

EE4146 – Informatics and Learning Systems

Week 2

Introduction to NLP

Tommy WS Chow

CityU EE

January 2025

Classification accuracy NOT always a reliable metric

- In Machine learning performance evaluation, we often need to overcome the limitation of “accuracy” measure.
- **Accuracy**: classify 100 instances with 95 correctly classified, 5 incorrect gives an accuracy of 95%
- Is this always a good metric to rely on?
- Problem arises when there is a **significant class imbalance** issue
- 2 cancerous out of 100 patients cancer diagnosis
- 98 patients are benign (noncancerous)
- 99% accuracy but 1 cancer case misclassified as benign
- Is it good enough?



CONFUSION MATRIX

		Actual Value	
		P	N
Predicted Value	P	TP	FP
	N	FN	TN

TP - True Positive
TN - True Negative
FP - False Positive
FN - False Negative
P - Positive
N - Negative

Confusion Matrix

Classification: True vs False and Positive vs Negative

Example: The boy who cried wolf

“**Wolf**” **+VE** positive class,

“**No wolf**” is **-VE** negative class

True Positive (TP): - Reality: A wolf threatened. - Shepherd said: "Wolf." - Outcome: Shepherd is a hero.	False Positive (FP): - Reality: No wolf threatened. - Shepherd said: "Wolf." - Outcome: Villagers are angry at shepherd for waking them up.	+VE
False Negative (FN): - Reality: A wolf threatened. - Shepherd said: "No wolf." - Outcome: The wolf ate all the sheep.	True Negative (TN): - Reality: No wolf threatened. - Shepherd said: "No wolf." - Outcome: Everyone is fine.	-VE

True positive: an outcome the model *correctly* predicts the *positive* class.

True negative: an outcome the model *correctly* predicts the *negative* class.

False positive: an outcome the model *incorrectly* predicts the *positive* class.

False negative: an outcome the model *incorrectly* predicts the *negative* class.

Classification Metrics

- **Confusion matrix**
 - A matrix of numbers indicates where a model gets confused.
 - Class-wise distribution of the predictive performance.
 - Can only be used in supervised learning, where the labels are known.
 - It compares the predicted values with the true values.
- **TP**: classifier predicted **POSITIVE** (they **have disease**), and correct class is POSITIVE (patient have disease).
- **TN**: model predicted **NEGATIVE** (**no disease**), and correct class is NEGATIVE (patient do not have disease).
- **FP**: classifier predicted POSITIVE, but **correct class is NEGATIVE** (patient do not have disease).
- **FN**: classifier predicted NEGATIVE (patient do not have disease), **but patient has disease. This could be fatal.**

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure: Confusion Matrix for a binary class dataset

There are several evaluation metrics target for different needs

Classification Metrics

- Accuracy
 - The number of samples correctly classified out of all the samples present in the test set.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Most often used but not necessary the most suitable one to certain applications

Class balancing is an important issue

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure: Confusion Matrix for a binary class dataset

Classification Metrics

- Precision

- The number of TRUE Positive samples out of all the predicted Positive samples
- When model predicted TRUE class, how often was it right?

$$Precision = \frac{TP}{TP + FP}$$

- FP=0 is desirable

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure: Confusion Matrix for a binary class dataset

- To raise the precision, one must reduce the FP.
- House alarm system is a kind of example, as we do not want a flying bird to activate a house alarm system and call the police few times a day.
- Precision is suitable in cases where false positives are costly. For instance, recommendation system that flags “safe videos” for kids should have high precision.

Classification Metrics

- Recall (Sensitivity)
 - The number of samples **correctly predicted as positive** out **of all the positive samples**.
 - The ability of finding the positive cases, or how often it may miss out the positive case

$$Recall = \frac{TP}{TP + FN}$$

- Boeing engine fault testing is a good example to minimize FN (critical to safety).
- False negative would have terribly critical consequence

In brief:

- Accuracy shows how often a classification model is correct overall.
- Precision shows how good a model is finding the positive samples.
- Recall shows if the model can minimize costly false negative cases such as cancer and jet engine fault detection

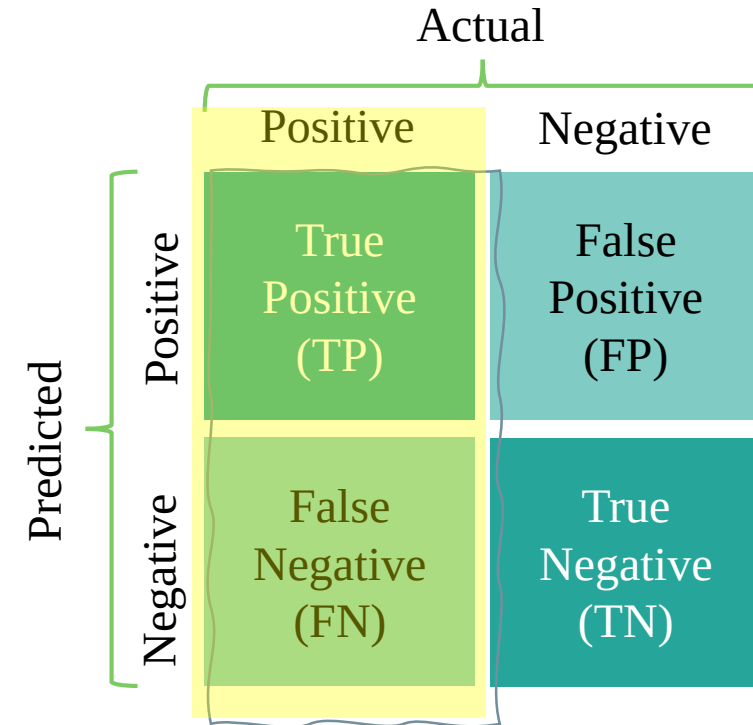


Figure: Confusion Matrix for a binary class dataset

Document Searching

- Unsupervised: document clustering in search engine, where **labels are missing**.
- **Precision** of a search engine: when given a query, return only the clusters of documents that are relevant to the query (i.e., **reduce false positive**)
- **Recall** of a search engine: when given a query, return the documents relevant to the query and all documents belonging to the correct cluster (i.e., **reduce false negative**)

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Precision can be seen as a measure of **quality**, and **recall** as a measure of **quantity**.

Higher precision: algorithm returns more relevant results than irrelevant ones

High recall: algorithm returns most of the relevant results (whether or not irrelevant ones are also returned)

Classification Metrics

- F1-Score.
 - It measures a model's accuracy
 - It **combines precision and recall scores** of a model (harmonic mean of precision & Recall)
It integrates precision and recall into a single metric to give a better understanding on model performance
 - It computes how many times a model made a correct prediction across the entire dataset

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

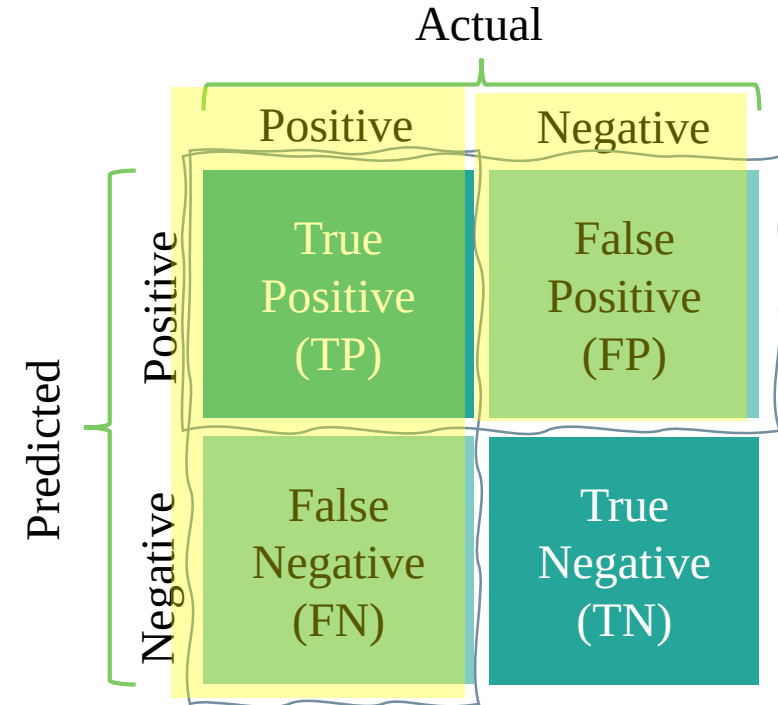


Figure: Confusion Matrix for a binary class dataset

Classification Metrics

- Specificity (Selectivity)
 - The number of samples predicted correctly as negative class out of all negative samples.

$$\text{Specificity} = \frac{TN}{FP + TN}$$

- Useful when the objective is to measure the model's ability to correctly identify the negative class instances.
- For example, in medical diagnostics, a high specificity is essential to avoid unnecessary treatments due to false positive

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure: Confusion Matrix for a binary class dataset

Introduction to Natural Language Processing

What is a Large Language Model?

Large Language Model LLM is the most commonly heard AI term since ChatGPT became available to public

- What is Language Model (LM)?
- LM aims to model the generative likelihood of word sequences, so as to predict the probabilities of future words (tokens).
- It can then be used for language generation (using Recurrent Neural Network)
- Machine translation: from one language to another language (using LSTM or RNN)
- Language sequence labeling/prediction (usually based on RNN)
- A large language model (LLM) is a pre-trained language model (PLM) of significant size (e.g., containing tens or hundreds of billions of parameters), which displays some surprising task solving abilities that are not present in smaller PLMs.

How Do Large Language Models (LLMs) Work?

- At their most basic LLMs are statistical pattern-recognition and prediction systems
- Tokens are building blocks of natural language. Tokenization is a way of separating piece of text into smaller units called tokens.
- LLMs output the next likely word (“token”) in a sentence (“sequence”)
 - sequence: context - section (“window”) of text e.g. sentence, paragraph, book
- The likelihood of the next word appearing is determined by
 - the context in which the words are seen in a larger body of **text (“corpus”)** and
 - the input to the chat

LLMs “Understand” “Meaning”

- Learning from a large corpus allows LLMs to understand the meaning of words.

For example

- the training data may consist of many sentences beginning with “my favourite colour is...”
 - the next word will be a colour, allowing LLMs to cluster the words “red, blue, green...” into a set that represents the concept of “colour”
-
- It’s important to note that LLMs don’t really understand anything. **They create statistical patterns that groups similar tokens based on a complex measure of how similar or dissimilar they are.**

Data used to Train LLMs

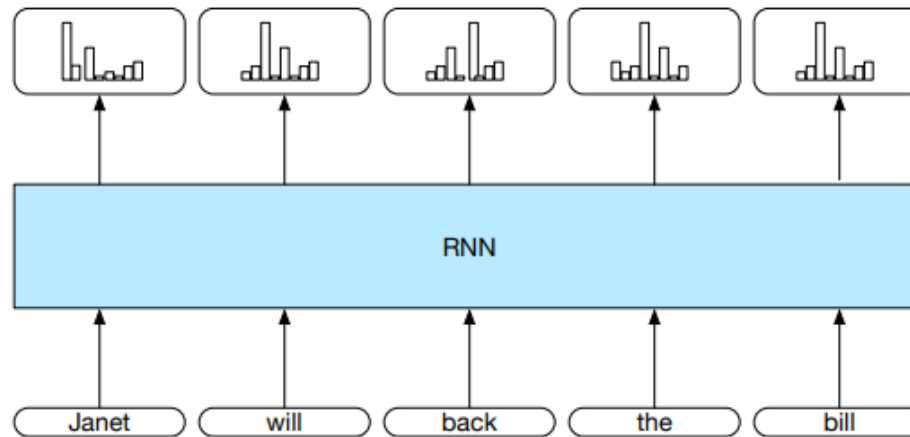
- LLMs are trained in an unsupervised manner on vast quantities of open source and licensed data e.g.
 - Common Crawl (~20B URLs)
 - GPT3: 175B parameters; GPT4: undisclosed: est. 500B – 1000B - bigger is better (for now, at least)
- Responses are refined using question-response pairs (“InstructGPT”) from the web, humans or bootstrapped (i.e. the LLM outputs its own pairs)
- reinforcement learning with human feedback (RLHF) is used to reward LLMs to give an improved responses

Large language model (LLM) and language sequence labeling

- Sequence labeling: assign one label to each element in a language sequence

RNN Architecture:

Each time step has a distribution over output classes



Evolution of Language Model

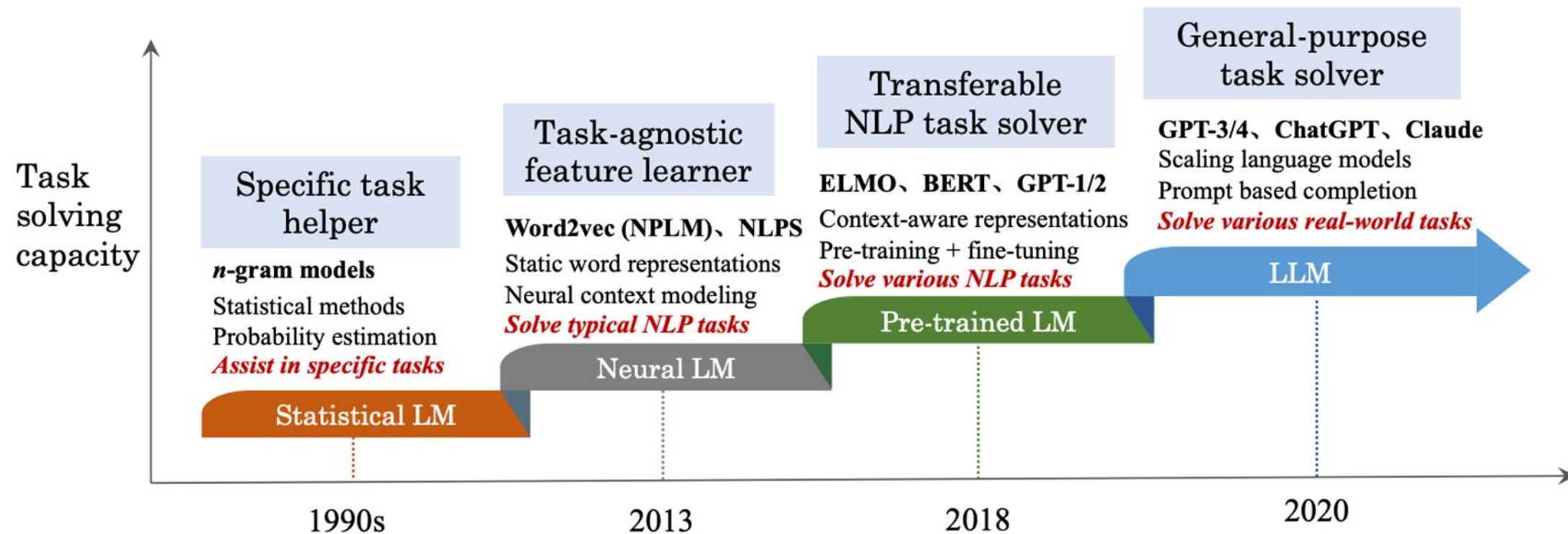


Figure*. An evolution process of language models

Large Language Model (LLM)

- **Large** language model is a deep learning technology applied to natural language processing
- LLM mostly use a deep learning architecture. **Transformer** is currently the most widely used and powerful deep-learning architecture to perform LLM applications like language recognition, prediction, translation, generation and other related applications
- LLM must be trained making it works like human brains so that it can perform applications in the past that only humans can handle, ChatGpt is typical example
- LLM also works on estimating the probability of a language token. Typical example

When I heard my mother shout, I _____ .

<i>knew she was cooking in the kitchen</i>	70%
<i>knew she was talking to my Dad</i>	15%
<i>knew she was on the phone</i>	8%
<i>knew she had trouble</i>	5%
<i>guess she hurt herself</i>	2%

LLM: Next word prediction

- LLM is influenced by the number of occurrence of words seen in many many contexts (how commonly used are those words)
- Still there are certain degree of randomness so that the appearance of words is not straightly based on the ranking of probability, i.e., 98.76% may not appear, whereas 97.8% word may appear instead
- This caters the human language randomness used by different people. This also eliminates the mechanical robot type output making output very human
- The model is mostly based on western countries culture and usages. A person come from **XX** country that is never exposed to internets might get a totally incorrect output

Prediction in favorite color

For example:

Green 12%

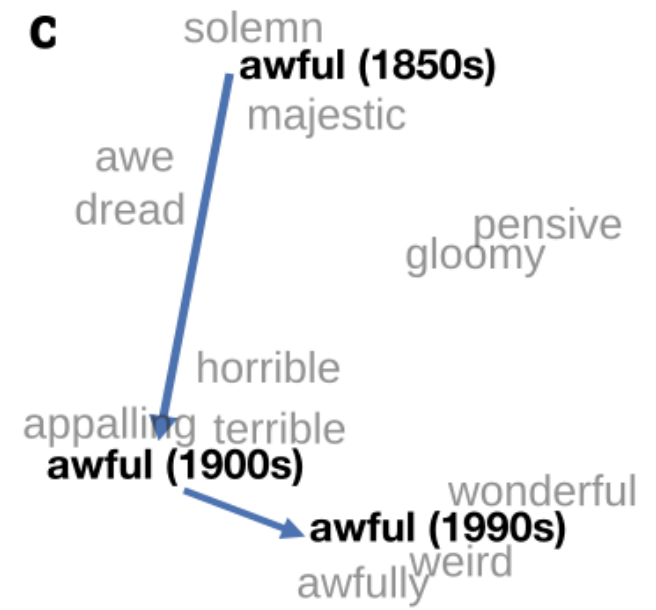
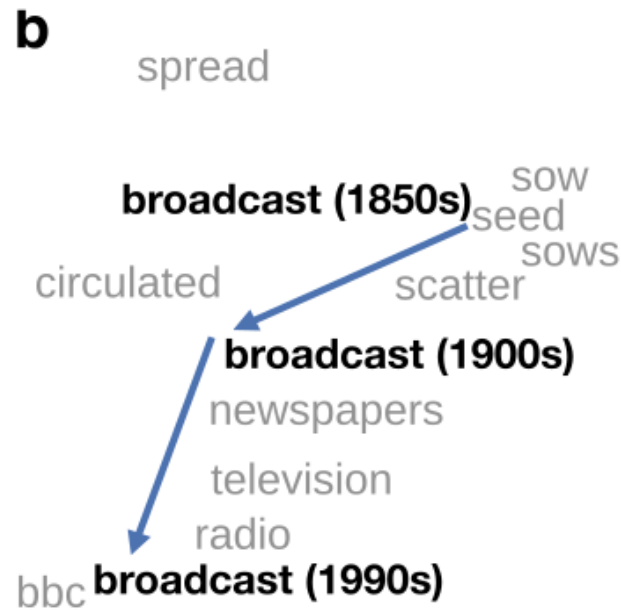
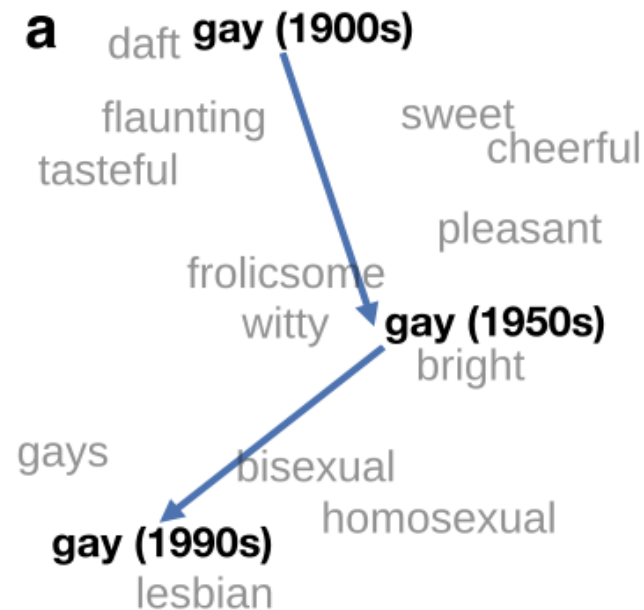
Red 17%

Blue 14%

Puce 1.5%

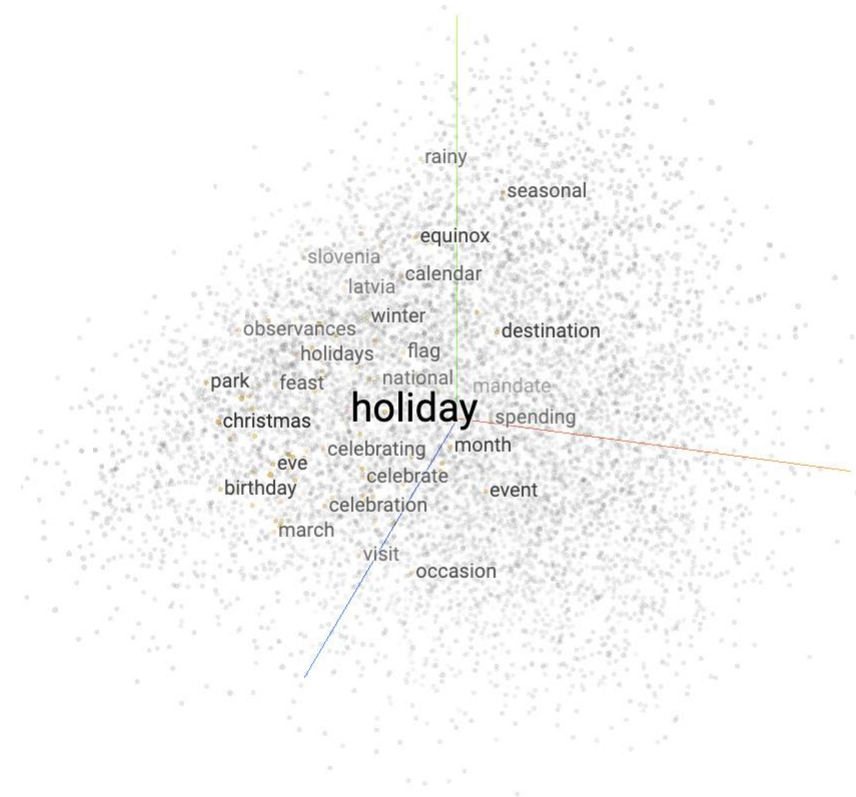
Meanings can change based on context

In each these examples the meaning of the same word changes over time



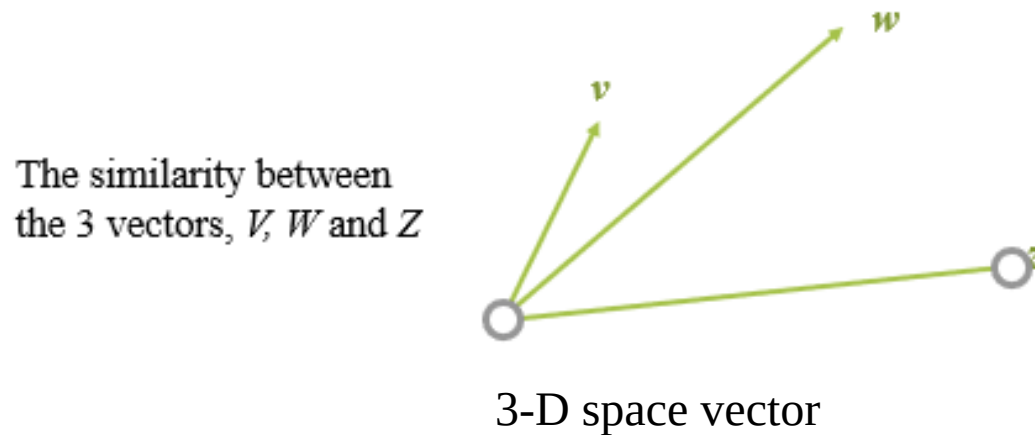
Fundamental Tools behind LLM

- **Word embeddings (vectors)**
 - An approach with which we represent documents and words.
 - Words are represented in a high dimensional vector that depends on the way of encoding
 - **Similar words** are “nearby in the word high dimensional space”, (Xmas, present, holiday)
 - **Contextualized features** that capture the semantic and syntactic information of words.
 - **Learned** from a large corpus of text using various techniques such as Word2Vec, BERT, etc.
 - **Fundamental** for training language models for various tasks.



Fundamental Tools behind LLM

- Word/ sentence/ paragraph/ documents as high dimensional vectors. Very common to be over few hundreds even after dimensionality reduction
- Similar sentence/ paragraph/ document compared in high dimensional space



Introduction to Natural Language Processing (NLP):

And Why NLP is a highly complex problem

- Ambiguity
 - Lexical: **silver** (noun, adjective, or verb ?)
 - Syntactic: Buy books for children. (Syntactic analysis involves analyzing the grammatical syntax of a sentence to understand its meaning.)
 - Semantic: “We saw her duck.” (computer understands the meaning of a text by analyzing the text as a whole and not just looking at individual words. RNN, LSTM solve the problem)
 - Discourse: The horse ran up the hill. It was very steep. It soon got tired. (with multiple sentences)
- Non-standard English
 - “Were SOO PROUD of what youve accomplished!”, “ur“, “dude”, “oops” (Typical not proper English appears on Twitter or FB)
- Segmentation issues
 - Multiword expressions are complex lexical units in English and NLP: *bite a bullet, make a call, ran into, make up, live with it, crush in, crush on*
- Scale
 - Over 470,000 entries in Webster's Third New International Dictionary, Unabridged.

Machine learning and NLP

- 90% of World's data is held in an unstructured (textual) format. Situations are getting even serious when we are dealing with social platform language processing (people seldom use formal English)
 - e.g.: e-mails, Opinion post, Facebook posts, X, webpages, press articles, etc.
- The amount of data available is rapidly growing in size. There is a need to:
 - Organize and classify text document databases to facilitate interpretation.
 - Extract useful information from large textual resources (e.g., trends, economical situation, emotional state of individuals etc.).

People need to use different kinds of language style under different platform, or pretty awkward

Applications of NLP

- Classification of news stories according to their content.
- Posts filtering in X, Facebook.
- Classification of documents in large databases.
- Extraction of information about entities and relationships between them (e.g., relationship between countries, corporations, etc.)

Example: OpenAI, headquarters at Mountain View California, was started by ..

- Analysis of the perception of the society regarding a particular event (e.g., US President election, Brexit, movie release) through automatic extraction of emotion in tweets, comments, etc.
- Understanding customers satisfaction comments (usually short) and not complete, ie., this product is s..t, s..k!

Text summarization

- Producing a shorter version of a document while preserving its important information.

Abstractive

- Generating entirely new phrases and sentences. (GPT-4 Generative pretrained Transformer)
- **Pros**
 - Can generate more human-like summaries.
 - Can summarize text that does not have a clear structure.
- **Cons**
 - Can introduce false or toxic information that is not present in the original text.
 - Can be computationally expensive.

Extractive

- Selects a subset of sentences from the text.
- **Pros**
 - Preserves the original meaning of the text.
 - Can be computationally efficient.
- **Cons**
 - May not be able to capture the essence of the text.
 - May produce summaries that are less readable.

Example 1 of Text Summarization

Source

100 pages of newspaper articles related to the Australian mining industry.

Query

What are the **obstacles** encountered in the development of Australia's uranium **mine** project in Kakadu **National Park**?

Automatic Summarization

‘Environmental groups and local **Aborigines oppose the mine.**’

‘The Australian branch of the International Council on Monuments and Sites said the **impact on archaeological and rock art sites** in the region of the mine had not been properly assessed.’

‘The team from the **U.N. World Heritage Bureau** is visiting the park to examine claims that the **environmental and cultural values** of the park are being threatened by the Jabiluka mine.’

‘It said the mine, which could process 19.5 million metric tons of uranium ore over its 28-year life, could **damage the environment.**’

Example 2 of Text Summarization

Source

100 pages of articles related to the Oslo Accord

Query

Identify the **principles and the terms** of the **Oslo Accord** of 1993 between Israel and the Palestinian Authority.

Automatic Summarization

‘Following months of secret talks held in Oslo, the Israeli and Palestinian negotiators reached agreement on August 20 of 1993 on principles of Palestinian self-rule, widely-known later as the **Oslo Accords** which started the Israeli-Palestinian **peace process**.’

‘Under the **agreement**, Israel will **hand over** another 13.1 percent of the West Bank to the Palestinians and open two roads between the West Bank and Gaza Strip as safe passage. ‘

‘The Palestinian side **convenes** its National Council to **abrogate** parts of the anti-Israeli articles from its Chart and other concrete plans to strengthen security **cooperation** with Israel.’

‘The **Oslo Accords** prepared the ground for the Israeli-Palestinian **peace process** on the **principle** of “land-for-peace.”’

Opinion mining

- Aiming to extract opinions and sentiments from natural language text using computational method.
- To perform analysis on people's opinions, sentiments, appraisals, attitudes, emotions expressed in text.
- Opinion summarization aims to represent subjective content from single or multiple documents in a brief and fluent manner.
- Opinion: concrete view
 - “Just the right length and neither fitted nor loose.”
- Sentiment: feeling
 - “If you're looking for an undershirt this pack of Gildan's are PERFECT!”

A Typical Result of Opinion Summarization

Below is a typical result based on a 72.26 million reviews on Amazon Kindle Review dataset

Method	Summary	TOV Score Lower the better
Human	The font is clear and adjustable in size. The amount of whitespace as well as font size on screen is configurable. Font size is easily adjusted depending on your preferences. Even if the font is large, multiple paragraphs can still be viewed on screen. The text is easy to identify due to crisp and precise contrast. Adjustable font makes reading easy and enjoyable. Varying font size. Clear and adjustable font.	0.00
QT*	All the fonts look exactly the same, regardless of the book. font sizes and page orientation can be adapted to the reader's needs. With good lighting it is very readable and the adjustable font makes up for this shortcoming. azw conversion the pdf will usually have teeny tiny unreadable font, this solves that, text to speech it, etc. azw you can change the font size of the document.	1.61
TSOSum^	Being able to change the font sizes is awesome! All the fonts look exactly the same, regardless of the book. The font is clear and adjustable in size. Very accessible and friendly for those with a visual impairment, all raised keys, font sizing, text to voice. The print appears faded especially with the smaller font sizes. Sleek style, easy to hold, the white causes it to disappear as suggested in its description, highly transportable, well thought out button placement, easy to use, adjustable font, ease of downloading new books.	1.49

Recent developments in NLP

- Generative Pre-trained Transformer (GPT)
 - It is a family of **large language models (LLMs)**
 - They are based on a **Transformer** Deep Neural Network architecture that has been pre-trained on vast amounts of text data using unsupervised learning.
 - These models are capable of generating human-like text and have significantly advanced the field of NLP.
- ChatGPT: A recent, intriguing set of capabilities.
 - Further enhanced by Reinforcement Learning (RL) from Human Feedback (RLHF).
 - A "reward model" directly from human feedback.
 - Improving the robustness and exploration of RL agents.
 - Fine-tuning the "reward model" using Proximal Policy Optimization.



Challenges of NLP

- Text data is in an unstructured form: text units (words, sentences) contain a hidden structure that a computer is unable to capture.
- Online text databases may contain a very large number of texts and may be dynamic and vary with places (continuous stream of text data), ie., some HK young guys use “*big 6*”.
- Texts have a potentially high “dimension” (potentially large number of words in a text).
- Relationships between concepts may be subtle and complex.
 - e.g., “Arcelor was bought by Mittal” vs “Arcelor merges with Mittal” (subtle difference).
- Word meanings may be ambiguous and context dependent.
 - e.g., “crush”, “duck”, “saw”, etc.
- Noisy data: documents may contain spelling or grammar mistakes. Huge problems to deal with social platform type language.
- *We need statistical methods and/or domain specific knowledge to transform textual data into a structured, “computer understandable” format.*

Probabilistic Language Models

- Probabilistic Language Models can be used to assign a probability to a sentence
 - Prob (very cold tonight) > Prob (Extremely cold tonight)
 - Spell Correction: Hm is a student of CityU Prob (He is a) > Prob (Hen is a)
 - Speech recognition: Prob (His bag's black) > Prob (His bag's back) “l”
 - Language translation: Prob (I saw him in the morning) > Prob (I saw him every morning)

Probabilistic Language Model

- Our goal is to compute the probability of a sentence or sequence of words

$$W (=w_1, w_2, \dots, w_n): P(W) = P(w_1, w_2, w_3, w_4, w_5 \dots w_n)$$

What is the probability of the upcoming word? – $P(w_5 | w_1, w_2, w_3, w_4)$

- A model that computes these:

– $P(W)$ or $P(w_n | w_1, w_2 \dots w_{n-1})$ is called a probabilistic language model

Chain rule, Bayes rule, Conditional probabilities are all the essential tools for these part of work

Language Model (N-grams)

- Models that assign probabilities to sequences of words are called language models (LMs).
- **N-gram** is a widely used simple language model that assigns probabilities to sentences and sequences of words
- An n-gram is a sequence of N words:
 - A 1-gram (unigram) is a single word sequence of words like “Hello” or “wonderful”.
 - A 2-gram (bigram) is a two-word sequence of words like “back up”, “good morning”, or “my assignment”.
 - A 3-gram (trigram) is a three-word sequence of words like “I love you”, or “**see you tomorrow**”.
- We can use n-gram models to estimate the probability of the last word of an n-gram given the previous words, and also to assign probabilities to entire word sequences.

Language Model: N-grams

- The intuition of the n-gram model (simplifying assumption): – instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

$P(w_n w_1 \dots w_{n-1}) \approx P(w_n)$	unigram
$P(w_n w_1 \dots w_{n-1}) \approx P(w_n w_{n-1})$	bigram
$P(w_n w_1 \dots w_{n-1}) \approx P(w_n w_{n-1} w_{n-2})$	trigram
$P(w_n w_1 \dots w_{n-1}) \approx P(w_n w_{n-1} w_{n-2} w_{n-3})$	4-gram
$P(w_n w_1 \dots w_{n-1}) \approx P(w_n w_{n-1} w_{n-2} w_{n-3} w_{n-4})$	5-gram

N-grams: Computing Probabilities of Word Sequences

- Unigram

$$P(<s> \text{ the man from Jupiter came } </s>) = \\ P(\text{the}) P(\text{man}) P(\text{from}) P(\text{jupiter}) P(\text{came})$$

- Bigram

$$P(<s> \text{ the man from Jupiter came } </s>) = \\ P(\text{the}|<s>) P(\text{man}|\text{the}) P(\text{from}|\text{man}) P(\text{jupiter}|\text{from}) P(\text{came}|\text{jupiter}) P(</s>|\text{came})$$

- Trigram

$$P(<s> \text{ the man from Jupiter came } </s>) = \\ \frac{P(\text{the}|<s> <s>) P(\text{man}|<s> \text{ the}) P(\text{from}|\text{the man}) P(\text{jupiter}|\text{man from})}{P(\text{came}|\text{from jupiter}) P(</s>|\text{Jupiter came}) P(</s>|\text{came } </s>)}$$

N-grams

- In general, n-gram is insufficient to model language accurately because languages have long-distance dependencies

“The assignment **file** that I saved in my computer **crushed**

- There is no easy and computationally non-heavy way, and N-grams can still do an effective job
- Bigger N means the model will be more accurate, but it will increase the computation burden accordingly
- Smaller N means the model is less accurate but is less computationally demanding
- In most practical cases, Trigram will be good enough
- Another drawback: The memory required for handling a corpora significantly increases when N increases

Current LLMs

- **BERT** is a family of LLM that Google developed and introduced in 2018. BERT is a **Transformer-based model** that converts sequences of data to other sequences. BERT has an architecture of Transformer and with over 342 million parameters.
- **GPT**: GPT was first introduced by Open AI in 2018 using Transformer deep learning architecture. GPT-3 was introduced in 2020 with more than 175 billion parameters.
- **GPT-4** is the largest model released in 2023, exhibiting excellent language fluency, and displaying superior command on grammar, vocabulary and syntax. It provides very updated accurate responses based on its gigantic knowledge. It can provide up-to-date information better than ChatGPT.
- **ChatGPT**, based on GPT-3.5 model, is excellent engaging in realistic conversations with users. It focuses on generating user-friendly and context-aware responses to users.
- **Ernie** is Baidu's LLM that powers the Ernie 4.0 Chatbot which was released in 2023. It is rumored that Ernie has **10 trillion parameters**.

GPT-4 is a highly complex model and requires huge computational power.

ChatGPT compared to GPT-4 is computationally less intensive

GPT

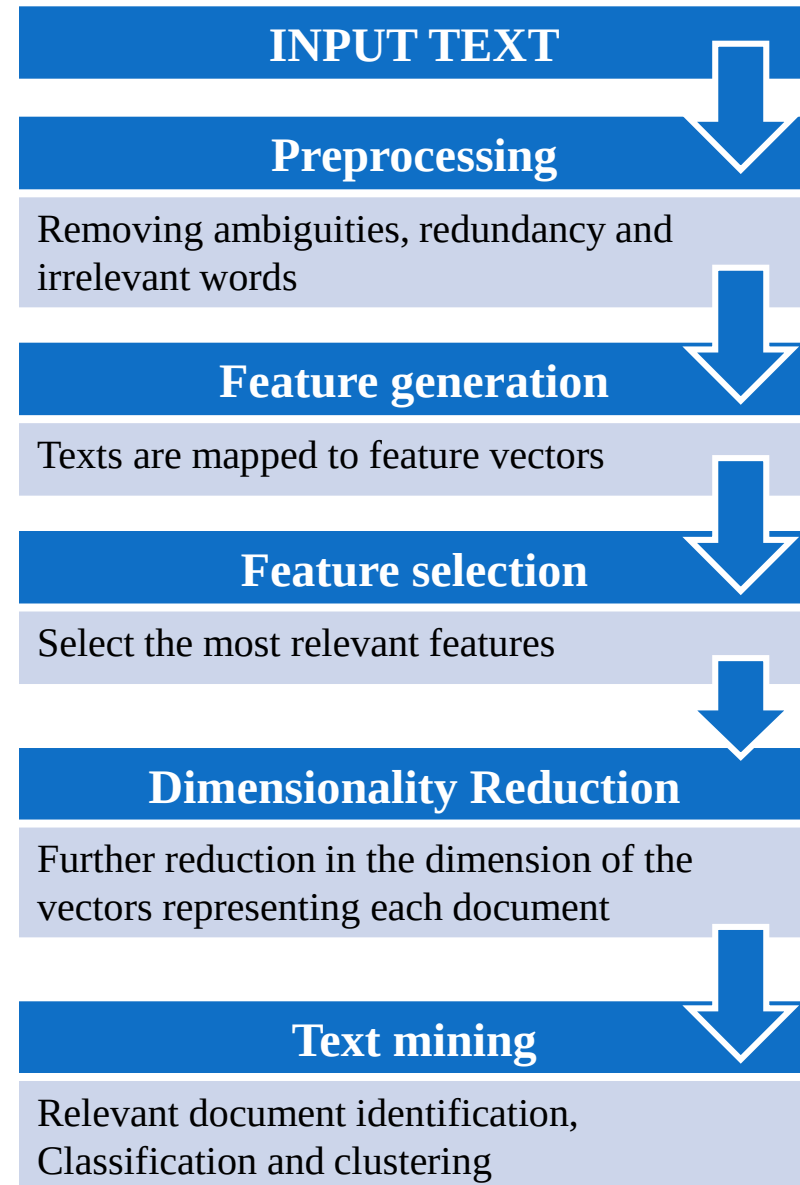
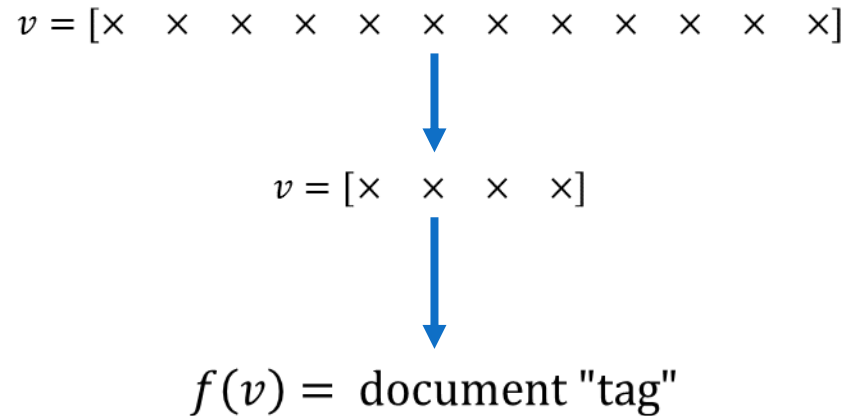
- GPT-3 memory 700 Gbytes; GPT-4 memory 700 terabytes
- GPT-3.5 short-term memory is around 8,000 words
- GPT-4's short term memory is around 64,000 words and is expected to grow to 128,000 in weeks or months
- The data used for training ChatGPT or GPT-4 is business confidential that is totally unknown outside OpenAI
- UC Berkeley researcher found that OpenAI research found GPT-4 are trained on text from copyrighted books and the degree of memorization is tied to the frequency that the passages appear on the web
- They also found that GPT-4 tends to bias toward science-fiction and fantasy books. GPT-4 has memorized all the Harry Potter series, Orwell's Nineteen Eighty-Fours, the Lord of the rings, Hunger Games, A Game of Thrones, Dune, which are all highly popular frictions in the English language market

AI generated text:

Will the generated text suffer from copyright infringement?

Will the generated text be copyright protected?

All steps can be finished by using Natural Language Toolkit (NLTK)[^] and scikit-learn*.



Preprocessing

- Goal: transform unstructured data (text) into a structured form (**numerical feature vector**, that is the inputs to classifier such as FNN, CNN or SVM).

IT IS THE LARGEST environmental enforcement recovery by the Department of Justice, exceeding even the \$400 million paid by BP in 2012 to resolve criminal proceedings over the 2010 Deepwater Horizon spill. Police at the Ministry of Public Security yesterday said 45 suspects at CSC's Chinese subsidiary had been identified as part of a "complete bribery chain" that funneled money to hospitals, doctors and government officials between 2009 and 2012. Mark Kelly, a Briton who was head of the unit, ordered subordinates to offer the illegal payments, they said. The allegations will almost certainly lead to charges, which could strain China's relations with Britain. David Cameron, UK prime minister, had sought to any to the London-based company, regulators in the US and UK are closely monitoring events in China and could launch their own investigations, although legal experts said they were likely to wait for Beijing to complete its probe first. A senior official at the Chinese ministry's economic crime investigation bureau said "all departments" of CSC's China subsidiary were "fully engaged" in the alleged bribery. The company paid for the bribes by inflating the price of medicines in China, some of which were seven times more expensive than in other markets, the official said. According to the official, the company set up several internal units with code names such as "Operation Great Wall" and "Operation soaring eagle" to pay the bribes. The developments will send a chill through the international business community. So far very few expatriates have faced corruption charges. CSC has previously denied any systemic corruption. In a statement yesterday it said: "We take the allegations that have been raised very much to heart, and we have taken steps to offer the illegal payments, they said. The allegations will almost certainly lead to charges, which could strain China's relations with Britain. David Cameron, UK prime minister, had sought to limit any damage to the London-based company, Affiliated Managers Group could hardly resemble its food sales than Vanguard less, except in one respect: no two US companies have less experience of investor outflows. AMG, a holding company for boutique asset managers, has posted positive sales for 16 straight quarters. Unlike Vanguard, however, AMG specialises in higher priced actively managed equities and alternative investments, driving average annual profit growth topping 50 per cent for the past four years. Though the bulk of AMG's \$540m in assets are from institutional clients outside the US, it has made US retail growth a priority. It rolled out the AMG Funds retail brand last month, hired a US retail sales chief and prepared to advertise, US retail may not seem its most logical market, given that Vanguard's ruthless discounting and no-frills index products have long dominated, but Sean Healey, chief executive of AMG, says demand for boutique investing is building. "We don't need to convince anyone that passive is going away," Mr Healey says. "Rather, we need to convince investors that we are on the other end, on the alpha-generating end of the barrel." The long anticipated rotation out of fixed income and general risk aversion bodes well for AMG.



$v = [v_1 \quad v_2 \quad \cdots \quad v_N]$ high dimensional feature vector

Steps in preprocessing:

- Natural language processing operations (***tokenization, stop-word removal, stemming***),
- Feature generation based on statistical properties of words or sentences,
- Feature selection: selection of the most relevant features.

Step 1 Preprocessing: Tokenization

Goal: breaking the continuous character stream into meaningful constituents, in general words or phrases.

Tokenization: It is an important part for text data mining and text processing. It breaks down text into words, separates a piece of text into smaller units of tokens for further processing and classification. The word-level analysis enables computer to capture semantic meaning of a given document.

Example:

[Text mining algorithms aim at extracting information from textual data. Classification methods are often incorporated in the mining process. Indeed, classifying documents allows the identification of topics which is an essential step in the process of information extraction.]



*[Text] [mining] [algorithms] [aim] [at] [extracting] [information] [from] [textual] [data]
[Classification] [methods] [are] [often] [incorporated] [in] [the] [mining] [process] [Indeed]
[classifying] [documents] [allows] [the] [identification] [of] [topics] [which] [is] [an] [essential]
[step] [in] [the] [process] [of] [information] [extraction]*

Step 2 Preprocessing: Stop-word removal

Goal: Remove function words and common words that do not contribute much to the semantic understanding of the text.

e.g. “a”, “an”, “the”, “that”, “if”, “when”, “who”, etc.

- **Stop-words** are not useful features for classification.
- **Stop words carry no identity** of documents and will simply make the model less accurate, especially when there are so many stop-words compared to other representative words used for classification and clustering.

Example:

[Text mining algorithms aim at extracting information from textual data. Classification methods are often incorporated in the mining process. Indeed, classifying documents allows the identification of topics which is an essential step in the process of information extraction.]



[Text] [mining] [algorithms] [aim] ~~[at]~~ [extracting] [information] ~~[from]~~ [textual] [data] [Classification] [methods] ~~[are]~~ ~~[often]~~ [incorporated] ~~[in]~~ ~~[the]~~ [mining] [process] ~~[Indeed]~~ [classifying] [documents] ~~[allows]~~ ~~[the]~~ [identification] ~~[of]~~ [topics] ~~[which]~~ ~~[is]~~ ~~[an]~~ [essential] [step] ~~[in]~~ ~~[the]~~ [process] ~~[of]~~ [information] [extraction]

Step 3 Preprocessing: Stemming

Goal: Reducing words to their root form (received, receiving to receive) to reduce the number of words and ease the process of query matching for search engines.

- This is an important procedure for text clustering/classification.

e.g. “fishing, fished, fisher” → “fish”, “argued, argues, arguing” → “argue”, etc.

Example:

[Text mining algorithms aim at extracting information from textual data. Classification methods are often incorporated in the mining process. Indeed, classifying documents allows the identification of topics which is an essential step in the process of information extraction.]



[Text] [mining] [algorithms] [aim] [extracting] [information] [textual] [data] [Classification] [methods]
[incorporated] [mining] [process] [classifying] [documents] [identification] [topics] [essential] [step] [process]
[information] [extraction]

Tough job: “mining” or “mine”

Step 4 Preprocessing: Feature generation

Convert text into Numerical vector

- We represent each document as a vector with **numerical features** that extracted from text .
- The method must be good to maintain the information contained in the text.

General methods used for test feature generation include

1. **Bag-of-words**: document is treated as a bag of words or terms. Each word forms a feature and the order of words is usually ignored
 - Terms Frequency**: features represent the frequency in which a term appears in a document (Number of occurrences/Number of words in the document)
 - TF-IDF**: Term frequency-inverse document frequency is used to determine the importance or relevance of words/ phrases in a document amongst a given corpus (a collection of documents)
2. Part of Speech (**POS**): POS to figure out if a given token represents a proper noun, common noun, verb, adjective etc. Not a straightforward process and there are many POS, ie., 93.
3. **N-grams**: Retain the text sequences information in a given token. A more complicated process.

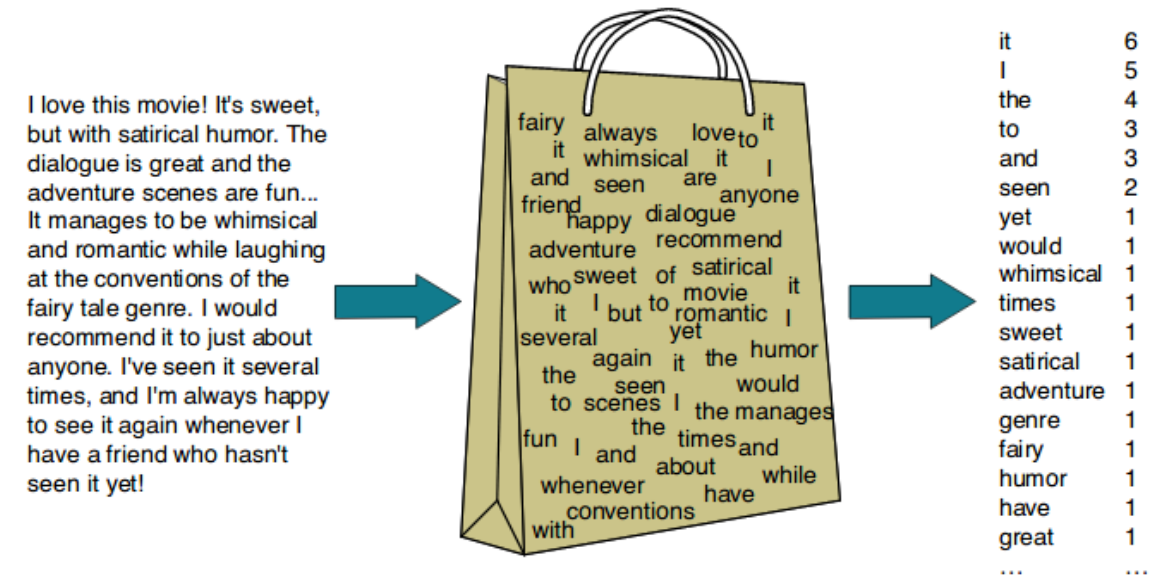
Step 4 Feature Generation: Bag-of-Words

Bag-of-words approach is the most commonly used method for document processing/ classification based on words frequency.

BOG is a simple method extracting features from text.

The idea is to represent each sentence as a bag of words disregarding grammar. Just to consider the occurrence of words in a sentence

- Form a huge corpus that contains many documents. All the words are based on this huge corpus
- Frequency of occurrence: count the number of each word occurs as a feature and is used to train a classifier
- TF-IDF: Term frequency-inverse document frequency

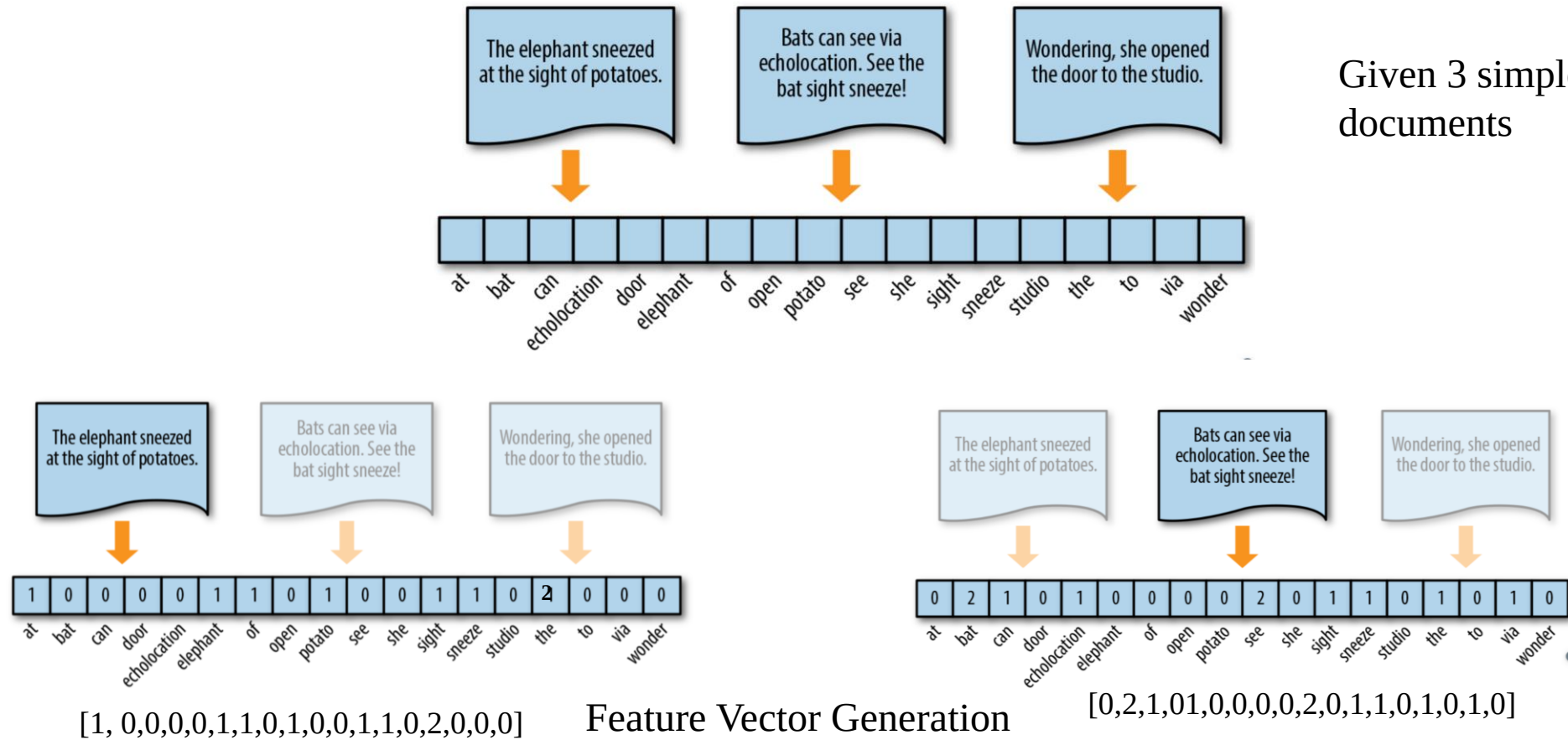


Acknowledge: <https://koushik1102.medium.com/nlp-bag-of-words-and-tf-idf-explained-fd1f49dce7c4>

- We have a huge corpus that are formed from different topics
- The dimensionality of vectors equal to the size of the vocabulary in a corpus
- word vector dimension is the dimension we use for training document. The norm can be around 1000

Step 4 Feature Generation: Bag-of-Words

Given 3 simple short documents



The problem is that the vectors (high dimensional vector i.e., 1000) that covers all the words of the corpus are usually very sparse because no single document used all the words of the corpus

Disadvantages of BOW

- Considering the frequency of certain word/term occurrence is a respectable way but not the most accurate way
- Massive amount of weights: High dimensionality of the input vectors create massive amounts of weight for feedforward neural network classifier.
- No meaningful relations or consideration to word order: The bag of words does not consider the order in which the words order appeared in the sentences or a text, whereas word order may mean something, ie., “Jonny fails to plan” vs “Jonny plans to fail”, vs “Jonny’s plan failed”
- Two words come together may mean different things: “computer science”, is different from “computer”, “science”
- Computationally intensive: With more weight comes, the need for more computation to train and predict irrespective of the type of classifier used.

Step 4 Feature Generation: TF-IDF

Term Frequency Inverse Document Frequency (TFIDF) If a term (word) appears frequently, it may , however, not a highly informative feature and its weight is decreased instead.

- TF (term frequency): IDF (inverse document frequency): indicates how common a word is used in a certain corpus, i.e., in football corpus “score”, “beat”, “foul”, “goal” appears a lot but may not mean highly special in the corpus. The high weights caused by the high frequency occurrence of these words may need to be suppressed in order to allow room for the real special terms to be pay noticed on, i.e., “collapsed”, “brain”, “concussion”, “final”. Another typical example, “coffee” in talking about coffee tasting corpus.
- IDF: For example stop word (“the” the most frequently used word) is not good feature to characterize a specific document. The word “coffee” is a very common and informative word in a corpus about politics. The word “coffee” thus needs to be given higher weights on a politics corpus.
- TF-IDF is a way of inverse extracting the weights in order to minimize certain super high weights terms in certain corpus.

For each **term** i in a **document** d , let

- n_i^d = number of occurrences of i in d ,
- n^d = total number of terms in d ,
- m = number of documents in the database,
- m_i = number of documents containing term i .

$$TF_i^d = \frac{n_i^d}{n^d}$$
$$IDF_i = \log \frac{m}{m_i}$$

$$TFIDF = (TF_i^d)(IDF_i)$$

- Documents contain “ball” in all football corpus, “ball” not an important word, but appears a lot
- $m/m_i=1$, $\log (m/m_i)= 0$
- Documents contain “heart” in football corpus is very rare, $m/m_i= 10000$, $\log (m/m_i)= 4$

Step 4 Feature Selection: TF-IDF

- The dimension of the feature vector is still very high (number of terms after stemming and stop-word removal). For large corpus containing many documents, the number of words involved may exceed a thousand.
- Not all features are relevant for text mining tasks (e.g., document classification)
- Feature vector of a document are also sparse (contains many zeros terms) as many other words may not have occurred in a document
- After forming the feature vector, we reduce the number of features using *dimensionality reduction* (e.g., **Principal Component Analysis** which will be introduced in this course).
- The dimensionality of the vector i.e., from 1500 will be reduced to ie., 200.

Google uses TF-IDF to determine rank the relevance of a document for a query.
TF-ID is also used in text classification, text clustering

The diagram illustrates the process of text mining. It begins with a large vector $v = [x_1 \ x_2 \ \dots \ x_n]$, where each x_i represents a document. A red oval highlights one specific document, $v = [x_1 \ x_2 \ \dots \ x_m]$. This document is then mapped to a feature vector, $f(v) = \text{document "tag"}$, which is shown in a green oval. The text "We now have feature vectors representing documents." is written in red between the two ovals. The final text, "Now we start text mining.", is written in green at the bottom.



A diagram showing a vector $v = [x \ x \ x \ x]$ enclosed in a red oval. A blue arrow points down to the oval, and another blue arrow points up from the oval.

$f(v) = \text{document "tag"}$

Now we start text mining.



Identification of relevant documents in a web search

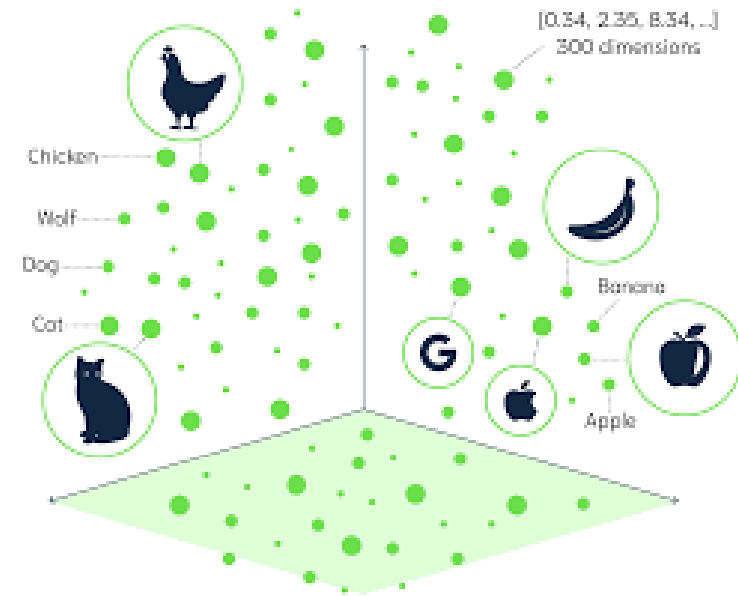
- **INPUT:** a set D of preprocessed documents formatted as feature vectors and a query q in the form of keywords (can be considered as a short document)
- **OUTPUT:** a ranking of all documents in D from the most relevant to the least relevant to q .

METHOD:

1. Project the query q into the feature space to obtain a feature vector v^q .
2. Define a similarity measure $s(v^i, v^j)$ for any pair of documents i, j with feature vectors v^i, v^j .
3. Compute $s(v^i, v^q)$ for all documents i in D and rank the documents accordingly.

Similarity Measures: Cosine Distance

- **Cosine distance** is generally used to determine how the two text documents close to each other with applications such as plagiarism detection
- In the most simplest case of 2-D space, cosine distance measures the cosine angle between two 2-D vector; $\cos(0)$ means the same 2 vector, $\cos(5)$ means very similar vectors, $\cos(180)$ means opposite direction vectors.
- Numerical value of $\cos(x)$; 1 means the same, 0 means different, -1 means very different.
- In real application, we are dealing with a high n-dimensional cosine distance (dot product calculation)
- In text application, cosine similarity measure is preferred over Euclidean distance because the high dimensionality space issue that ED may not provide the best reliable information. Cosine distance provides the frequency of occurrence that made text, LLM, documents application suitable.



Similarity Measures: Cosine Distance

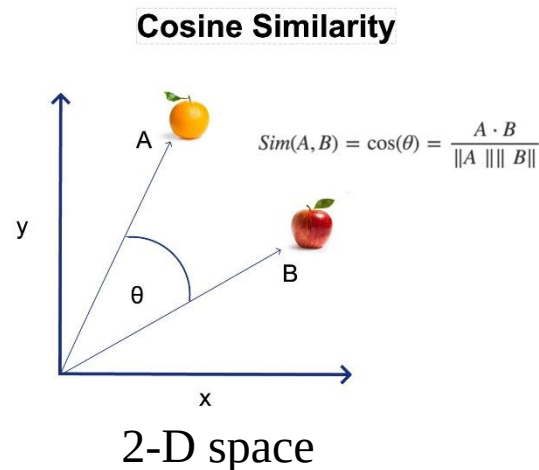
Two high dimensional vectors, $\mathbf{v}$, $\mathbf{w}$, representing 2 documents in the n dimensional space that is usually around 200 dimensions after using PCA as dimensionality reduction

$$V = (v_1, v_2, \dots, v_n) \text{ and } W = (w_1, w_2, \dots, w_n),$$

The cosine of the angle, θ , between these two vector can be computed through the dot product of the two vectors and dividing it by the product of their magnitudes

$$\text{sim}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v}^T \mathbf{w}}{\|\mathbf{v}\|_2 \|\mathbf{w}\|_2} = \frac{\sum_i v_i w_i}{\sqrt{\sum_i v_i^2 \sum_j w_j^2}},$$

where $\text{sim}(\mathbf{v}, \mathbf{w})$ (or cosine distance) is the similarity between vectors $\mathbf{v}$, $\mathbf{w}$



Similarity Measure: Cosine Distance

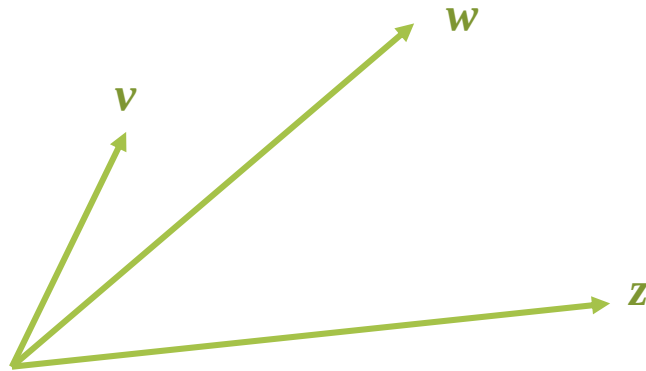
- Similarity based on angle measure:

$$\text{sim}(v, w) = \frac{v^T w}{\|v\|_2 \|w\|_2} = \frac{\sum_i v_i w_i}{\sqrt{\sum_i v_i^2 \sum_j w_j^2}}.$$

$$a \cdot b = \|a\| \|b\| \cos \theta$$

$$\begin{aligned} \text{similarity} &= \cos \theta = \frac{a \cdot b}{\|a\| \|b\|}, \quad a = [1, 3, -5], \quad b = [4, -2, -3] \\ &= \frac{(1)(4) + (3)(-2) + (-5)(-3)}{\sqrt{(1^2 + 3^2 + (-5)^2)} \sqrt{(4^2 + (-2)^2 + (-3)^2)}} \end{aligned}$$

The similarity between
the 3 vectors, V, W and Z



- Home exercise

The following data are collected from a SD RAM manufacturing plant fault diagnosis system. There are two types of data, namely normal and electrostatic fault. Each data consists of 3 attributes namely the magnitude of 3 different testing parameters. A typical case of normal and electrostatic fault data is given in the following.

Normal data
 $N = [-0.50 \quad -0.13 \quad 0.20];$

Faulty data
 $F = [0.4 \quad 10.0 \quad -0.35].$

Use cosine distance to determine if the following test data are normal, faulty, or otherwise. Justify your result.

Test data :

$T1 = [-0.20 \quad 0.21 \quad 0.18]$

$T2 = [-0.27 \quad -0.42 \quad 0.28]$

$T3 = [-0.50 \quad 10.30 \quad -0.26]$

$T4 = [-0.10 \quad 10.50 \quad -0.30]$

Feature generation: Part-of-Speech (POS) tagging

- What are parts of speech?
 - Noun, pronoun, proper noun, verb, auxiliaries, etc.
- A popular list of part-of-speech (POS) tags is Penn Treebank.*
 - Noun,
 - NN - Noun, singular or mass
 - NNS - Noun, plural
 - Pronoun,
 - NNP - Proper noun, singular
 - NNPS - Proper noun, plural
 - etc.

*Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313-330.

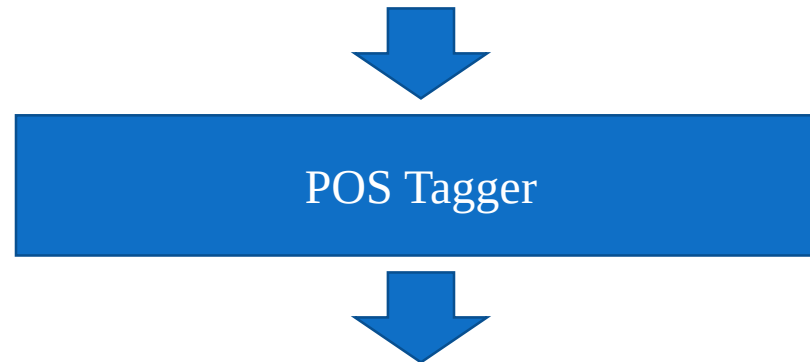
Revisit Feature generation: Part-of-Speech (POS) tagging

- Determine the POS tag for all tokens in a sentence.
- Words often have more than one POS:
 - The **back** door (adjective, JJ)
 - On my **back** (noun, NN)
 - Promised to **back** the bill (verb, VB)
- We cannot depend on a dictionary to determine the correct part-of-speech tags because of the presence of ambiguity.

Revisit Feature generation: Part-of-Speech (POS) tagging

- For a language (or domain)
 - Step 0: Define a POS tag set
 - Step 1: Annotate a corpus with these tags
 - Step 2: Choose a POS tagging model, e.g., Hidden Markov Models (HMMs)
 - Step 3: Train your model on your training corpus
 - Step 4: Evaluate your model on your test corpus

“There are many resources available to help you learn how to use Python.”



('There', 'EX'), ('are', 'VBP'), ('many', 'JJ'), ('resources', 'NNS'), ('available', 'JJ'), ('to', 'TO'), ('help', 'VB'), ('you', 'PRP'), ('learn', 'VB'), ('how', 'WRB'), ('to', 'TO'), ('use', 'VB'), ('Python', 'NNP'), ('.', '.')]

Revisit Feature generation: Part-of-Speech (POS) tagging

- POS tagging is one of the fundamental steps in the process (right after tokenization in pre-processing).
- POS tagging can be viewed as a prerequisite for further analysis:
 - For example, visualizing regional English, comparing style of English writing.

Feature generation: Part-of-Speech (POS) tagging

- A style marker refers to a textual feature or an attribute.
- 140 popular style markers which means to form a 140 dimension vector.*
- An extension of POS tags.

Type	Description	Example
Symbol	Numbers of the punctuations.	. ‘ “” ()
Vocabulary	Numbers of the POS tags in the Penn Treebank.	NN, VB
Syntax	Numbers of the POS tag combinations.	NNP VB
Derivative	Average Length of sentence, Average Length of paragraph, Intimacy, Number of stop words.	

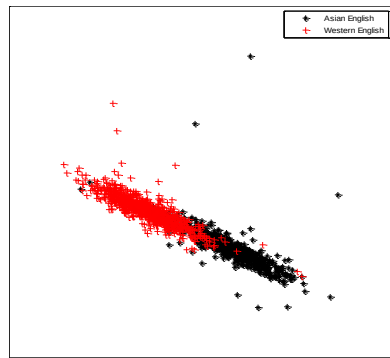
*Tang, P., Zhao, M., & Chow, T.W.S. (2016). Locality alignment discriminant analysis for visualizing regional English. *Neural Processing Letters*, 43, 295-307.

Examples to illustrate the significance of Visualization and Dimensionality Reduction

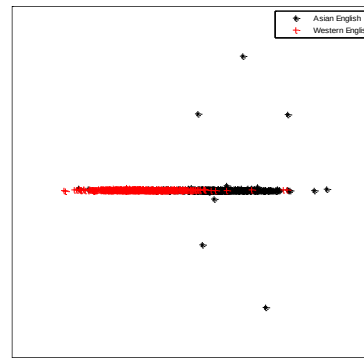
2015 results on Visualization of British, American & Asian English

2-D visualization (140 dimensional vector reduced to 2-D) Discrimination of Chinese, Japanese and Korean English from US, British and Australian English. We used D-R to reduce the 140 dimensions into 2 for visualization and classification

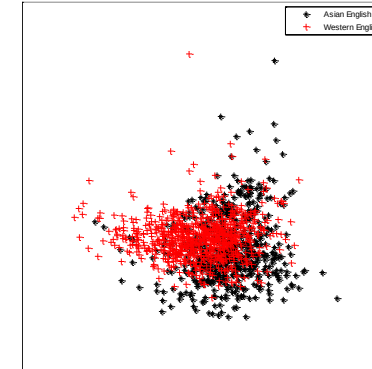
***** = Chinese, Japanese and Korean English **+** = the US, British and Australian



TR-LDA



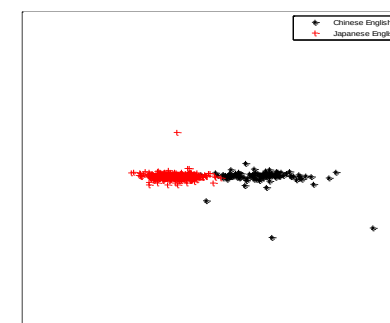
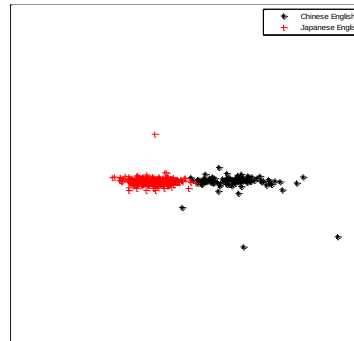
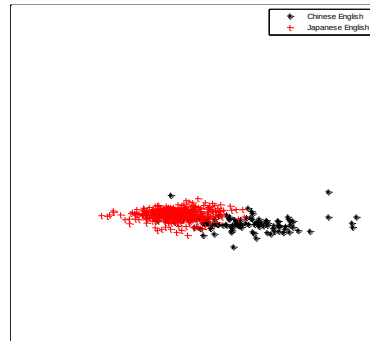
LDA



PCA

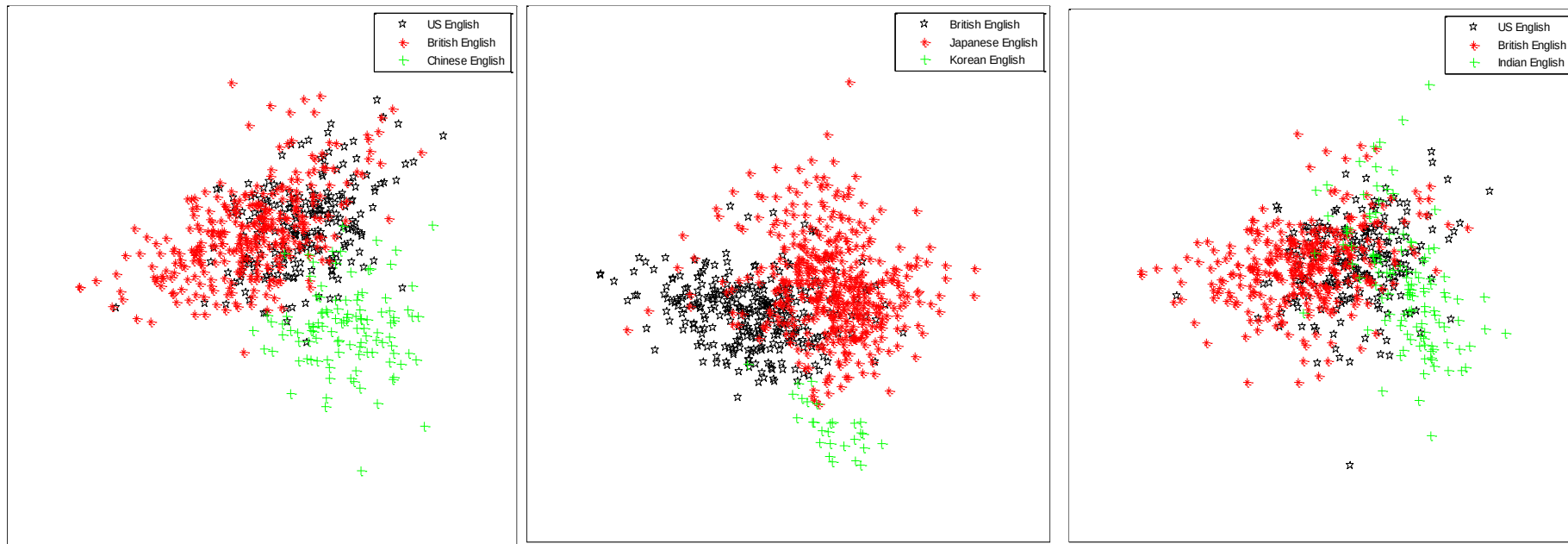
TR-LDA, LDA and PCA are different kinds of DR method aiming to provide good classification after D reduction

Discrimination of the **Japanese English** from **Chinese English**



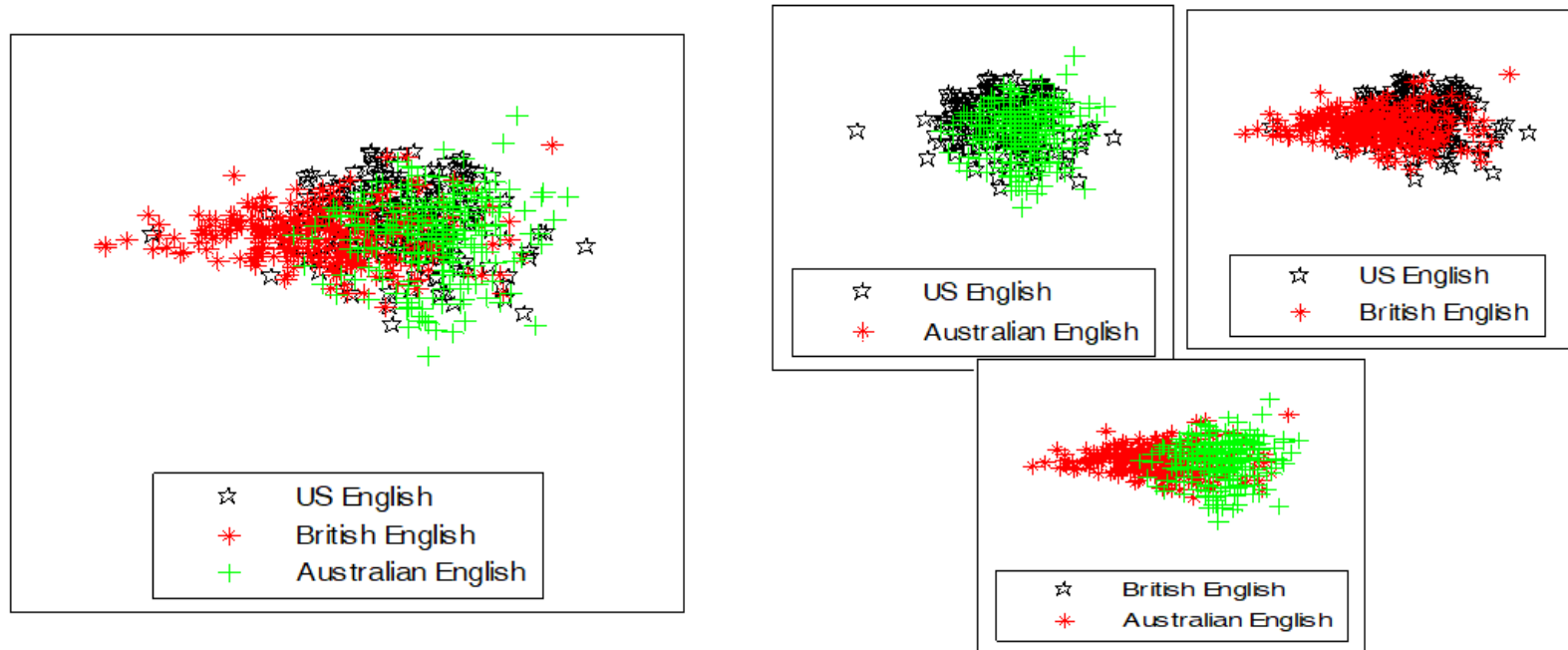
Text mining application: Visualizing regional English*

- Asian English vs the US and British English
- Indian English are closer to the British English
- There is clean boundary between Japanese and Korean English, implying their written English are significantly different in styles
- There is a relatively clean boundary between Asian English and Native English; Asian English is easy to be discriminated from the native English



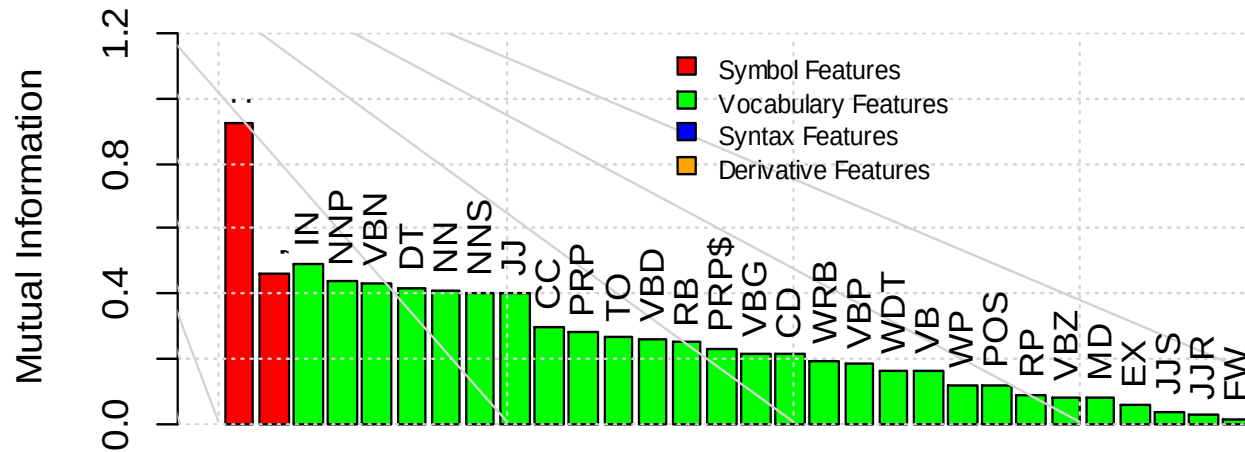
*Tang, P., Zhao, M., & Chow, T.W.S. (2016). Locality alignment discriminant analysis for visualizing regional English. *Neural Processing Letters*, 43, 295-307.

Text mining application: Visualizing regional English*



- We could not obtain a clean separation for the US, British and Australian English
- The US and Australian English are close to each other
- The British English drifts noticeably to the left
- The US and the British English exhibit significant overlapped, which means the difference from them is insignificant

Dominant Features for Discriminating Regional English



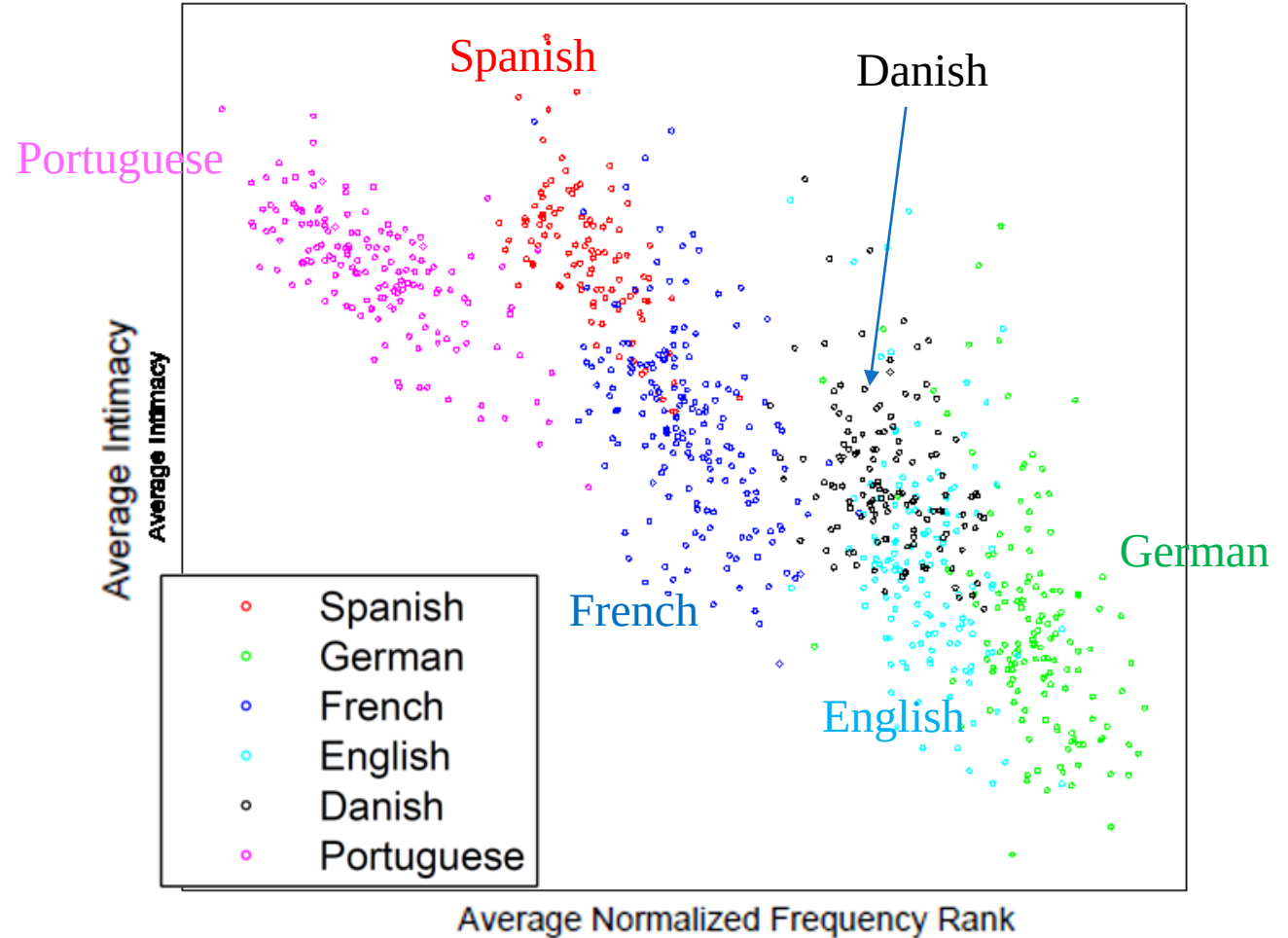
Use feature subset selection to analyze

Feature subset selection is different from D-R

- 1) The symbol features, i.e. using of colon symbol “:” and comma symbol “,” are the most significant features
- 2) Most of the dominate features are vocabulary features, for example, IN (preposition or subordinating conjunction) and NNP (proper noun, singular)
- 3) The use of stop words and average sentence length are very sensitive to distinguish the components.

Visualization Results on Holy Bible

- Different translations of the Holy Bible
 - Spanish
 - German
 - French
 - English
 - Danish
 - Portuguese



END