cheese" in pronouncing.rhymes("wheeze")
True
>>> "fast" in pronouncing.rhymes("last")
True
>>> "dog" in pronouncing.rhymes("fog")
False
>>>
```

AW	cow	K AW
AY	hide	HH AY D
B	be	B IY
CH	cheese	CH IY Z
D	dee	D IY
DH	thee	DH IY
EH	Ed	EH D
ER	hurt	HH ER T
EY	ate	EY T
F	fee	F IY
G	green	G R IY N

Cross referencing databases

```
Sentence 1 tagged with part of speech

[('the', 'DT'), ('darkness', 'NN'), ('comes', 'VBZ'), ('with', 'IN'), ('every',
'DT'), ('night', 'NN')]
Sentence 2 tagged with part of speech

[('one', 'CD'), ('sleep', 'NN'), ('more', 'JJR'), ('we', 'PRP'), ('will', 'MD')
 ('lose', 'VB'), ('that', 'DT'), ('battle', 'NN')]


ALL synonyms for the first sentence in pair

{'hail', 'fall', 'Night', 'get', 'cum', 'wickedness', 'number', 'Nox', 'night',
'seed', 'every', 'ejaculate', 'get_along', 'issue_forth', 'occur', 'shadow', 'co
me_in', 'fare', 'seminal_fluid', 'arrive', 'derive', 'nighttime', 'follow', 'swa
rthiness', 'do', 'dark', 'come_up', 'total', 'amount', 'come', 'iniquity', 'dusk
iness', 'semen', 'make_out', 'add_up', 'descend', 'darkness'}

 ALL synonyms for the second sentence in pair

{'nap', 'ane', 'engagement', 'unitary', 'mislay', 'matchless', 'ace', 'sopor', '
l', 'leave', 'suffer', 'I', 'combat', 'i', 'quietus', 'one_and_only', 'unrivaled
', 'kip', 'more_than', 'conflict', 'eternal_rest', 'battle', 'testament', "log_Z
's", 'to_a_greater_extent', 'bequeath', 'peerless', 'nonpareil', 'will', 'single
', "catch_some_Z's", 'unrivalled', 'more', 'rest', 'unity', 'one', 'Sir_Thomas_M
ore', 'struggle', 'lose', 'turn_a_loss', 'unmatchable', 'unmatched', 'More', 'sl
umber', 'fall_back', 'recede', 'fight', 'drop_off', 'volition', 'sleep', 'mispla
ce', 'eternal_sleep', 'fall_behind', 'Thomas_More', 'miss'}
```

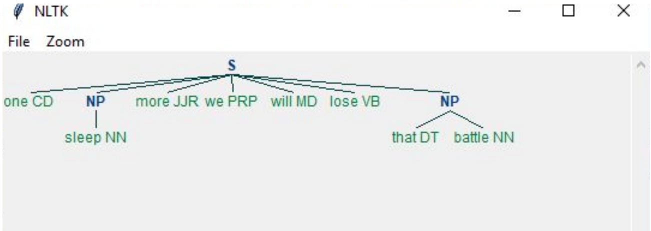
```
['night', 'number', 'seed']
Words that rhyme with words/synonyms in sentence 1

['fight', 'slumber', 'recede']
Position in sentence for word1, positon for its rhyming partner
```

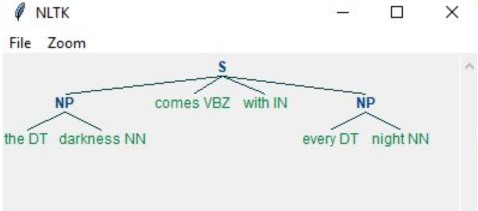
Building a suitable set of rhymes for any given sentence is a very difficult task even with these large databases that contain 130,000+ words.

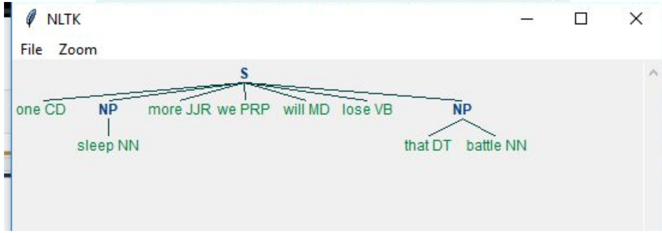
Rearranging sentences, tagging POS

PRA works by moving the rhyming words to the end of the sentence, while attempting to leave the sentence as intact as possible to preserve meaning.

```
['three', 'days', 'ago', 'we', 'walked', 'through', 'mountains', 'high']
['I', 'want', 'to', 'lay', 'down', 'and', 'expire']
go
low
blue
lie
i
die
[0.2857142857142857, 0.23529411764705882, 0.3076923076923077, 0.3333333333333333
, 1.0, 0.2857142857142857, 0.23529411764705882, 0.10526315789473684, 0.375, 0.
6666666666666666, 0.3076923076923077, 0.125, 0.35294117647058826, 0.10526315789
3684, 0.25, 0.26666666666666666, 0.3076923076923077, 0.125, 0.2352941176470588
 0.10526315789473684, 0.25, 0.4, 0.3076923076923077, 0.125, 0.2222222222222222
0.21052631578947367, 0.23529411764705882, 0.25, 0.5, 0.47058823529411764, 0.35
4117647058826, 0.10526315789473684, 0.25, 0.26666666666666666, 0.3076923076923
7, 0.125, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001]
4
['I', 'want', 'to', 'lay', 'and', 'expire', 'die']
0.010656849447781438
0.49693251533742333
0.0005829317581221825
0.001
0.001
0.001
0.001
Porbability of this sentence actually occuring
3.0870521150658292e-15
```







Determining the most accurate rhyme

Wu-Palmer Similarity: Return a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer

Wu-Palmer Similarity is from wordnet and is used to pick most accurate rhyme.

Brown corpus was used to train bigram model and determine the accuracy of entire sentence compared.(Brown corpus is mostly news articles and well written documents allowing us to compare rhymes to well written documents.

BROWN
UNIVERSITY

The Brown Corpus of Standard American English was the first of the modern, computer readable, general corpora. It was compiled by W.N. Francis and H. Kucera, Brown University, Providence, RI. The corpus consists of one million words of American English texts printed in 1961.

The Brown corpus consists of 500 texts, each consisting of just over 2,000 words. The texts were sampled from 15 different text categories. The number of texts in each category varies.

Best case, and the best input

The best input is poetry or abstract writing with NO concatenations.

Placing similar meaning words at the end of sentences is an easy way to ensure high probability.

```
['the', 'darkness', 'comes', 'with', 'every', 'night']
['one', 'sleep', 'more', 'we', 'will', 'lose', 'that', 'battle']
fight
[0.26666666666666666, 0.2857142857142857, 0.14285714285714285, 0.2666666666
66, 0.42857142857142855, 0.0001, 0.26666666666666666, 1.0]
7
['one', 'sleep', 'more', 'we', 'will', 'lose', 'that', 'fight']
0.001
0.001
0.001
0.021794221996958945
0.0013611615245009074
0.001
0.001
0.001
Porbability of this sentence actually occuring
2.9665456438691844e-20
```

```
There was a dog in the street
0.294947121034077755
0.07936994988237701
0.0006398245052785522
0.05714285714285714
0.28388615888615887
0.0014191634908232744
0.0014191634908232744
3.448257877406973e-10
```

```
prob_sentence = cprob_brown_2gram["how"].prob("do") * cprob_brown_2gram["do"].pr
print(prob_sentence)
 result: 1.5639033871961e-09
```

What I would change

If possible I would use an Natural Language Generation model that uses deep learning to completely break down the meaning of a sentence and then using NLG methods create a new sentence with similar meaning but with specific input(synonyms/rhymes) could be created.

There is a lot of work being done on how to break down and analyze different sets of data, but minimal work has been done on representing similar data as separate entities.

Maybe use a different corpus to train my bigram model over to show higher accuracy.

Thank You