Compiler construction

Natural Language Processing (NLP)

CSE, IUS

Gratitude to Dr. P. Kuppusamy

Natural Language Processing

- NLP is the technology used to assist computers (intelligent system) to understand the human's natural language.
- It deals with the interaction between computers and humans using the natural language.

1. A human talks to the machine
2. The machine captures the audio
3. Audio to text conversion takes place
4. Processing of the text's data
5. Data to audio conversion takes place
6. The machine responds to the human by playing the audio file

Two Components of NLP

1. Natural Language Understanding (NLU)

- NLU helps the machine to understand and analyse human language by extracting the metadata from content such as concepts, entities, keywords, emotion, relations, and semantic roles.
- NLU used in Business applications to understand the customer's problem in both spoken and written language.
- NLU involves the following tasks -
  - It is used to map the given input into useful representation.
  - It is used to analyze different aspects of the language.

2. Natural Language Generation (NLG)

- Natural Language Generation (NLG) acts as a translator that converts the computerized data into natural language representation. It
- involves Text planning, Sentence planning, and Text Realization.

Objectives of NLP

- Read the human languages in a manner that is valuable.
- Decipher the human languages
- Understand the human languages

NLP Advantages

- NLP helps users to ask questions about any subject and get a direct response within seconds.
- NLP offers exact answers to the question i.e. does not offer unnecessary and unwanted information.
- NLP helps computers to communicate with humans in their languages.
- Improves the efficiency of documentation processes, accuracy of documentation, and identify the information from large databases.

NLP Disadvantages

- May not show context.
- Unpredictable
- May require more keystrokes.
- NLP is unable to adapt to the new domain, and it has a limited function. So, NLP is built for

NLP Applications

- Language translation applications such as Google Translate
- Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.
- Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.
- Personal assistant applications such as OK Google, Siri, Cortana, and Alexa.

Build an NLP pipeline

1. Sentence Segmentation

- Sentence Segmentation breaks the paragraph into separate sentences.
  - e.g: ***Independence Day is one of the important festivals for every Indian citizen. It is celebrated on the 15th of August each year ever since India got independence from the British rule. The day celebrates independence in the true sense.***

Sentence Segmentation:

1. "Independence Day is one of the important festivals for every Indian citizen."
2. "It is celebrated on the 15th of August each year ever since India got independence from the British rule."
3. "This day celebrates independence in the true sense."

Build an NLP pipeline

2: Word Tokenization

- Word Tokenizer is used to break the sentence into linguistic units (tokens) such as words, punctuation, numbers, alphanumeric, etc.
- Tokens need not to be further decomposed for subsequent processing.
- Also, Handling Abbreviations, Hyphenated Words, Numerical and special expressions
    - **e.g.** Coursera offers Corporate Training, Online Training, and Winter Training.

Word Tokenizer output:

- " Coursera", "offers", "Corporate", "Training", "Online", "Training", "and", "Winter", "Training", "."

3. Stemming

- Stemming is used to normalize words into its base form or root form.
    - e.g. The words celebrates, celebrated and celebrating are originated with a single root word "celebrate."
- The issue with stemming is that sometimes it produces the root word which may not have any meaning.
    - E.g. intelligence, intelligent, and intelligently are originated with a single root word "intelligen."
    - In English, the word "intelligen" do not have any meaning.

Build an NLP pipeline

4. Lemmatization

- Lemmatization is grouping the different inflected forms of the word. Lemmatization is quite similar to the Stemming. The difference between Stemming and lemmatization is that lemma produces the root word, which has a meaning.
  - *e.g.* In lemmatization, the words intelligence, intelligent, and intelligently has a root word intelligent, which has a meaning.

5. Identifying Stop Words

- In English, Many words appear very frequently such as "is", "and", "the", and "a".
- NLP pipelines will flag these words as stop words. **Stop words** might be filtered out before doing any statistical analysis.
  - **e.g.** He **is a** good boy.
  - Filter the words 'is' and 'a'

6. Dependency Parsing

- Dependency Parsing finds how all the words in the sentence are related to each other

Build an NLP pipeline

7. POS tags

- POS stands for parts of speech, which includes Noun, verb, adverb, and Adjective. It indicates that how a word functions with its meaning as well as grammatically within the sentences. A word has one or more parts of speech based on the context in which it is used.
  - **e.g. "Google"** something on the Internet.
  - In the above example, Google is used as a verb, although it is a proper noun.

8. Named Entity Recognition (NER)

- Named Entity Recognition (NER) is the process in which consider a string of text (sentence or paragraph) as input and identifies relevant nouns (people, places, and organizations) that are mentioned in that string.
  - **e.g. Steve Jobs** introduced iPhone at the Macworld Conference in San Francisco, California.

9. Chunking

- Chunking or shallow parsing is used to extract the individual piece of information from unstructured text and grouping them into bigger pieces of sentences.

Natural language understanding

- Mapping the given input in natural language into useful representations
- Analyzing different aspects of the language

Raw speech signal

↓ • **Speech recognition**

Sequence of words spoken

↓ • **Syntactic analysis** using knowledge of the grammar

Structure of the sentence

↓ • **Semantic analysis** using information about meaning of words

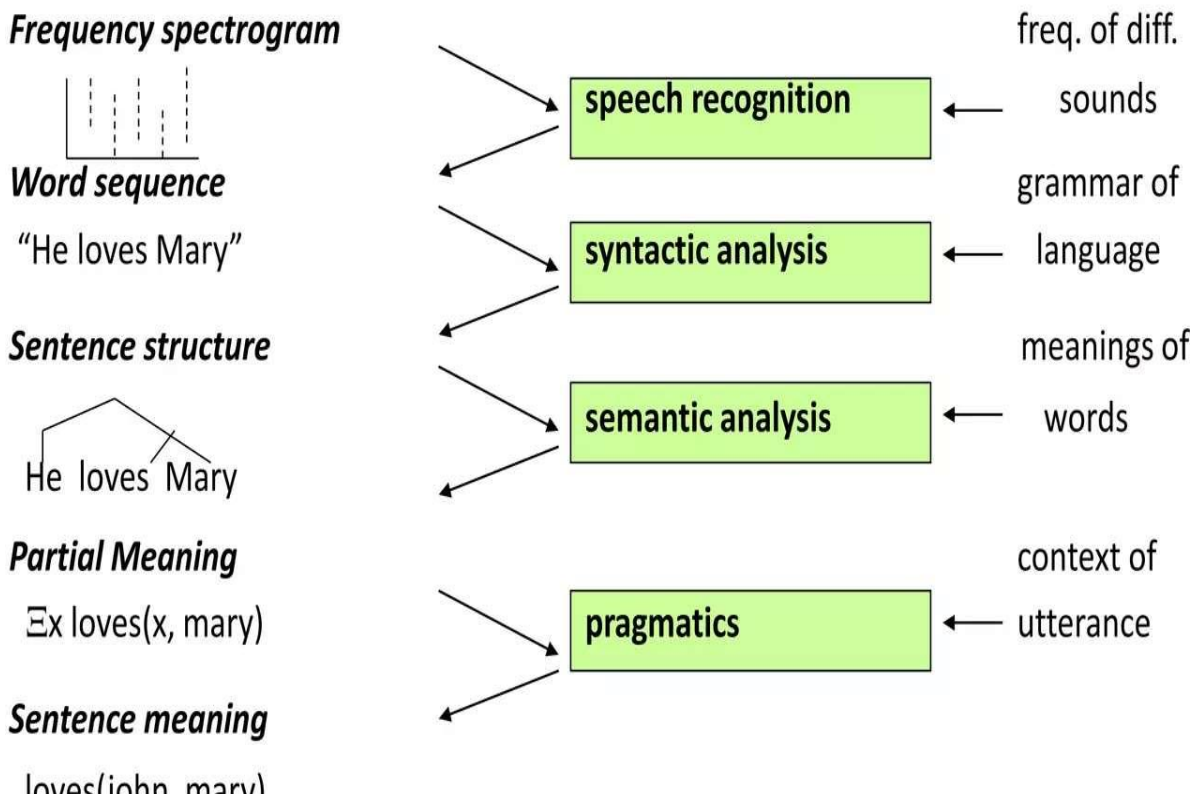
Partial representation of meaning of sentence

↓ • **Pragmatic analysis** using information about context

Final representation of meaning of sentence

Natural language understanding

- Input/Output data — Processing stage — Other data used



Phases of NLP

- Lexical (Morphological) analyzer
- Parser (syntactic analysis)
- Semantic analysis (transform into a logical form, semantic network, etc.)
- Discourse analysis (semantics beyond individual sentences)
- Pragmatic analysis (Practical usage of language: what a sentence means in practice)

Morphological (Lexical) analysis

- **Morphological analyzer** Separate words into individual morphemes (meaningful linguistic unit e.g. dog) and identify the class of the morphemes. The difficulty of this task depends greatly on the complexity of the morphology (*i.e.*, the structure of words) of the language being considered.
- It recognizes the word and category using a dictionary (word + category).
- ***Lemma*** is the task of removing inflectional (process of word formation) endings only and to return the base dictionary form of a word.
- Simply model all possible forms of a word (*e.g.*, "open, opens, opened, opening") as separate words.
- Morphological rules:

  Lemma + ed -> Lemma + ed (verb in past form)
- Checks the Lemma is in dictionary or not. If yes, the transformation is possible.
- Then, Form a set of possible lemmas

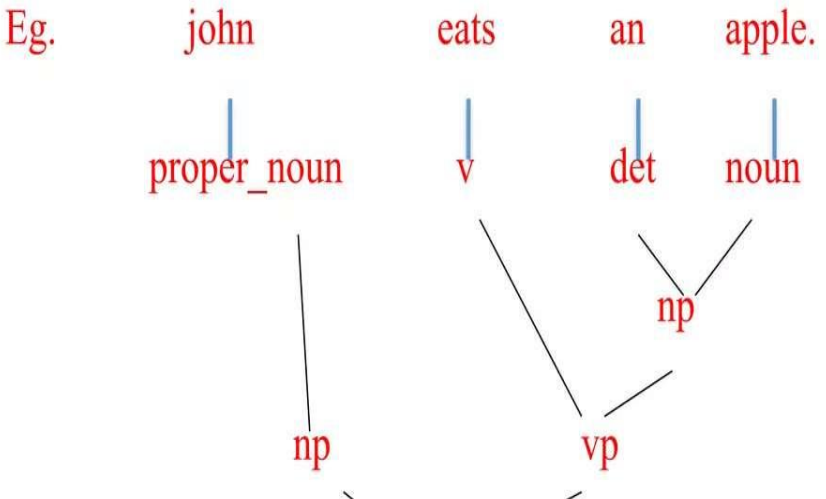
Syntactic analysis / **Parsing** (DCG)

- Assess how the natural language aligns with the grammatical rules.
- Computer algorithms are used to apply grammatical rules to a group of words and derive meaning from them
- ***Syntactic categories:*** Verb, Noun, determiner…
- ***Analysis:*** Recognize the category of an element

Semantic analysis grammar

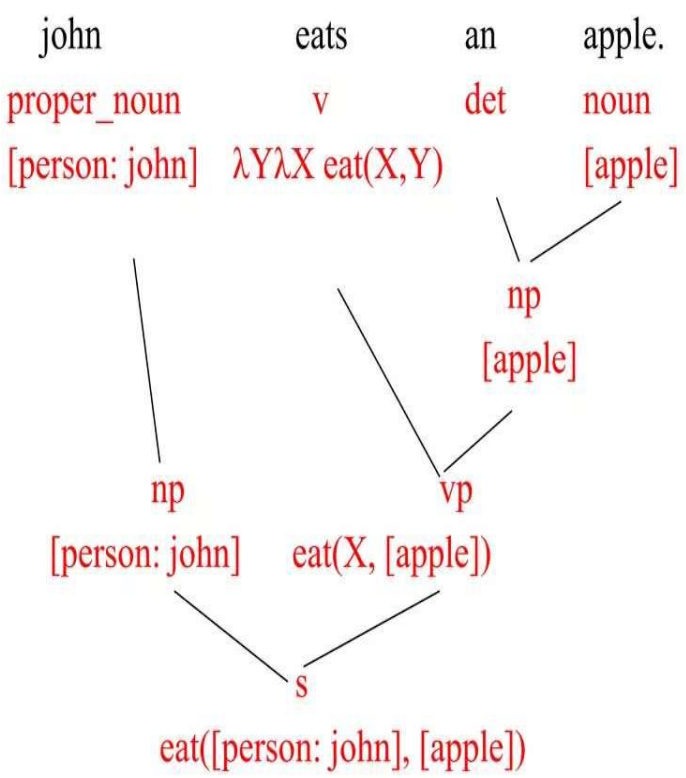
```
s --> np, vp.
np --> det, noun.
np --> proper_noun.
vp --> v, np.
vp --> v.
```

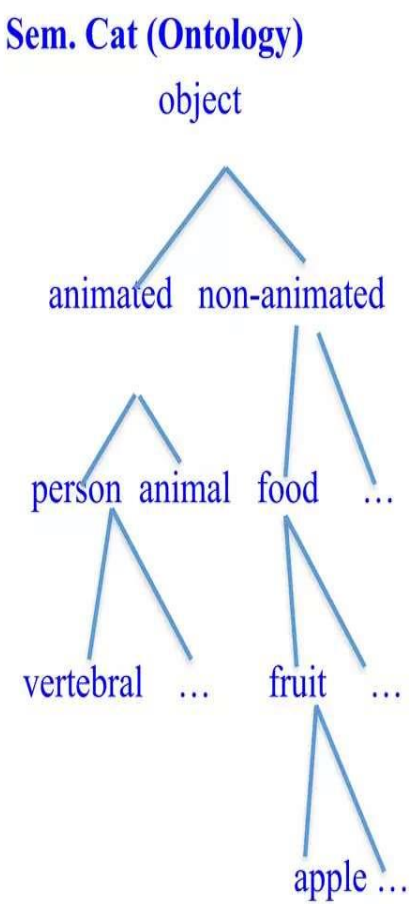
```
det -->[a].        det --> [an].
det --> [the].
noun --> [apple].
noun --> [orange].
proper_noun --> [john].
proper_noun --> [mary].
v --> [eats].
v --> [loves].
```



Semantic analysis

- Semantics refers to the meaning that is conveyed by a text.
- Apply algorithms to understand the meaning, and interpretation of words and how sentences are structured.
- ***Semantic categories:*** Person, Fruit, Apple, …
- ***Analysis:*** Check how different elements can be combined into a sentence





Discourse analysis

- A discourse is a sequence of sentences. Understanding discourse structure is extremely important for dialog system.
- Understanding a text
  - Who/when/where/what … are involved in an event?
  - How to connect the semantic representations of different sentences?
  - What is the cause of an event and what is the consequence of an action?
- Discourse level of NLP works with units of text longer than a sentence i.e. it does not interpret multi-sentence texts as just concatenated sentences, each of which can be interpreted single sentence.
- Discourse focuses on the properties of the text as a whole that convey meaning by making connections between component sentences.

Discourse analysis

- Several types of discourse processing can occur at this level like anaphora resolution and discourse/text structure recognition.
- Anaphora resolution is the replacing of words such as pronouns which are semantically vacant with the appropriate entity to which they refer.
    - He hits the car with a stone. It bounces back.

Example: The dialog may be

- When does the bus to Hyderabad leave?
- There is one at 10 a.m. and one at 1 p.m.
- Give me two tickets for the earlier one, please.

- The problems with discourse analysis may be non-sentential utterances, cross-sentential

Pragmatic analysis

- Practical usage of language: what a sentence means in real time (practice)
  - Do you have time?
  - How do you do?
  - It is too cold to go outside!
- Pragmatic analysis is concerned with the purposeful use of language in situations and utilizes context over and above the contents of the text for understanding.
- The goal is to explain how extra meaning is read into texts without actually being encoded in them. This requires much world knowledge including the understanding of intentions, plans and goals.
- Some NLP applications may utilize knowledge bases and inferencing modules.
- Speech acts in the pragmatic processing is the illocutionary force, the communicative force of an utterance, resulting from the function associated with it.
  - *Example:* Suppose the sentence is *I will see you later.*
    - *Prediction:* I predict that I will see you later.

NLP Challenges (difficulties)

- NLP is difficult because Ambiguity and Uncertainty exist in the language.

Ambiguity: Three ambiguity exists.

1. **Lexical Ambiguity**
   - Lexical Ambiguity exists in the presence of two or more possible meanings of the sentence within a **single word.** *e.g.* Manya is looking for a **match**.
   - The word match refers to that either Manya is looking for a partner or Manya is looking for a match.
2. **Syntactic Ambiguity**
   - Syntactic Ambiguity exists in the presence of two or more possible meanings within the sentence. *e.g.* I saw the girl with the binocular.
   - DID I have the binoculars? Or did the girl have the binoculars?
3. **Referential Ambiguity**
   - Referential Ambiguity exists when you are referring to something using the pronoun.

Sentiment analysis

Sentiment

- A thought, view, or attitude, especially based on emotion instead of reason.

Sentiment Analysis

- Also called opinion mining
- It uses natural language processing (NLP) and computational techniques to automate the extraction or classification of sentiment from typically unstructured text.
- Extract subjective information usually from a set of documents, often using online reviews to determine "polarity" (positive or negative or neutral opinion) about specific objects. Also focus on feelings and emotions (angry, happy, sad, etc), and even on intentions (e.g. *interested* v. *not interested*).
- It is especially useful for identifying trends of public opinion in social media, for marketing.
- *E.g* sentiment analysis analyzes 3,000+ reviews about the product automatically that could

Types of Sentiment analysis

1. Fine-grained Sentiment Analysis

- If polarity precision is important to the business, consider expanding the polarity categories and interpret 5-star ratings in a review as:
  - Very positive = 5 stars
  - Positive = 4 stars
  - Neutral = 3 stars
  - Negative = 2 stars
  - Very negative = 1 stars

2. Emotion detection sentiment analysis

- This aims at detecting emotions such as happiness, frustration, anger, sadness, etc. Many emotion detection systems use lexicons (i.e. lists of words and the emotions).
- One of the disadvantage of using lexicons is that people express emotions in different ways.
- Some words that typically express anger such as *bad* or *kill* (e.g. *product is so bad* or *customer*

Types of Sentiment analysis

3. Aspect-based Sentiment Analysis

- Product reviews based sentiment analysis requires particular aspects or features of the product i.e. review is in a positive, neutral, or negative way.
- E.g: *"The battery life of this cellphone is too short"*, an aspect-based classifier would be able to determine that the sentence expresses a negative opinion about the feature battery life.

4. Multilingual sentiment analysis

- It involves a lot of preprocessing and resources. Most of these resources are available online (e.g. sentiment lexicons), while others need to be created (e.g. translated corpora or noise detection algorithms).
- Multilingual sentiment analysis can be difficult due to complex coding.
- Alternatively, use existing classifiers to detect language in texts automatically, then train a custom

Benefits of Sentiment analysis

- Huge volumes of text data is created every day: emails, support tickets, chats, social media conversations, surveys, articles, documents, etc).
- It's time-consuming and expensive to analyze, understand, and sort.

1. **Sorting Data at Scale**
   - Huge amount of data is sorted efficiently.
2. **Real-Time Analysis**
   - immediately identify these kinds of situations, so you can take action right away.
3. **Consistent criteria**
   - Organization can apply the same criteria to all of data to improve accuracy and gain better insights.

Features and Recognition

- Features in anyone of the form
  - Words (unigrams)
  - Phrases/n-grams
  - Sentences
- Features Interpretation methods for sentiment detection
  - Bag of words (Information Retrieval)
  - Annotated lexicons (WordNet, SentiWordNet)
  - Syntactic patterns
  - Paragraph structure

Sentiment Analysis Algorithms

1. **Rule-based systems**: Perform sentiment analysis based on a set of manually crafted rules.
2. **Automatic systems:** Using machine learning techniques to learn from data.
3. **Hybrid systems:** Combine both rule-based and automatic approaches.

1. Rule-based Approaches

- It uses a set of human-crafted rules to identify subjectivity, polarity, or the subject of an opinion.
- These rules may include various techniques developed in computational linguistics such as:
  - *Stemming*, *tokenization*, *part-of-speech tagging* and *parsing*.
  - Lexicons (i.e. lists of words and expressions).

Rule-based system working principle:

1. Defines two lists of polarized words (e.g. negative words such as *bad*, *worst*, *ugly* and positive words such as *good*, *best*, *beautiful*, etc.).
2. Counts the number of positive and negative words that appear in a given text.
3. If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral sentiment.

- **Drawback:** Rule-based systems are very naive since not considers how words are combined in a sequence. Also, new rules

Sentiment Analysis Algorithms: **Automatic systems**

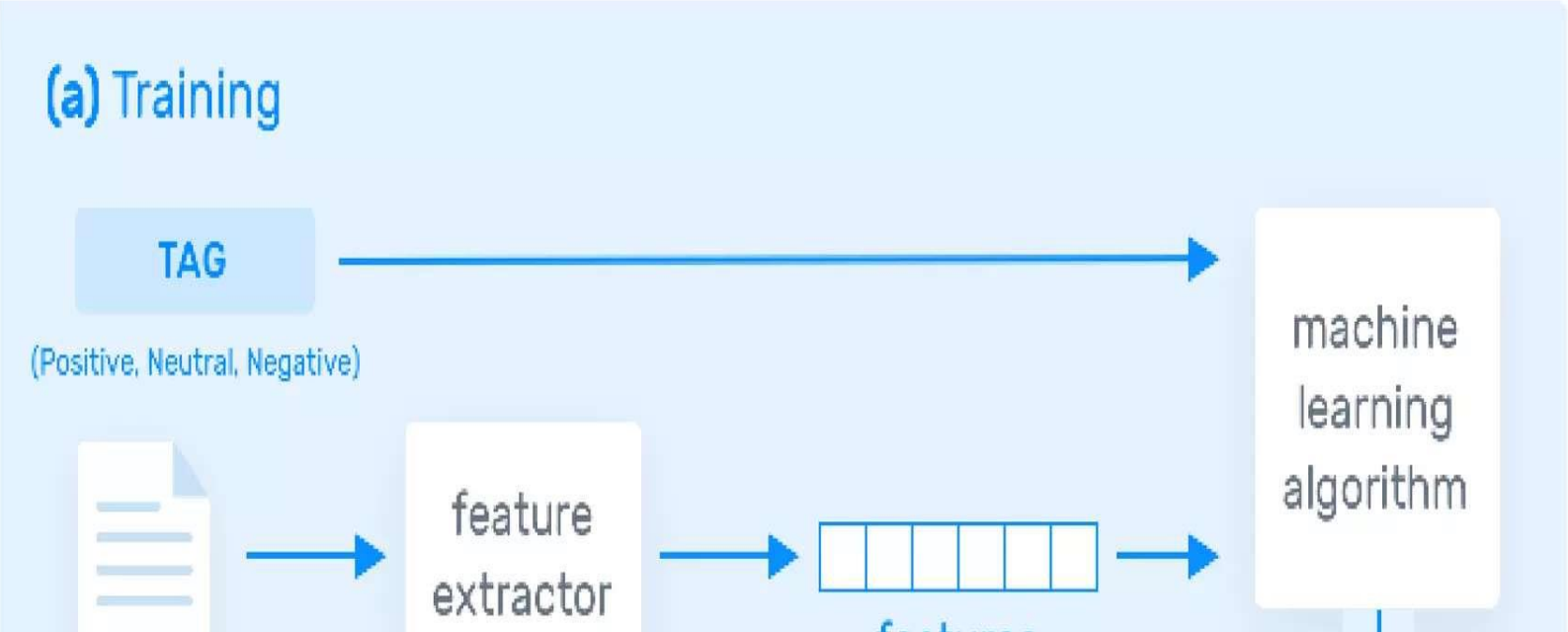
2. Automatic systems

- It does not rely on manually crafted rules
- It needs machine_learning techniques.
- A sentiment analysis task is usually modeled as a classification problem, whereby a classifier is fed a text and returns a category, e.g. positive, negative, or neutral.

Sentiment Analysis Algorithms: **Automatic systems**

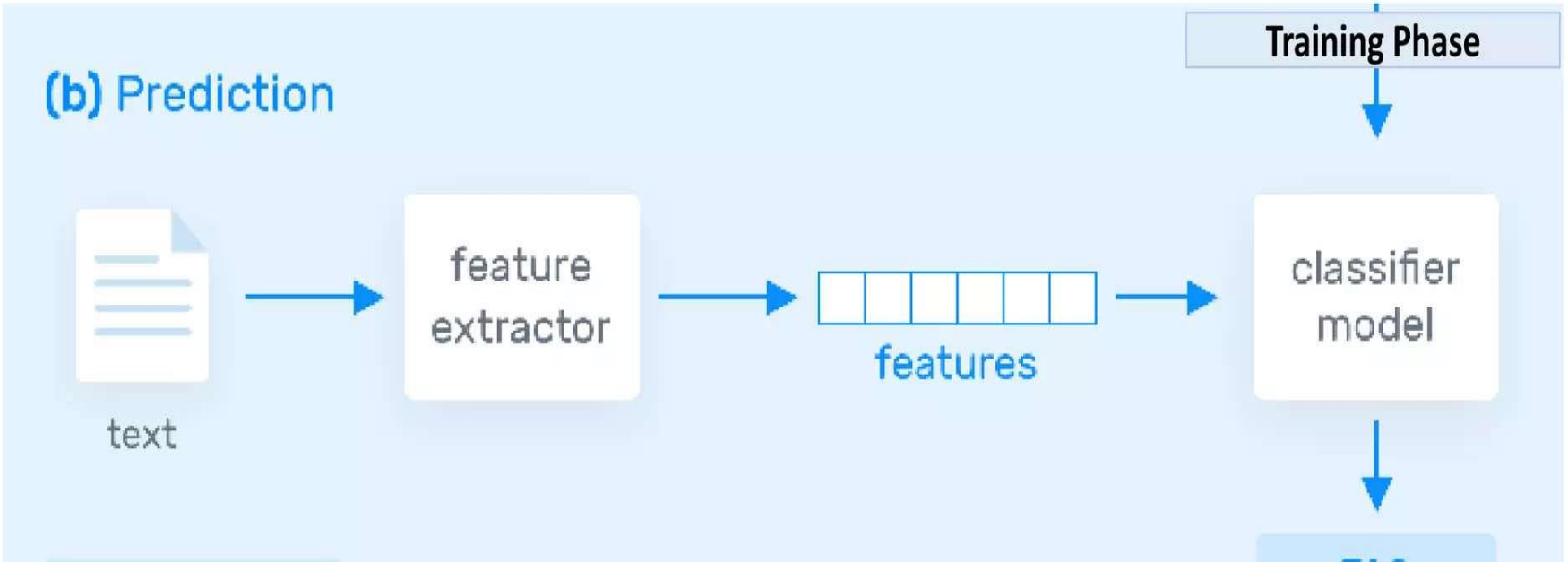
Training Processes

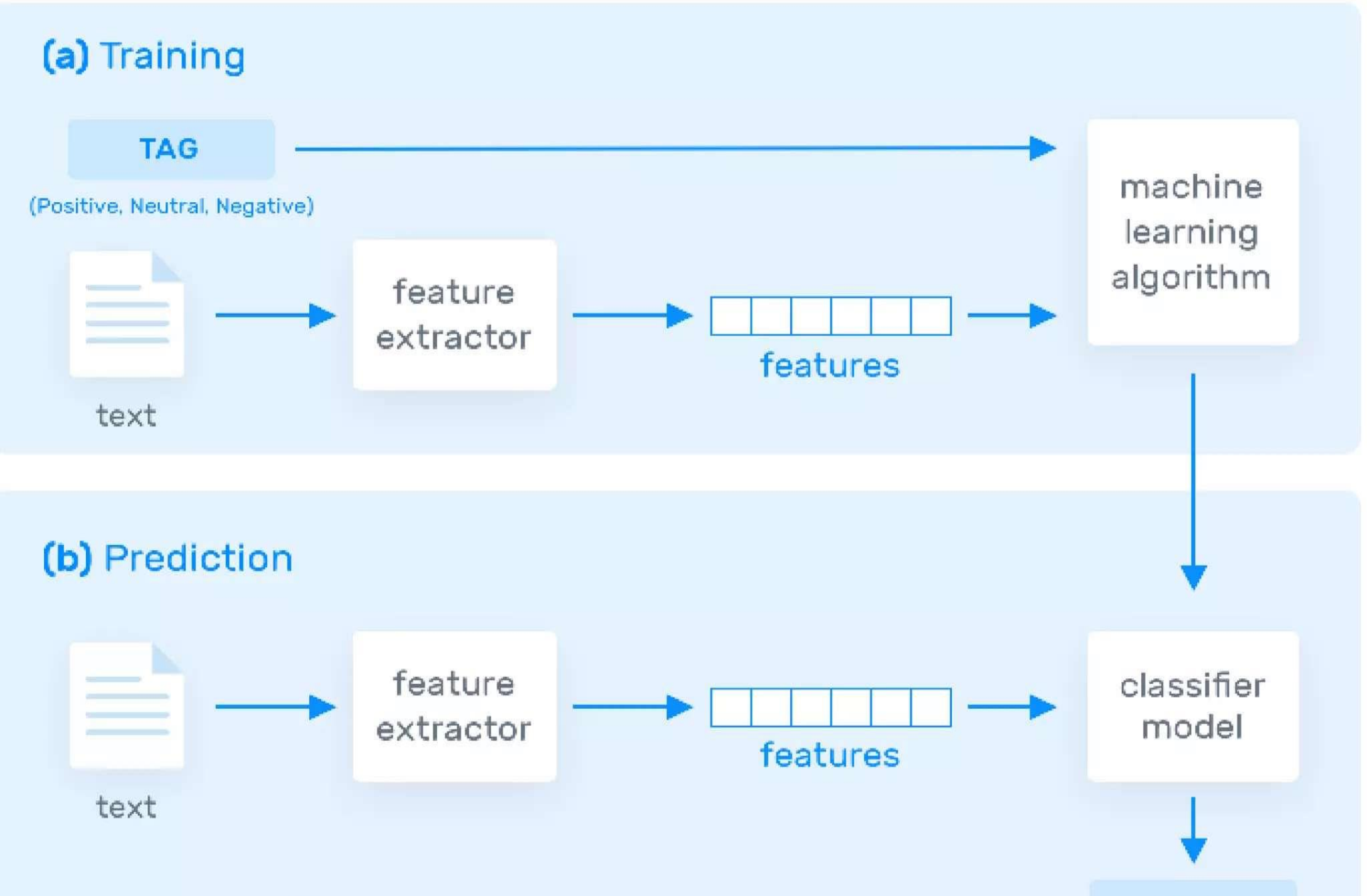
- Model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the test samples used for training.
- The feature extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g. *positive*, *negative*, or *neutral*) are fed into the machine learning algorithm to generate a model.



Sentiment Analysis Algorithms: **Automatic systems**

- In the prediction process, the feature extractor is used to transform unseen text inputs into feature vectors.
- These feature vectors are then fed into the model, which generates predicted tags (*positive*, *negative*, or *neutral*).




(a) Training
TAG
(Positive, Neutral, Negative)
machine learning algorithm
feature extractor
features
text
(b) Prediction
feature extractor
features
classifier model
text

Sentiment Analysis Algorithms: **Automatic systems**

Feature Extraction from Text

- The first step in a machine learning text classifier is to transform the text extraction or text vectorization, and the classical approach has been bag-of-words or bag-of-ngrams with their frequency.
- More recently, new feature extraction techniques have been applied based on word embeddings (*word vectors*). This kind of representations makes it possible for words with similar meaning to have a similar representation, which can improve the performance of classifiers.

Sentiment Analysis Algorithms: **Automatic systems**

Classification Algorithms

- Naïve Bayes: Probabilistic algorithm Bayes's Theorem predicts the category of a text.
- Linear Regression: Statistics algorithm predicts some value (Y) given a set of features (X).
- Support Vector Machines: Non-probabilistic model uses a representation of text samples as points in a multidimensional space. Samples of different categories (sentiments) are mapped to distinct regions within that space. Then, new texts are assigned a category based on similarities with existing texts and its corresponding regions.
- Deep Learning: Mimic the human brain using artificial neural networks to process data.

3. Hybrid Approaches

Use Cases & Applications

- Consumer information
  - Product reviews
- Marketing
  - Consumer attitudes
  - Trends
- Politics
  - Politicians want to know voters' views
  - Voters want to know politicians' stances and who else supports them
- Social
  - Find like-minded individuals or communities

Challenges

- Harder than topical classification, in which bag of words features perform well.
- Must consider other features due to…
  - Subtlety (precision) of sentiment expression
    - irony
    - expression of sentiment using neutral words
  - Domain/context dependence
    - words/phrases can mean different things in different contexts and domains
  - Effect of syntax on semantics

SentiWordNet

- Based on WordNet "synsets"
  - http://wordnet.princeton.edu/
- Ternary classifier
  - Positive, negative, and neutral scores for each synset
- Provides means of gauging sentiment for a text
- Hierarchically organized lexical database
- Contains online thesaurus + aspects of a dictionary

Category	Unique Strings
Noun	117,798
Verb	11,529

SentiWordNet: Construction

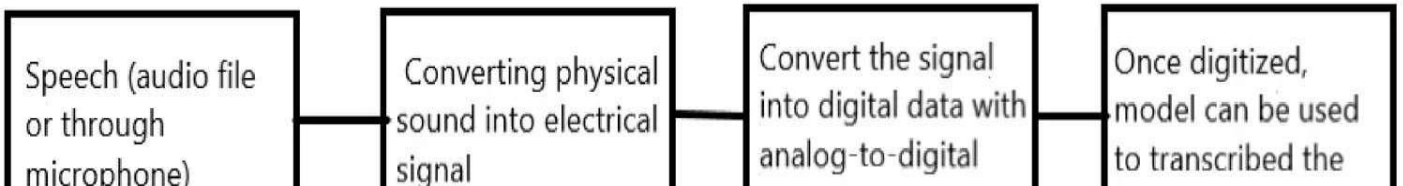
- Created training sets of synsets, L_p and L_n
  - Start with small number of synsets with fundamentally positive or negative semantics, e.g., “nice” and “nasty”
  - Use WordNet relations, e.g., direct antonymy, similarity, derived-from, to expand L_p and L_n over K iterations
  - L_o (objective) is set of synsets not in L_p or L_n
- Trained classifiers on training set
  - Rocchio and SVM
  - Use four values of K to create eight classifiers with different precision/recall characteristics
  - As K increases, P decreases and R increases

SentiWordNet: How to use it

- Use score to select features (+/-)
  - Use words in corpus with subjectivity score of 0.5 or greater
- Combine pos/neg/objective scores to calculate document-level score

Speech recognition

- Also known as Automatic Speech Recognition (ASR), Computer Speech Recognition, Speech-to-Text
- **Speech recognition** process human speech into a written format i.e. the translation of speech from a verbal format to a text.
- **Voice recognition** just identifies an individual user's voice.
- IBM released "Shoebox" in 1962. This machine recognizes 16 different words.
- IBM launches VoiceType Simply Speaking application in 1996 contains 42,000-word vocabulary, supported English and Spanish, and included a spelling dictionary of 100,000 words.



Key Features for effective Speech recognition

- **Language weighting:** Improve precision by provide weighting to specific words that are spoken frequently beyond terms already in the base vocabulary such as product names or industry jargon.
- **Speaker labeling:** Output a transcription that cites or tags each speaker's contributions to a multi-participant conversation.
- **Acoustics training:** Attend to the acoustical side of the business. Train the system to adapt to an acoustic environment (ambient noise in a call center) and speaker styles (voice pitch, volume and pace).
- **Profanity filtering:** Use filters to identify certain words or phrases and sanitize speech

Speech Recognition

- Speech recognizers are made up of a components such as the speech input, feature extraction, feature vectors, a decoder, and a word output.
- The decoder leverages acoustic models, a pronunciation dictionary, and language models to determine the appropriate output.
- Speech recognition technology is evaluated on its accuracy rate, i.e. word error rate (WER), and speed. A number of factors can impact word error rate such as pronunciation, accent, pitch, volume and background noise.
- **Natural language processing (NLP)** focuses on the interaction between humans and machines through natural language. Many mobile devices incorporate speech recognition to

Speech Recognition API

- Recognizer Class has instance that is used to recognize the speech. Each instance contains 7 methods using various APIs.
- recognize_bing() - Microsoft bing speech API
- recognize_google() - google Web speech API
- recognize_google_cloud() - Google cloud speech API
- recognize_houndify() - SoundHound API (For Music Discovery)
- recognize_sphinx() - works in offline with CMU sphinx engine
- recognize_wit() - Provided by wit.ai

Speech Recognition Applications

- **Automotive:** Speech recognizers improves driver safety by enabling voice-activated navigation systems and search capabilities in car radios.
- **Technology:** Virtual assistants are integrated within our daily lives through mobile devices. Human use voice commands to access them through smartphones such as through Google Assistant or Apple's Siri, for voice search, or through our speakers, via Amazon's Alexa or Microsoft's Cortana, to play music.
- **Healthcare:** Doctors and nurses leverage dictation applications to capture and log patient diagnoses and treatment notes.
- **Sales:** Help a call center to transcribe thousands of phone calls between customers and agents to identify common call patterns and issues. **Cognitive bots** can also talk to people via a webpage, answering common queries and solving basic requests without waiting for a contact center agent. In both cases, speech recognition systems reduce time to provide solution for consumer issues.
- **Security:** As technology integrates into our daily lives, security protocols are an increasing priority.

Text to Speech - Speech Recognition Tool

- Text-to-speech (TTS) is a type of **assistive technology** that reads digital text aloud. It's sometimes called "read aloud" technology.
- TTS take words on a computer or other digital device and convert them into audio. TTS is very helpful for kids who **struggle with reading**. But it also help kids with writing and editing.

Working method:

- TTS works with all personal digital devices, including computers, smartphones and tablets. All kinds of text files can be read aloud, including Word, pages and online web pages documents.
- The voice in TTS is computer-generated, and reading speed can usually be sped up or slowed down. Voice quality varies. There are even computer-generated voices that sound like children speaking.
- Many TTS tools highlight words while read aloud. This allows kids to see text and hear it at the same time.
- TTS based tool like Optical Character Recognition allows to read text aloud from images.
    - *E.g.* Child could take a photo of a street sign and have the words on the sign turned into audio.

Text to Speech - Speech Recognition Tool

- Speech Recognition tools and services that enable organizations to automate their complex business processes while gaining essential business insights.
- IBM Watson Text to Speech generates human-like audio from written text.
- Improve the customer experience and engagement by interacting with users in multiple languages and tones.
- Increase content accessibility for users with different abilities, provide audio options to avoid distracted driving, automate customer service interactions.
- Text to Speech service supports a variety of languages, voices, and dialects.
- It offers multiple voices, including both male and female voices.

NL in the cloud

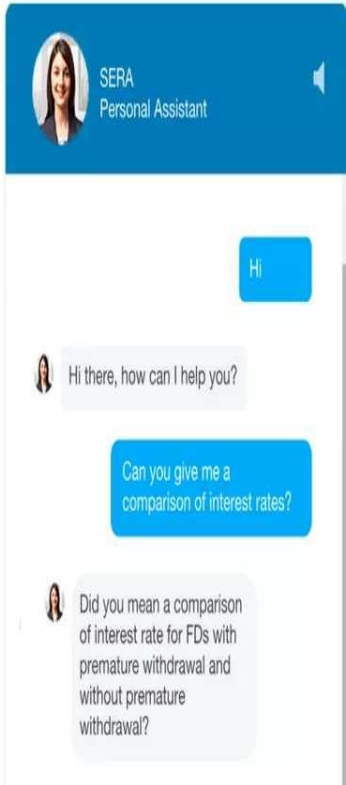
- Natural Language uses machine learning to reveal the structure and meaning of text. User can extract information about people, places, and events, and better understand social media sentiment and customer conversations.
- Natural Language enables the user to analyze text and also integrate it with the document storage on Cloud Storage.

NL Interface

- The Cloud Natural Language API provides natural language understanding technologies to developers, including sentiment analysis, entity analysis, entity sentiment analysis, content classification, and syntax analysis.
- This API is part of the larger Cloud Machine Learning API family.

Chatbot

- A chatbot is a software program for simulating intelligent conversations with human using rules or artificial intelligence.
- Users interact with the chatbot via conversational interface through written or spoken text.
- Chatbots can live in messaging platforms like Slack, Facebook Messenger and Telegram to provide services like ordering products, knowing about weather and managing finance, etc



Contextual communication

- Chatbots are conversing contextually i.e. dynamic intelligence of chatbots allow to converse with users like human converse and communicate in real-life situations.

CHATBOT TRANSFORMES USER EXPERIENCE

- User experience has always been considered to convenient and easy interactions with a product.
- When the interface connecting human and machine that becomes intelligent and conversational. It becomes easy for the user to converse with the machine like a human.
- Chatbot has demonstrated the power and convenience of conversational interfaces. As of now, language is emerging as the interface.
- According to Gartner, 30% of browsing done by users will turn out to be screen less in 2020. Chatbots are being built to handle user requests, provide relevant information and be the friend who would seek information.
- Moreover, chatbots have brought the twist to how businesses reach out to their target market

Build a Chatbot

- **Problem identification** – For which problem, the bot need to design to provide solution.
- Chatbots can be built in **two** approaches:
    1. Rule-based approach - Need hard coding (i.e. Basic Chatbots)
    2. Machine learning - Need streaming data to learn on its own. (i.e. AI based Chatbots)
- Decide the **platforms** to host the cahtbot like Facebook Messenger, Slack, Discord, Telegram and Kik.
- Choose the **Services** that can be leveraged to build a bot like Microsoft bot frameworks, Wit.ai, Api.ai, IBM's Watson.
- Chatbots must understand the intent of the user. There is a need to infuse intelligent quotient into the bot.
- Let user asking a bot to suggest the best place for eating. The chatbot must understand what the user wants, provide a proper response. In this case, provide the name of the best hotel, though the user has mentioned it as 'place'.
- NLP API can be used to impart NLP capabilities into the chatbot. Some of the sources that can be used to build

Chatbot Design Elements

Selection of design elements of a chatbot depend on the messaging platform

- **Buttons** – Makes a user to take decisions. Buttons prompt action when the user clicks it. Interactive buttons can be added to help the user for faster decisions.
- **Get Started Button** – This is an intuitive feature prompting the user to set the bot into action. A chatbot built for Facebook Messenger cannot do without this feature.
- **Cards** – Used to serve information like links, text, images and buttons as blocked containers. Blocks fall in columns when the phone is turned on its side. A user can select the relevant card.
- **Smart Reply** – Smart Reply feature helps a user to respond fast to the chatbot's query without the user attempting to type anything. This feature is useful when the chatbot is context-aware and also has user information.

Chatbot Design Elements

- **Quick Reply** – A user can make use of this feature as a response button. Tapping quick reply, a user can send his response message to the bot.
- **Persistent Menu** – A user can make use of this feature to steer his way to another portion of the bot that is not readily available.
- These are some of the popular elements for an enriching bot experience. It is ideal to pick the best feature to suit the purpose, moment and above all user needs.

Best practices for Bot development

- Chatbot to be user-friendly. Assuring conversation with the user is seamless.
- For assuring successful chatbot development, it must know the target audience and identify use case scenarios, chatbot development platform, determine how to initiate conversation and the tone of chat.
- Best practices are:
    1. Understanding potential users – Know your audience
    2. Read user sentiment and make it emotionally rich
    3. Welcome the user for conversation
    4. Guide the user
    5. Reduce user struggle
    6. Listen to the user's voice

Best practices for Bot development

1.Understanding potential users – Know your audience

- Bot building experience starts with the understanding of the target audience and the purpose of the bot.
- Points to score
  - Know what you are building the bot for, it could be a bot to entertain audience, allow users to buy and sell, provide news or serve as a customer service channel
  - Learn about the customer product and know how the bot fits in to make it more customer-friendly.

2. Read user sentiment and make it emotionally rich

- The conversation should focus more about emotion. Read and understand user sentiments to promote content blocks that can prompt the user to continue with his conversation and end up having a rich experience.
- The chatbot should meet user's expectations. It is important to visualize the bot that shapes the conversation with the user. It should make people comfortable.
- The chatbot should not be emotionless and automated, but to be heartfelt and friendly as a human.
  - Leverage positive sentiments for promoting your product or turn users into brand ambassadors
  - Address negative comments to stay afloat in this conversation game
  - Make use of friendly language to make users feel that they are chatting with a familiar person

Best practices for Bot development

3. Welcome the user for conversation

- Though chatbots are growing in prominence, users are not that experienced to enter into a conversation with an object.
- This leads to extend a warm welcome to the user and make comfortable with the conversational interface.
    - Use familiar buttons (submit, register, etc) that comfortable for the user to enter into a conversation with the bot
    - Encourage the user to respond by typing, enabled through actionable command

4. Guide the user

- Guide the user to keep him engaged. In the case of a website or an app, users have the limitations of these platforms – like the buttons used for pressing or the page length.
- Also, it is essential to set limits and create paths for the user to make comfortable during the conversation with the chatbot.
    - Guide user conversation through actionable statements and buttons

Best practices for Bot development

5. Reduce user struggle

- Designing the chatbot with buttons and a template can keep a check on the bot breaking quite often.
- Users can get into limitless conversion.
    - Make users feel that you understand their pain
    - Help them land in safe areas of the system, offer human help when the conversation stalls.

6. Listen to the user's voice

- The chatbot cannot facilitate everything user wants or even anticipate everything they want with designed bot.
- Instead of trying to play the guessing game, include natural ways and allowing the user to give feedback during the interaction.
    - Prompt users to provide suggestions and ideas when they don't find features that they are looking for.
    - Allow users to submit articles when the chatbot fails to respond to queries or requests submitted by them.

Virtual Assistants

- Digital assistant or an intelligent virtual assistant (IVA) is an engineered entity residing in software that interacts with humans. It incorporates elements of **Interactive Voice Response** and artificial intelligence that projects to virtually converse with users.
- E.g. Google Assistant(contains Google Lens) and Samsung Bixby(image processing to recognize objects in the image)
- Activate a virtual assistant using the voice, a wake word. This is a word or groups of words such as "Hey Siri", "OK Google" or "Hey Google", "Alexa", and "Hey Microsoft".

Virtual Assistants

- When virtual assistant interacts with users, AI programming uses algorithms to learn from data input and predicting the user's needs and understands natural language voice commands to complete tasks for the user.
- Virtual assistants function to provide more capabilities and value to customers.
- In **Speech recognition** and natural language processing, virtual assistant's able to understand and perform requests.
- In **Voice Recognition** technology, virtual assistants uses in business management.

Bag 0f words (BOW)

- **Bag of Words (BoW)** is a **simplified representation of text** used in Natural Language Processing (NLP).
- In BoW, a **text (sentence or document)** is represented as a **"bag" of its words**, **ignoring grammar and word order**, but **keeping multiplicity** (how many times each word appears).
- Essentially, it converts text into **numerical features** that machine learning models can process.
- Order is ignored

  Example: "I love NLP" → "NLP love I" → same representation.
- **Focus on word frequency**

•Each word becomes a **feature** (column) in a vector.

Output

Typically, a vector with **counts** of each word in the vocabulary.

3 lines:

1: "I love NLP"

2: "I love coding"

3: "NLP is fun»

- Step-01:Unique words:
- ["I", "love", "NLP", "coding", "is", "fun"]
- **Step 2: Represent each sentence as a vector (word counts)**

Sentence	I	love	NLP	coding	is	fun
"I love NLP"	1	1	1	0	0	0
"I love coding"	1	1	0	1	0	0
"NLP is fun"	0	0	1	0	1	1

TF-IDF (Term Frequency-Inverse Document Frequency)

- TF-IDF (Term Frequency–Inverse Document Frequency) is a statistical method used in natural language processing and information retrieval to evaluate how important a word is to a document in relation to a larger collection of documents. TF-IDF combines two components:
- **1. Term Frequency (TF):** Measures how often a word appears in a document. A higher frequency suggests greater importance. If a term appears frequently in a document, it is likely relevant to the document's content.

TF(t, d) = (Number of times term t appears in document d) / (Total number of terms in document d)

- **2. Inverse Document Frequency (IDF):** Reduces the weight of common words across multiple documents while increasing the weight of rare words. If a term appears in fewer documents, it is more likely to be meaningful and specific.

IDF(t) = log (N / DF(t))

TF-IDF (Term Frequency-Inverse Document Frequency) (t, d, D) =

TF(t, d) * IDF(t)