Unit-5

Dr.P.Suthanthiradevi

Topics

- Introduction to Chatbot Applications
- Retrieval based- Conversation based
- Information Extraction and its approaches
- Information Retrieval
- Semantic Search and Evaluation
- Summarization
- Extractive Vs Abstractive, Summarization
- Single and Multi-document Summarization – Question Answering
- Machine Translation

Introduction to Chatbot Applications

Introductions to Chatbot Application

- This topics is about a conversation with AI today.
- To have a conversation with your AI, need a few pre-trained tools which can help you build an AI chatbot system.
- This topic guides you to combine speech recognition processes with an artificial intelligence algorithm.
- Natural Language Processing or NLP is a prerequisite for this project.
- NLP allows computers and algorithms to understand human interactions via various languages.
- In order to process a large amount of natural language data, an AI will definitely need NLP or Natural Language Processing.

What is Chatbot?

- A chatbot (conversational interface, AI agent) is a computer program that can understand human language and converse with a user via a website or a messaging app.
- Chatbots can handle various tasks online — from answering simple questions and scheduling calls to gathering customer feedback.
- Brands use bots to automate their business processes, speed up customer service, and lower support costs.

Chatbot Introductions

- 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 and serve many purposes – ordering products, knowing about weather and managing your finance among other things.

Chatbots Vs Bots

- A chatbot is a conversational interface that communicates with users via text or voice.
- On the other hand, the bot is a type of software that automates repetitive tasks like searching for information and adding it to sites, but it doesn’t chat with users.

Types of Chatbots

- ***Text-based chatbot***: In a text-based chatbot, a bot answers the user’s questions via a text interface.
- ***Voice-based chatbot***: In a voice or speech-based chatbot, a bot answers the user’s questions via a human voice interface.
- **Traditional chatbots**: They are driven by system and automation, mainly through scripts with minimal functionality and the ability to maintain only system context.
- **Current chatbot:** They are driven by back-and-forth communication between the system and humans. They have the ability to maintain both system and task contexts.
- **Future chatbot:** They can communicate at multiple levels with automation at the system level. They have the ability to maintain the system, task, and people contexts. There is a possibility of introducing of master bots and eventually a bot OS.

Approaches used to design the chatbots

- In a **Rule-based** approach, a bot answers questions based on some rules on which it is trained on.
- The rules defined can be very simple to very complex.
- The bots can handle simple queries but fail to manage complex ones.
- **Self-learning** bots are the ones that use some Machine Learning-based approaches and are definitely more efficient than rule-based bots.
- These bots can be further classified into two types: Retrieval Based or Generative.

Approaches used to design the chatbots

1. ***Retrieval-Based Bots:*** Retrieval-based bots retrieve pre-existing responses from a set of predefined answers or a knowledge base. They select the most appropriate response based on the input query. This approach is useful for fact-based or frequently asked questions where answers are readily available in the training data or knowledge base.

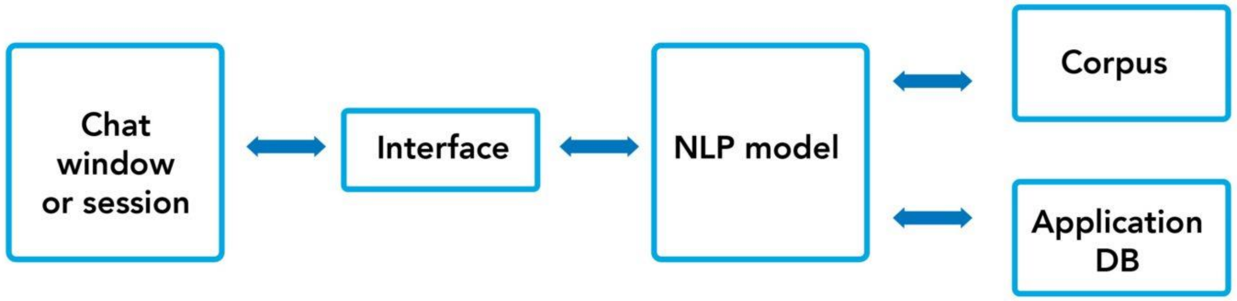
2. ***Generative Bots:*** Generative bots, in contrast, are capable of generating responses on the fly. They use machine learning models, often based on deep learning techniques, to generate text that is contextually relevant to the input query. This approach is more suitable for tasks that require creative or context-specific answers and is particularly useful for tasks like chatbots and content generation.

Applications of Chatbots

- Virtual reception assistant
- Virtual help desk assistant
- Virtual tutor or teacher
- Virtual driving assistant
- Virtual email, complaints, or content distributor
- Virtual home assistant [example: Google Home]
- Virtual operations assistant [example: Jarvis from the movie Iron Maiden]
- Virtual entertainment assistant [example: Amazon Alexa]
- Virtual phone assistant [example: Apple Siri]
- Assist the visually impaired person in describing the surroundings
- Can help a warehouse executive in locating the stocked product

Architecture of chatbots

- Typical chatbot architecture should consist of the following:
- Chat window/session/front end application interface
- The deep learning model for Natural Language Processing [NLP]
- Corpus or training data for training the NLP model
- Application Database for processing actions to be performed by the chatbot.



What is NLP Chatbot?

- An NLP chatbot is a conversational agent that uses natural language processing to understand and respond to human language inputs.
- It uses machine learning algorithms to analyze text or speech and generate responses in a way that mimics human conversation.
- NLP chatbots can be designed to perform a variety of tasks and are becoming popular in industries such as healthcare and finance.

How do you make a chat bot on NLP?

- To create an NLP chatbot,
  - define its scope and capabilities,
  - collect and preprocess a dataset,
  - train an NLP model,
  - integrate it with a messaging platform,
  - develop a user interface, and
  - test and refine the chatbot based on feedback.
- Tools such as Dialogflow, IBM Watson Assistant, and Microsoft Bot Framework offer pre-built models and integrations to facilitate development and deployment.

Simple Text-based Chatbot using NLTK with Python

- ***Algorithm for this text-based chatbot***
  - Input the corpus
  - Design NLTK responses and converse-based chat utility as a function to interact with the user.
  - Run the chat utility function.

Chatbot window

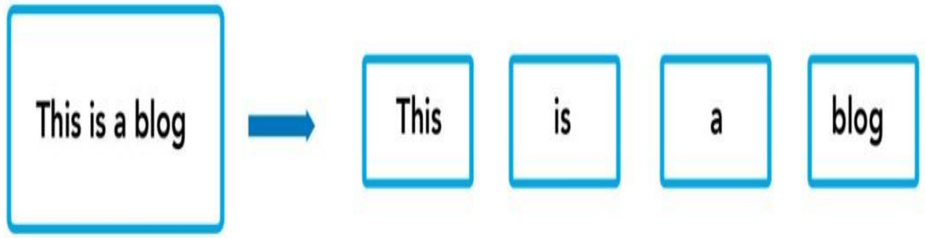
- The function keeps the chat window alive unless it is asked to break or quit.
- The name of our text bot is Jason.
- The algorithm for this function is as follows:
- The text bot introduces itself to the user.
- Chatbot asks the user to type in the chat window using the NLTK converse function.
- Bot understands what the user has typed in the chat utility window using NLTK chat pairs and reflections function.

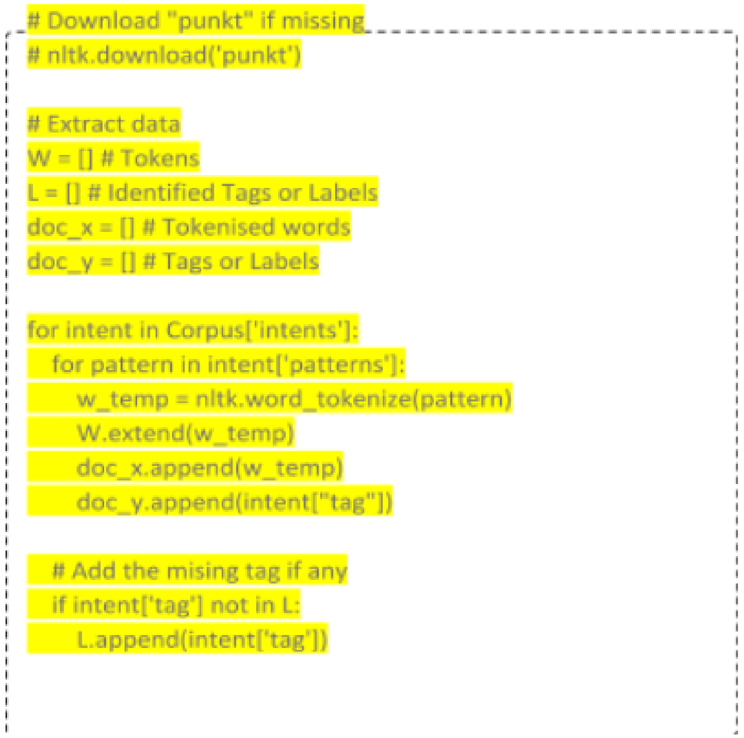
Data pre-processing
Text case [upper or lower] handling

- Convert all the data coming as an input [corpus or user inputs] to either upper or lower case.
- This will avoid misrepresentation and misinterpretation of words if spelled under lower or upper cases.

Tokenization

- Convert a sentence [i.e., a collection of words] into single words.

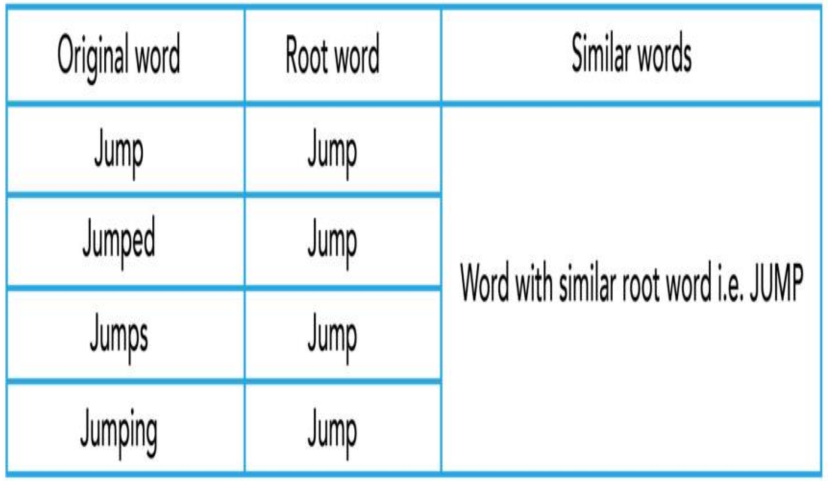




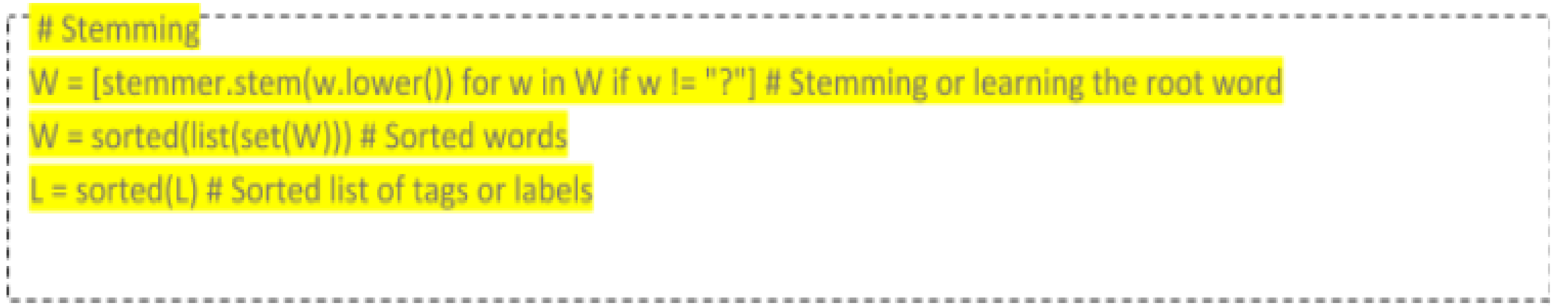
```
# Download "punkt" if missing
# nltk.download('punkt')

# Extract data
W = [] # Tokens
L = [] # Identified Tags or Labels
doc_x = [] # Tokenised words
doc_y = [] # Tags or Labels

for intent in Corpus['intents']:
  for pattern in intent['patterns']:
    w_temp = nltk.word_tokenize(pattern)
    W.extend(w_temp)
    doc_x.append(w_temp)
    doc_y.append(intent["tag"])

  # Add the mising tag if any
  if intent['tag'] not in L:
    L.append(intent['tag'])
```

Stemming

- It is a process of finding similarities between words with the same root words.
- This will help us to reduce the bag of words by associating similar words with their corresponding root words.



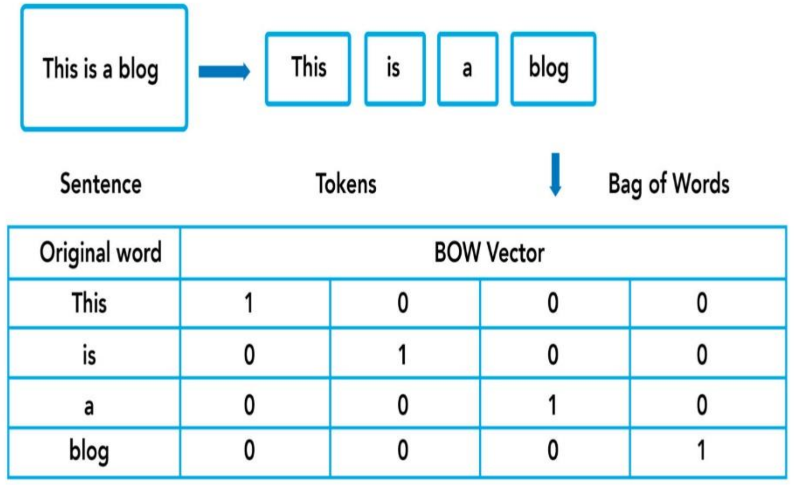
Original word	Root word	Similar words
Jump	Jump	Word with similar root word i.e. JUMP
Jumped	Jump	
Jumps	Jump	
Jumping	Jump	



```
# Stemming
W = [stemmer.stem(w.lower()) for w in W if w != "?"] # Stemming or learning the root word
W = sorted(list(set(W))) # Sorted words
L = sorted(L) # Sorted list of tags or labels
```

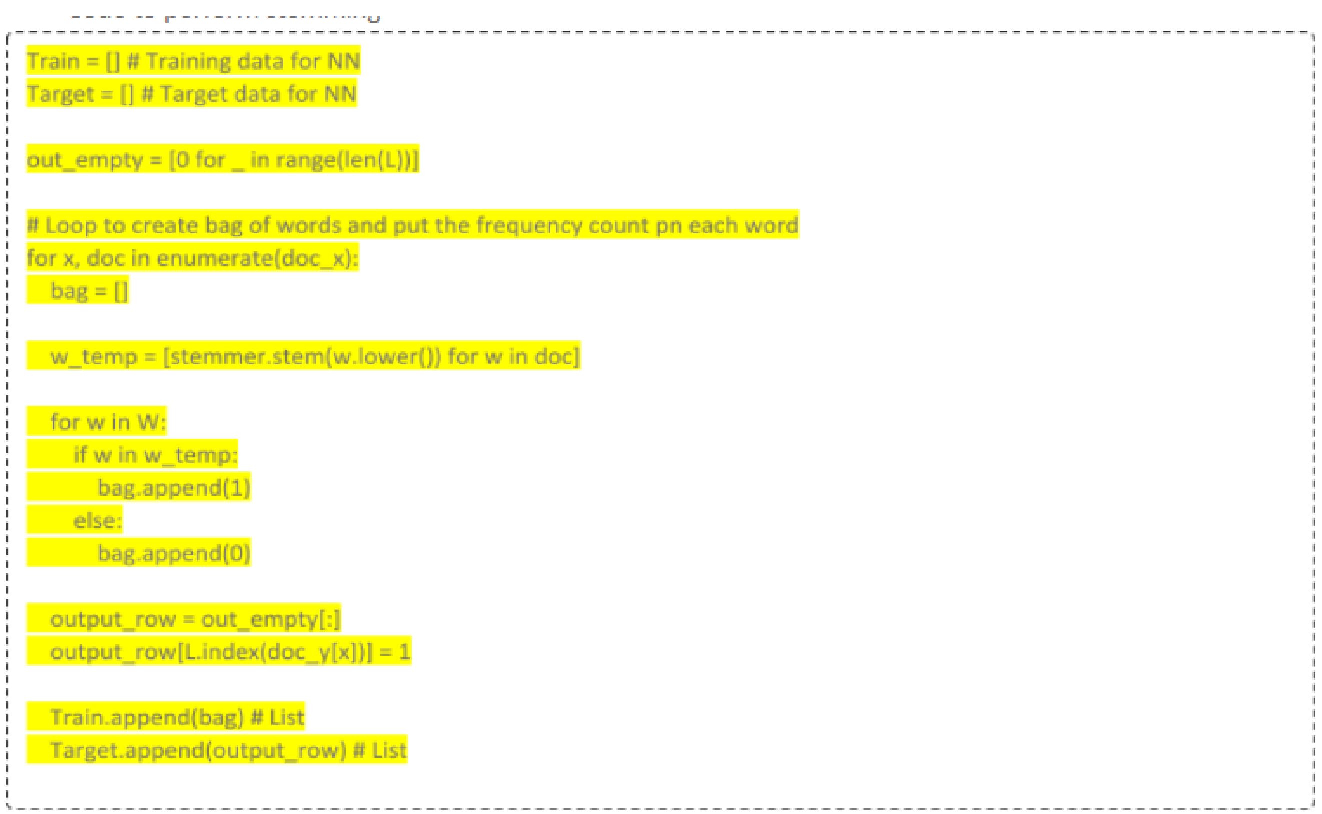
Generate BOW [Bag of Words]

- Process of converting words into numbers by generating vector embeddings from the tokens generated above.
- This is given as input to the neural network model for understanding the written text.



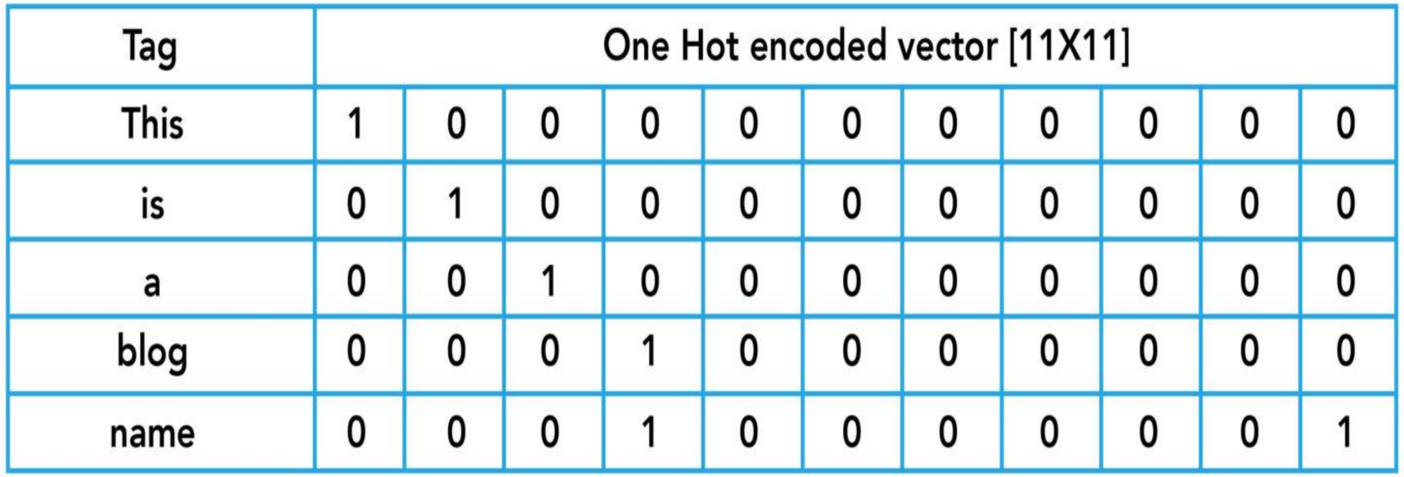
Original word	BOW Vector			
This	1	0	0	0
is	0	1	0	0
a	0	0	1	0
blog	0	0	0	1

Generate BOW [Bag of Words]



```
Train = [] # Training data for NN
Target = [] # Target data for NN

out_empty = [0 for _ in range(len(L))]

# Loop to create bag of words and put the frequency count pn each word
for x, doc in enumerate(doc_x):
    bag = []

    w_temp = [stemmer.stem(w.lower()) for w in doc]

    for w in W:
        if w in w_temp:
            bag.append(1)
        else:
            bag.append(0)

    output_row = out_empty[:]
    output_row[L.index(doc_y[x])] = 1

    Train.append(bag) # List
    Target.append(output_row) # List
```

One hot encoded tag



Tag	One Hot encoded vector [11X11]										
This	1	0	0	0	0	0	0	0	0	0	0
is	0	1	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	0
blog	0	0	0	1	0	0	0	0	0	0	0
name	0	0	0	1	0	0	0	0	0	0	1

Text classification

- Design a classifier model which can be trained on the corpus with respect to the target variable, i.e., the Tag from the corpus.
- There is a list of classifiers that can be used for this purpose which are as follows:
    - Multinomial Naïve Bayes
    - Support Vector Machines [SVM]
    - Neural network classifier

Chatbot platforms

•***Botsify***— User-friendly drag-and-drop templates to create bots.

• Easy integration to external plugins and various AI and ML features help improve conversation quality and analytics.

•***Flow XO***— This platform has more than 100+ integrations and the easiest-to-use visual editor. But, it is quite limited when it comes to AI functionality.

•***Beep Boop***— Easiest and best platform to create slack bots. Provides an end-to-end developer experience.

•***Bottr***— There is an option to add data from Medium, Wikipedia, or WordPress for better coverage. This platform gives an option to embed a bot on the website.

•***Microsoft Bot Framework***— Developers can kick off with various templates such as basic language understanding, Q&As, forms, and more proactive bots.

•***Wit.AI (Facebook Bot Engine)***— This framework provides an open natural language platform to build devices or applications that one can talk to or text.

•***API.AI (Google Dialogflow)***— This framework also provides AI-powered text and voice-based interaction interfaces. It can connect with users on Google Assistant, Amazon Alexa, Facebook Messenger, etc.

Student Evaluation **QB1) How does a chatbot works step by step?**

- ***Ans:*** Chatbots take three simple actions: understanding, acting on it, and answering. The chatbot analyzes the user’s message in the first phase. Then, after interpreting what the user stated, it takes action in accordance with a set of algorithms. Finally, it chooses one of several suitable answers.

Student Evaluation **QB2) Is Alexa a chatbot?**

- ***Ideally, Alexa is a chatbot.*** Amazon recently unveiled a new feature for iOS that allows users to make requests for Alexa and view responses on display.

Student Evaluation **QB3) Which algorithm is best for a chatbot?**

- Algorithms used by traditional chatbots are decision trees, recurrent neural networks, natural language processing (NLP), and Naive Bayes.

Reference

- https://www.mygreatlearning.com/blog/basics-of-building-an-artificial-intelligence-chatbot/#simple-text-based-chatbot-using-nltk-with-python

Retrieval based-Conversation based

Conversational - AI

- Conversational AI can be simply defined as human-
- computer interaction through natural conversations.
- This may be through a chatbot on a website or any social messaging app, a voice assistant or any other interactive messaging-enabled interfaces.
- This system will allow people to ask queries, get opinions or recommendations, execute needed transactions, find support or otherwise achieve a goal through conversations.
- Chatbots are basically online human-computer dialog system with natural language

Approaches of chatbot AI

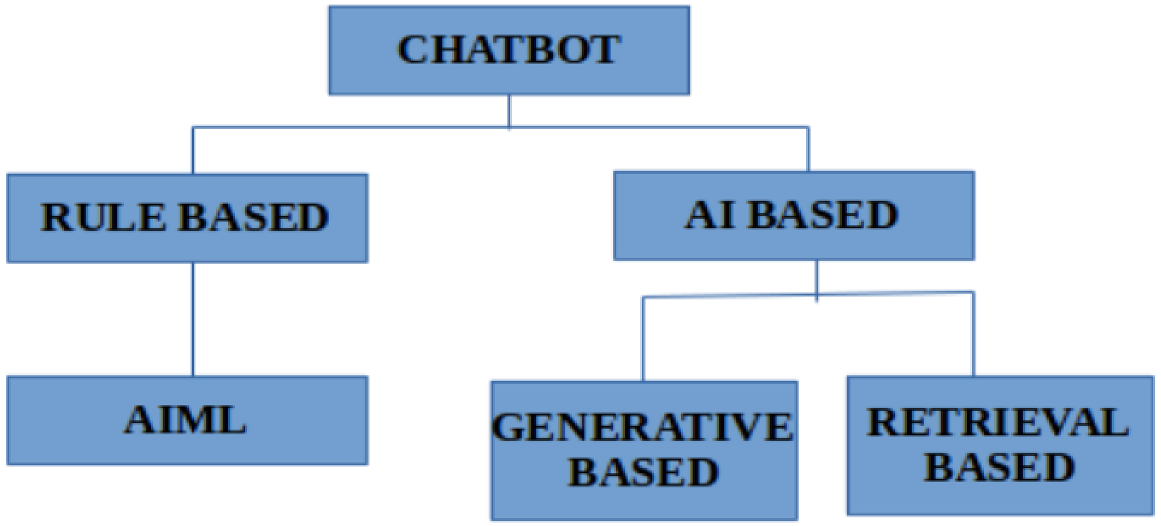


Fig. 1 Overview of chatbot

RULE-BASED SYSTEM

- Rule based system works on the basis of a set of defined rules.
- The system usually trained in such a way that, if a certain input had given by the user, the response should be a particular one.
- Actually here a transformation is taking place from input to output based on defined set of rules.
- However this rule based system are not able to produce output when a query which are not defined in the rule set is asked.

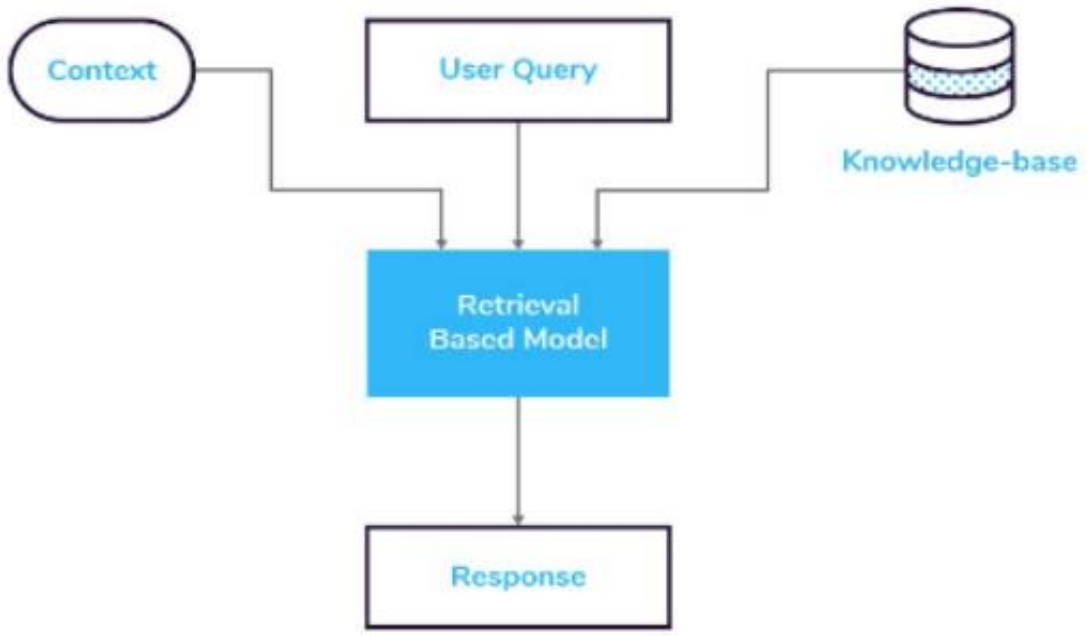
AI BASED SYSTEM

- Artificial Intelligence Based System generally uses the concept of machine learning approaches to produce responses for the input which is given by the user.
- As this system mainly depends on ML approaches, It is termed as more efficient than that of Rule Based Systems.
- Basically AI based system can be subdivided in to two types. Namely Retrieval based model and Generative based model.

Retrieval based system

- Maintain a database of predefined responses.
- It is some how similar to the rule based system, but more efficient than that of rule based.
- Here when user ask a query, the system actually selects the best response by means of simple algorithms like keywords matching or it may require also more complex processing with machine learning or deep learning.
- Basically this system need a-lot of data pre-processing.
- The databases should be manually updated at a particular interval to avoid the problem of outdation.
- The retrieval-based bots usually come up with a set of written responses to minimize the grammatical errors and to improve coherence and avoid a situation a system can be hacked to respond in less appropriate ways.
- Basically this type of chatbots are very suitable when we are dealing with domain specific conversational system

Retrieval based system



Generative based system

- It is trained in such a way that it can also produce response if a new query arrived.
- While the other two methods produces output on the basis of the rule defined or by considering matching from database, generative model needs large amount of data to get trained and from that the system try to generate new responses for a new query.
- When we are using generative model, there is high chance of producing grammatically error sentences as it have high freedom in generating responses.
- Generative system mainlyuse the concept of supervised learning, reinforcement learning,and adversarial learning for model building.

CONVERSATIONA AI - RETRIEVAL BASED MODEL

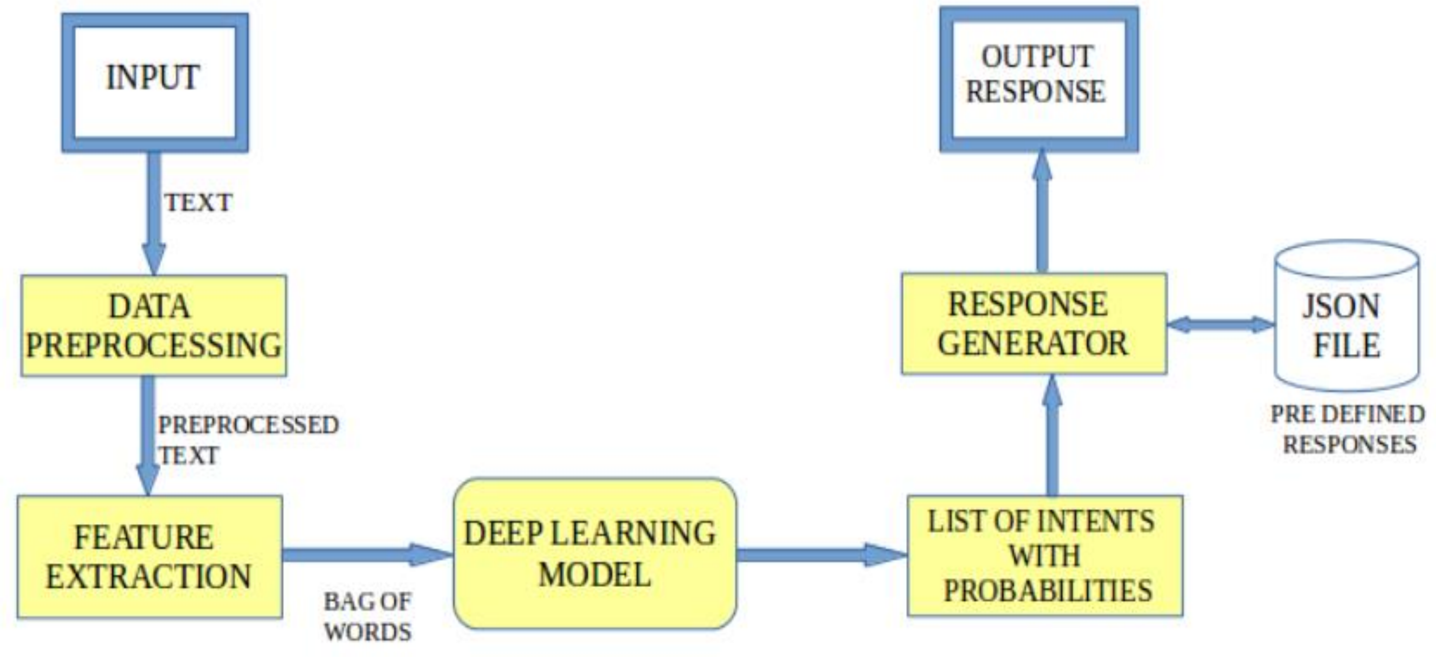
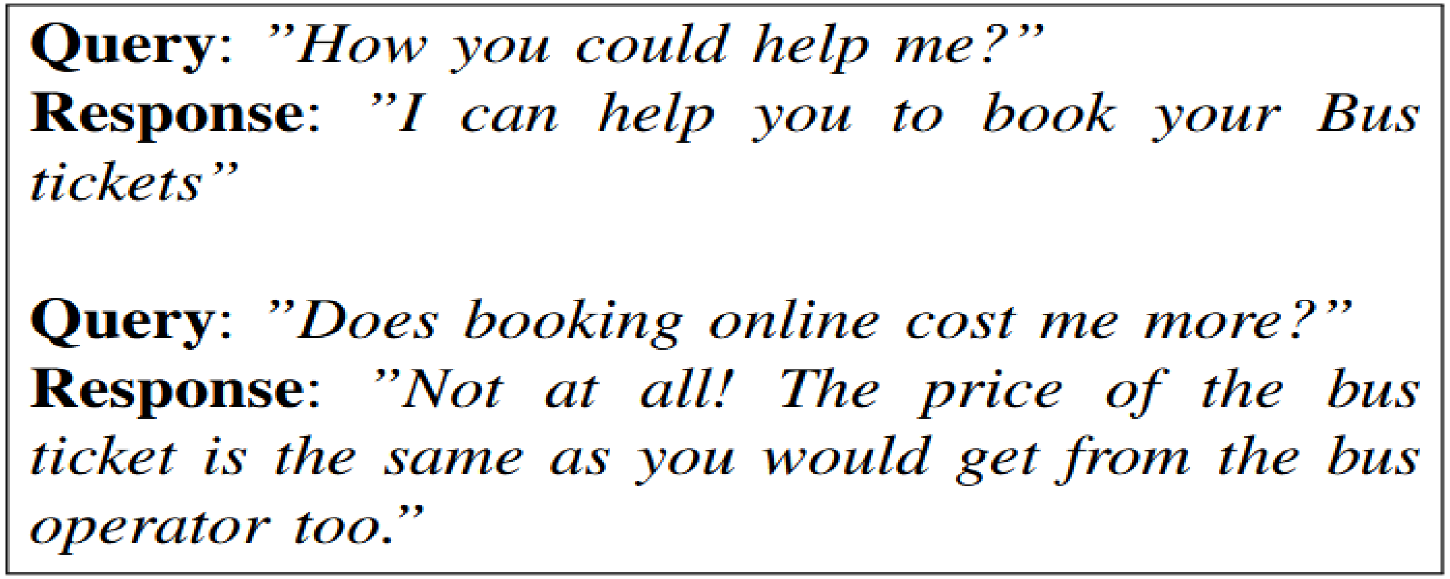


Fig. 3. System Overview

Sample Query and answers



Query: *"How you could help me?"*
Response: *"I can help you to book your Bus tickets"*

Query: *"Does booking online cost me more?"*
Response: *"Not at all! The price of the bus ticket is the same as you would get from the bus operator too."*

REFERENCE

- https://solozano0725.medium.com/retrieval-based-chatbots-using-nltk-keras-e4f86b262b17

Information Extraction and its approaches

Information Extraction (IE)

Information Extraction (IE) is an essential component of NLP for several reasons:

1. ***Structured Data from Unstructured Text:*** Much of the world's information is stored in unstructured text form, such as articles, websites, documents, and social media posts. ***IE enables the conversion of this unstructured text into structured data, making it usable for various applications***.
2. ***Data Organization and Summarization:*** IE techniques can ***extract key information from large volumes of text***, allowing for the organization and summarization of content. This is valuable for tasks like ***creating document summaries, generating metadata, and organizing data for analysis.***
3. ***Automating Repetitive Tasks:*** Many NLP tasks involve manual and repetitive work, such as cataloging information, tagging entities, or extracting facts from documents. IE automates these tasks, saving time and reducing human error.

Information Extraction (IE)

Information Extraction (IE) is an essential component of NLP for several reasons:

4. ***Search and Retrieval:*** Structured information can be indexed more efficiently than unstructured text. This improves the retrieval speed and relevance of search results since the search engine can access and process the indexed information quickly.

5. ***Entity Recognition and Linking:*** IE is used to identify and link entities (such as names of people, places, organizations, etc.) in text. This is crucial for tasks like entity resolution, disambiguation, and entity-based information retrieval.

Information Extraction (IE)

- Working with an enormous amount of text data is always hectic and time-consuming.
- Hence, many companies and organisations rely on **Information Extraction techniques** to automate manual work with intelligent algorithms.
- Information extraction can reduce human effort, reduce expenses, and make the process less error-prone and more efficient.

- https://nanonets.com/blog/information-extraction/

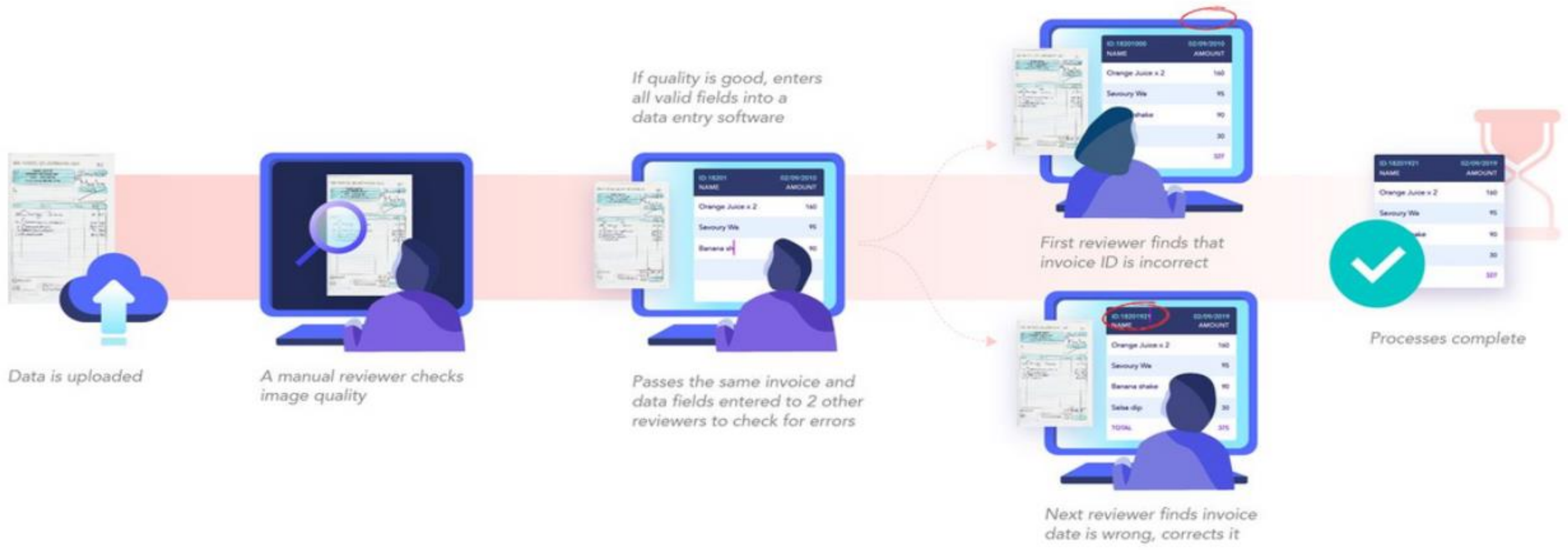
Information Extraction

- ***What is information extraction?***
  - Information Extraction is the ***process of parsing through unstructured data and extracting essential information into more editable and structured data formats.***
  - It is the task of automatically ***extracting structured information*** from unstructured and/or semi-structured machine-readable documents and other electronically represented sources.

Information Extraction

- ***What is information extraction?***
  - IE involves identifying specific pieces of information, often referred to as ***"facts" or "entities,"*** within a document or a corpus of text. These facts are typically predefined and could include entities like names of people, organizations, locations, dates, numerical values, or specific events.
  - ***Named Entity Recognition (NER) and relation extraction*** are common techniques used in IE. NER identifies and classifies entities in the text, while relation extraction discovers relationships between these entities.
  - **Use Cases:** IE is commonly used in applications such as extracting structured data from news articles, financial reports, academic papers, or biomedical literature. It's used to populate databases or knowledge graphs with structured information.

Ex of Information Extraction


If quality is good, enters all valid fields into a data entry software
Data is uploaded
A manual reviewer checks image quality
Passes the same invoice and data fields entered to 2 other reviewers to check for errors
First reviewer finds that invoice ID is incorrect
Next reviewer finds invoice date is wrong, corrects it
Processes complete

- This is a screenshot explaining how we can extract information from an Invoice.
- Traditionally, when dealing with financial documents, people often have to ***manually search for and extract*** the necessary information. This can be a time-consuming and error-prone process, especially if the data is spread across multiple documents or if there is a large volume of data to handle.

Ex of Information Extraction

- With the use of NLP algorithms, it becomes possible to ***automate the process of extracting required information from these documents.***
- For example, consider we're going through a company’s financial information from a few documents.
- Usually, we search for some required information when the data is digital or manually check the same. But with information extraction NLP algorithms, we can automate the data extraction of all required information such as tables, company growth metrics, and other financial details from various kinds of documents ***(PDFs, Docs, Images etc.).***

Information Extraction (IE)

- Information Extraction from text data can be achieved by leveraging Deep Learning and NLP techniques like Named Entity Recognition.
- However, if we build one from scratch, we should decide the algorithm considering the type of data we're working on, such as invoices, medical reports, etc.
- This is to make sure the model is specific to a particular use case.

How does IE works?

- To understand the mechanics of Information Extraction NLP algorithms, we should ***understand the kind of data*** we are working on. This will help us to sort out the information we want to extract from the unstructured data. For example, for invoice related information, the algorithm should understand the invoice items, company name, billing address etc. While working on medical reports, it should identify and extract patient names, drug information, and other general reports.
- After curating the data, we’ll then start applying the information extraction NLP techniques, to process and build models around the data.

Most common techniques for IE

- Tokenization
- Parts of Speech Tagging
- Dependency Graphs

Tokenization

- Computers usually won't understand the language we speak or communicate with. Hence, we break the language, basically the words and sentences, into tokens and then load it into a program. The process of breaking down language into tokens is called tokenization.

- For example, consider a simple sentence: "NLP information extraction is fun". This could be tokenized into:

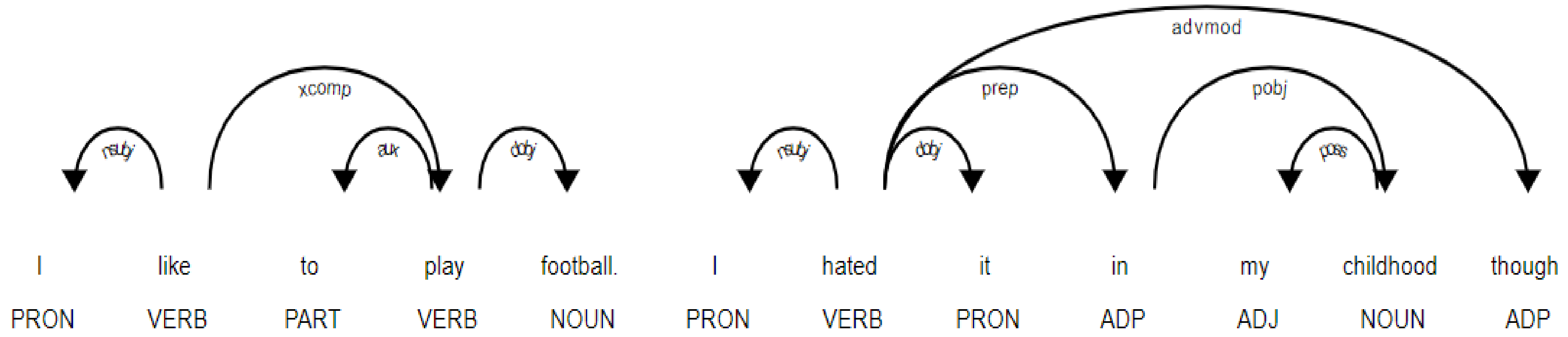
  1. One-word (sometimes called unigram token): NLP, information, extraction, is, fun
  2. Two-word phrase (bigram tokens): NLP information, information extraction, extraction is, is fun, fun NLP
  3. Three-word sentence (trigram tokens): NLP information extraction, information extraction is, extraction is fun

Code for Tokenization

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K.
startup for $1 billion")
for token in doc:
   print(token.text)
```

Parts of Speech Tagging

- Tagging parts of speech is very crucial for information extraction from text.
- It'll help us understand the **structure and context** of the text data.
- We usually refer to text from documents as "unstructured data" – data with no defined structure or pattern.
- Hence, with POS tagging we can use techniques that will provide the context of words or tokens used to categories them in specific ways.



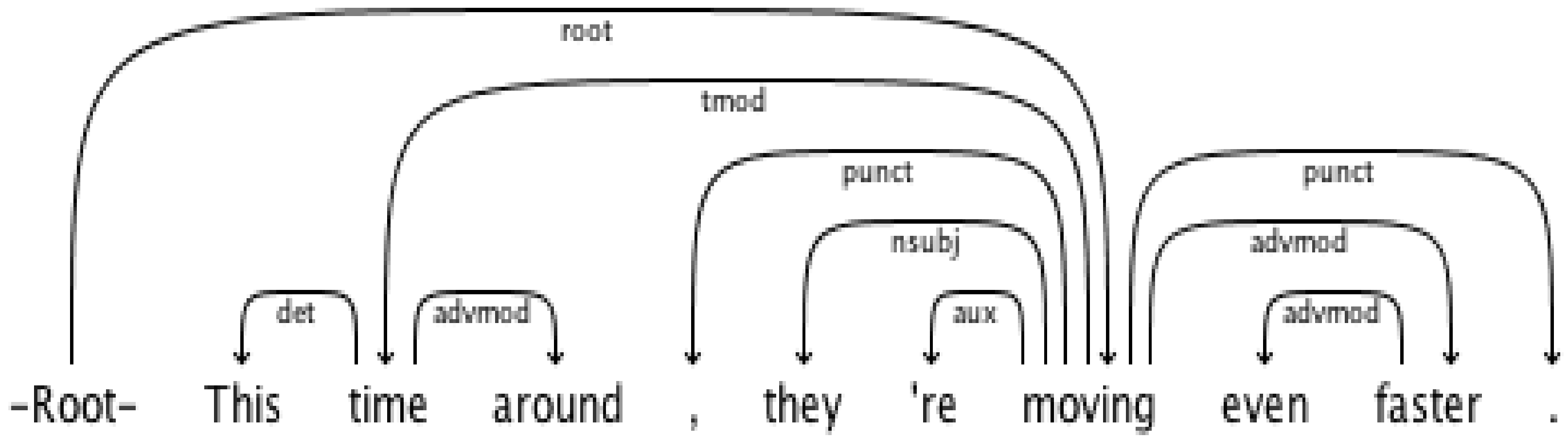
Parts of Speech Tagging

- In parts of speech tagging, all the tokens in the text data get categorised into different word categories, such as nouns, verbs, adjectives, prepositions, determiners, etc.
- This additional information connected to words enables further processing and analysis, such as sentiment analytics, lemmatization, or any reports where we can look closer at a specific class of words.

```
import spacy
NLP = spacy.load("en_core_web_sm")
doc = NLP("Apple is looking at buying U.K.
startup for $1 billion")
for token in doc:
   print(token.text, token.pos_)
```

Dependency Graphs

- Dependency graphs help us find relationships between neighbouring words using directed graphs. This relation will provide details about the dependency type (e.g. Subject, Object etc.).
- Following is a figure representing a dependency graph of a short sentence. The arrow directed from the word faster indicates that faster modifies moving, and the label `advmod` assigned to the arrow describes the exact nature of the dependency.

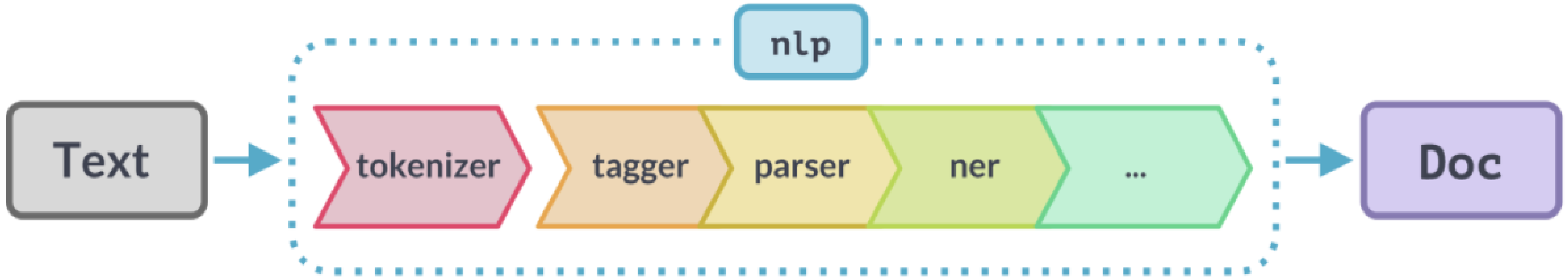


Dependency Graphs

```
import spacy
from spacy import displacy
NLP = spacy.load("en_core_web_sm")
doc = NLP("This is a sentence.")
displacy.serve(doc, style="dep")
```

Named Entity Recognition (NER) with Spacy

- Spacy is an open-source NLP library for advanced Natural Language Processing in Python and Cython.
- To extract information with spacy NER models are widely leveraged.



•Detecting the entities from the text (person, organization, place/location

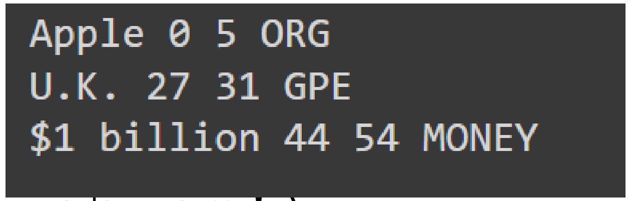
•Classifying them into different categories

Role of NER in the context of IE

- **Entity Identification:**
  - Role: NER identifies entities such as names of people, organizations, locations, dates, and other specific terms within the text.
  - Example: In a sentence like "Google was founded by Larry Page and Sergey Brin," NER would identify "Google" as an organization and "Larry Page" and "Sergey Brin" as persons.
- **Disambiguation:**
  - Role: NER helps disambiguate entities by contextually understanding the text and assigning the correct category.
  - Example: In the sentence "Apple is releasing a new product," NER identifies "Apple" as an organization rather than the fruit.
- **Automated Content Creation:**
  - Role: NER assists in generating structured reports and content by extracting and organizing relevant entities from unstructured text.
  - Example: Creating automated financial reports by extracting company names, stock prices, and dates from financial news.

NER with Spacy

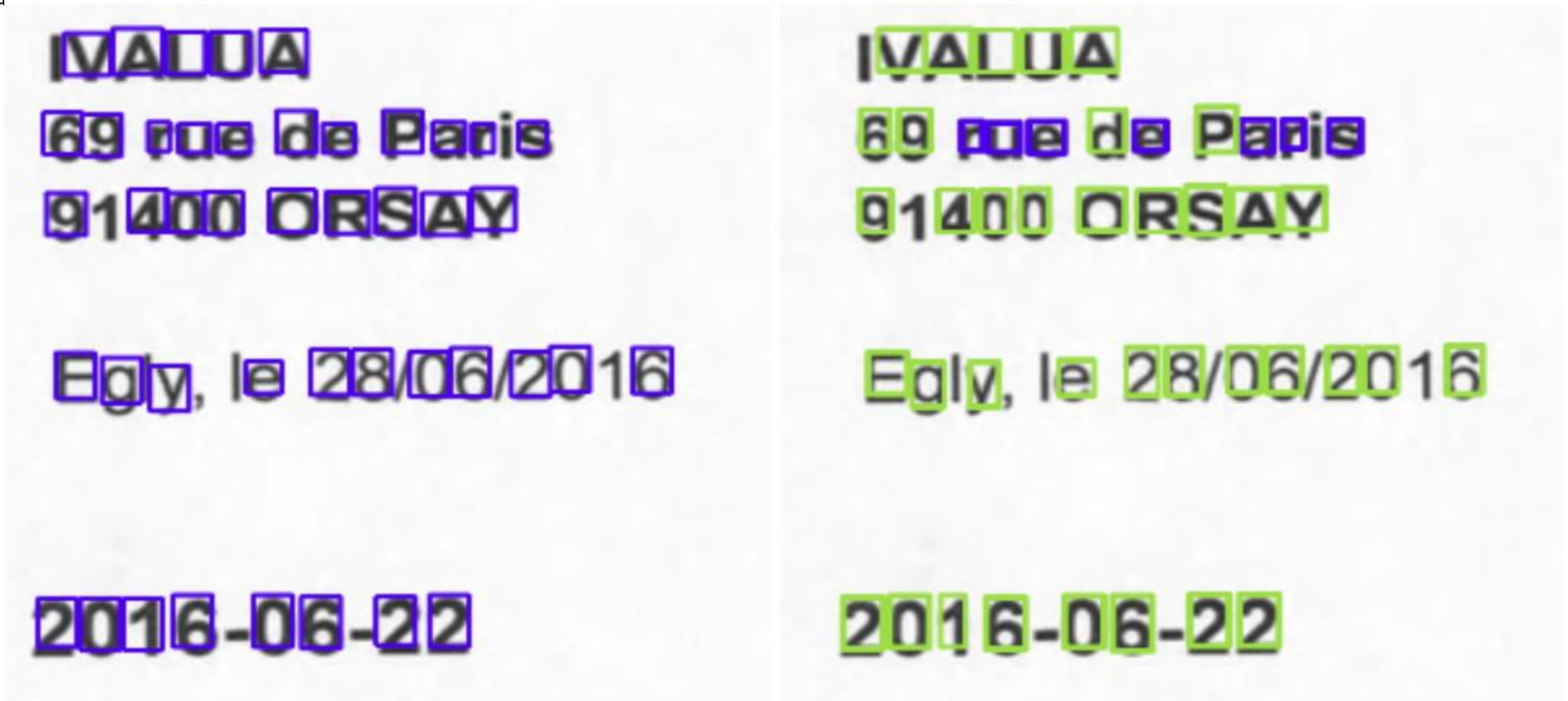
```
# import spacy
import spacy
# load spacy model
NLP = spacy.load('en_core_web_sm')
# load data
sentence = "Apple is looking at buying U.K.
startup for $1 billion"
doc = NLP(sentence)
# print entities
for ent in doc.ents:
    print(ent.text, ent.start_char,
ent.end_char, ent.label_)
```



```
Apple 0 5 ORG
U.K. 27 31 GPE
$1 billion 44 54 MONEY
```

An Example of Information Extraction

- Several industries deal with lots of documents every day and rely on ***manual work***.
- Those include finance, medical chains, transportation, and construction.
- Using NLP information extraction techniques on documents will allow everyone on the teams to search, edit, and analyse important transactions and details across bu



IVALUA
69 rue de Paris
91400 ORSAY

Egly, le 28/06/2016

2016-06-22

IVALUA
69 rue de Paris
91400 ORSAY

Egly, le 28/06/2016

2016-06-22

Steps on how IE from text

1. *Information Collection*

- Firstly, ***collect the data from different sources*** to build an information extraction model.
- Usually, we see documents on emails, cloud drives, scanned copies, computer software, and many other sources for business.
- Hence, we’ll have to write different scripts to collect and store information in one place.
- This is usually done by either using ***APIs on the web or building RPA*** (Robotic Process Automation) pipelines.

Steps on how IE from text

1. Data Collection sources:

•Emails: Extracting information from emails.

•Cloud Drives: Accessing documents stored in cloud services like Google Drive, Dropbox, etc.

•Scanned Copies: Processing and extracting text from scanned documents using OCR (Optical Character Recognition) tools.

•Computer Software: Integrating with business software to fetch data.

•Other Sources: Any other relevant sources for the business data.

Steps for data collection using API

- APIs (Application Programming Interfaces) provide a standardized way to access and interact with different services and applications on the web.
- ***Example: Collecting Data from Google Drive***
- Google Drive API allows you to access files stored in Google Drive. Below is a simplified example in Python to illustrate how to collect data from Google Drive using its API.
- ***Step 1: Setting Up Google Drive API***
  - Enable Google Drive API: Go to the Google Cloud Console, enable the Google Drive API, and create credentials.
  - Install Google Client Library: Use pip to install the required library
    “pip install google-auth google-auth-oauthlib google-auth-httplib2 google-api-python-client”
  - Authenticate: Use OAuth 2.0 to authenticate and authorize access.
- ***Step 2: Writing the Script***

Steps on how IE from text

2. ***Process Data***

- After we collect the data, the next step is to process them.
- Usually, documents are two types: electronically generated (editable) and the other non-electronically generated (scanned documents).
- For the electronically generated documents, we can directly send them into the preprocessing pipelines.
- Still, we’ll need OCR to first read all the data from images and then send them into preprocessing pipelines for the scanned copies.
- We can either use open-source tools like Tesseract or any online services like Nanonets or Textract. After all the data is in editable or electronic format, we can then apply to pre-process steps like Tokenization and POS tagging and then use data loaders to load the data into the NLP information extraction models.

Steps on how IE from text

3. ***Choosing the right model***
    - As discussed in the above sections, choosing a suitable model mostly depends on the type of data we're working with. Today, there are several state-of-the-art models we could rely on. Below are some of the frequently use open-source models:
    - Named Entity Recognition on CoNLL 2003 (English)
    - Key Information Extraction From Documents: Evaluation And Generator (GAN)
    - Deep Reader: Information extraction from Document images via relation extraction and Natural Language
    - These are some of the information extraction models. However, these are trained on a particular dataset. If we are utilising these on our models, we'll need to experiment on the hyperparameters and fine-tune the model accordingly.
    - The other way is to utilize the pre-trained models and fine-tuning them based on our data. For Information Extraction from text, in particular, BERT models are widely used.

Steps on how IE from text

4. Evaluation of the Model

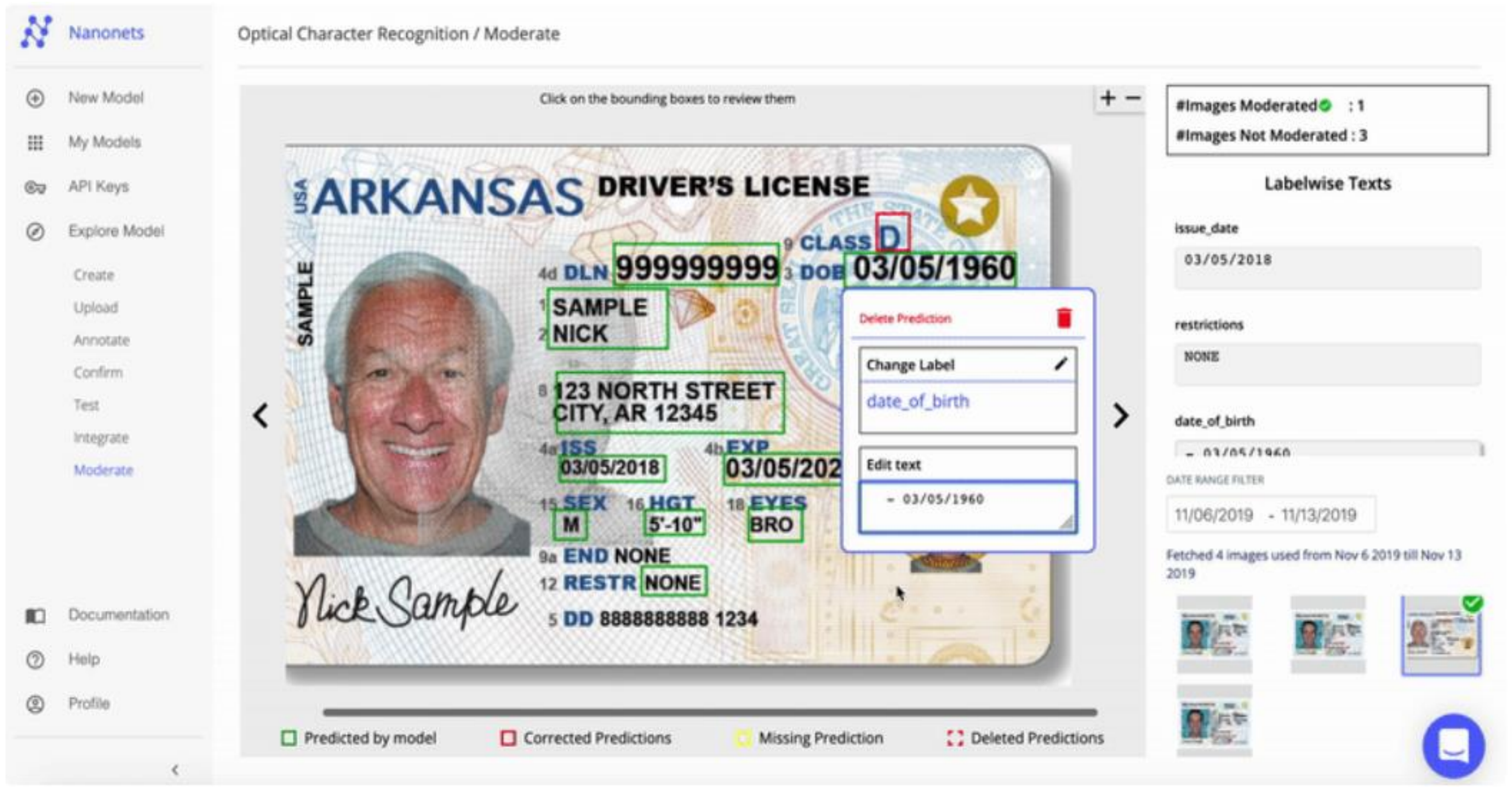
- Evaluate the training process is crucial before we use the models in production. This is usually done by creating a testing dataset and finding some key metrics:
- ***Accuracy: the ratio of correct predictions made against the size of the test data.***
- ***Precision: the ratio of true positives and total predicted positives.***
- ***Recall the ratio of true positives and total actual positives.***
- ***F1-Score: harmonic mean of precision and recall.***
- Different metrics take precedence when considering different use cases. In invoice processing, we know that an increase in the numbers or missing an item can lead to losses for the company. This means that besides needing a good accuracy, we also need to make sure the false positives for money-related fields are minimum, so aiming for a high precision value might be ideal. We also need to ensure that details like invoice numbers and dates are always extracted since they are needed for legal and compliance purposes. Maintaining a high recall value for these fields might take precedence.

Steps on how IE from text

5. Deploying model in production

- The full potential of the NLP models only knows when they are deployed in production.
- Today, as the world is entirely digital, these models are stored on cloud servers with a suitable background. In most cases, Python is utilised as its more handy programming language when it comes to Text data and machine learning. The model is either **exported as API or an SDK (**software development kit) for integrating with business tools.
- However, we need not build everything from scratch as there are several tools and online services for this kind of use-cases. For example, Nanonets has a highly accurate, fully trained invoice information extraction NLP model, and you can directly integrate on our applications using APIs or supported SDKs.

Example - IE on ID card



#Images Moderated : 1
#Images Not Moderated : 3

Change Label: date_of_birth
Edit text: - 03/05/1960

issue_date: 03/05/2018
restrictions: NONE
date_of_birth: - 03/05/1960

11/06/2019 - 11/13/2019
Fetched 4 images used from Nov 6 2019 till Nov 13 2019

ARKANSAS DRIVER'S LICENSE
USA
SAMPLE
9 CLASS D
4d DLN 999999999 3 DOB 03/05/1960
1 SAMPLE
2 NICK
8 123 NORTH STREET
CITY, AR 12345
4a ISS 03/05/2018 4b EXP 03/05/202
15 SEX M 16 HGT 5'-10" 18 EYES BRO
9a END NONE
12 RESTR NONE
5 DD 8888888888 1234

https://nanonets.com/blog/information-extraction/#how-does-information-extraction-work?

Applications of IE

1. KYC Automation: Automate the process of KYC by extracting ethical information from customer's identity documents.
2. Invoice Automation: Automate the process of invoice information extraction.
3. Healthcare Systems: Manage medical records by identifying patient information and their prescriptions.
4. Financial Investigation: Extract import information from financial documents. (Tax, Growth, Quarterly Revenue, Profit/Losses)

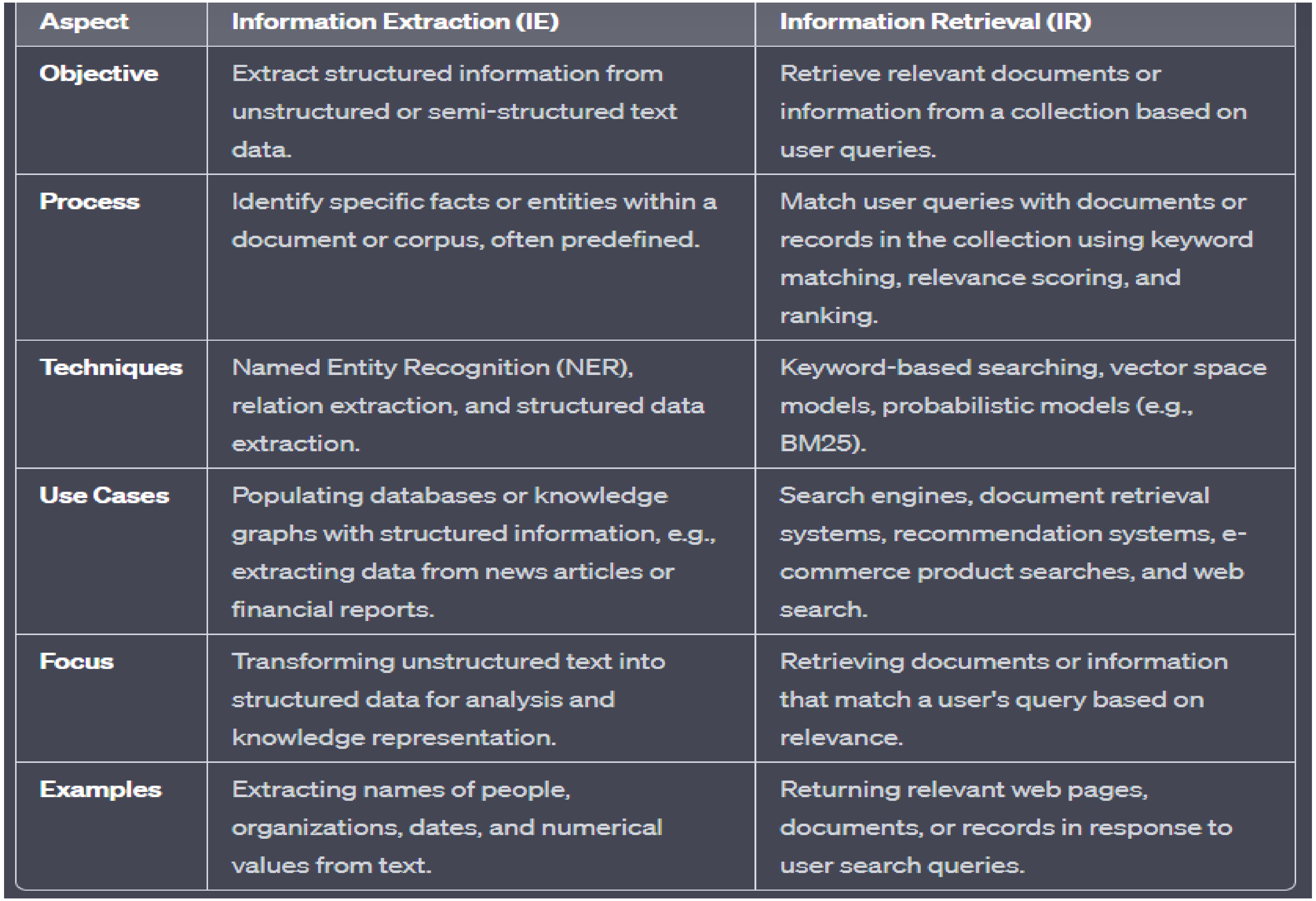
Information Retrieval

Information Retrieval (IR)

What is information retrieval?

- Information Retrieval is focused on ***retrieving relevant documents*** or pieces of information from a large collection (e.g., a database, web corpus) in **response to a user's query.**
- IR involves ***matching user queries with documents or records in the database***. It uses techniques like ***keyword matching, relevance scoring, and ranking*** to identify the most relevant documents based on the user's information needs.
- IR is widely used in search engines, document retrieval systems, and recommendation systems. When you perform a web search, the search engine is using IR techniques to retrieve and rank relevant web pages.
- ***Google Search is the most famous example of information retrieval.***

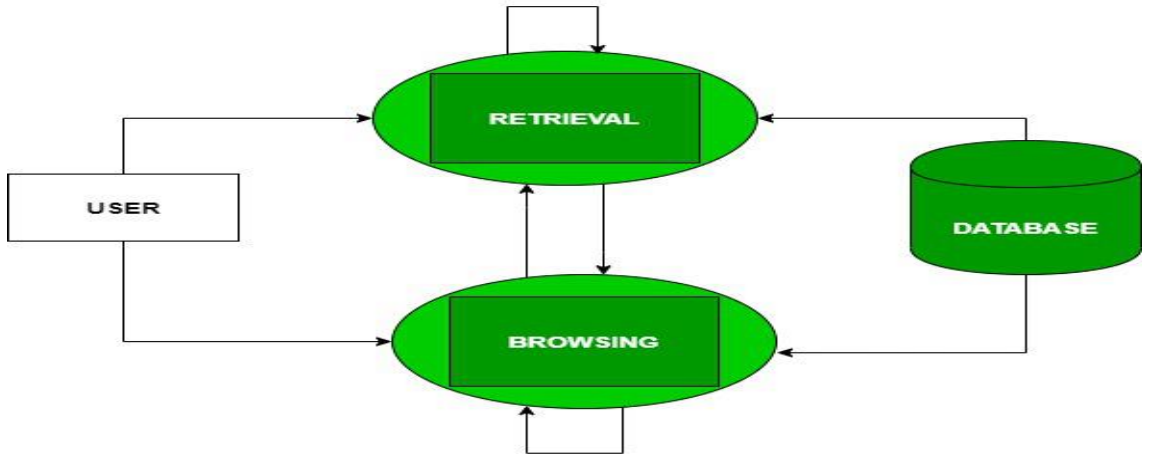
IE vs IR



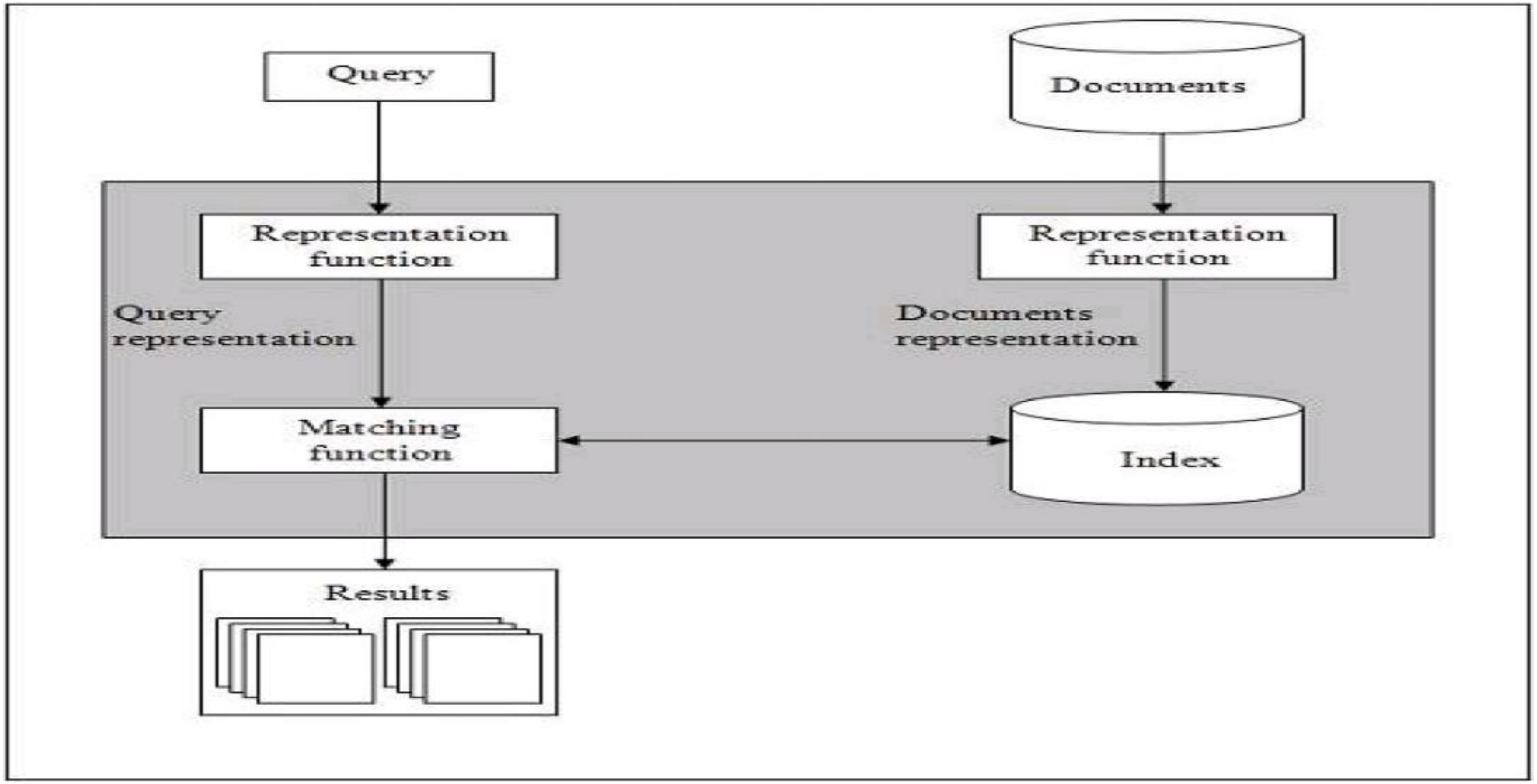
Aspect	Information Extraction (IE)	Information Retrieval (IR)
Objective	Extract structured information from unstructured or semi-structured text data.	Retrieve relevant documents or information from a collection based on user queries.
Process	Identify specific facts or entities within a document or corpus, often predefined.	Match user queries with documents or records in the collection using keyword matching, relevance scoring, and ranking.
Techniques	Named Entity Recognition (NER), relation extraction, and structured data extraction.	Keyword-based searching, vector space models, probabilistic models (e.g., BM25).
Use Cases	Populating databases or knowledge graphs with structured information, e.g., extracting data from news articles or financial reports.	Search engines, document retrieval systems, recommendation systems, e-commerce product searches, and web search.
Focus	Transforming unstructured text into structured data for analysis and knowledge representation.	Retrieving documents or information that match a user's query based on relevance.
Examples	Extracting names of people, organizations, dates, and numerical values from text.	Returning relevant web pages, documents, or records in response to user search queries.

Information Retrieval (IR)

- Information retrieval (IR) may be defined as a software program that deals with the storage, organization, retrieval and evaluation of information from document repositories particularly textual information.
- The system assists users in finding the information they require but it does not explicitly return the answers of the questions.
- It informs the existence and location of documents that might consist of the required information. The documents that satisfy user’s requirement are called ***relevant documents.*** A perfect IR system will retrieve only relevant documents.



Basics of Information Retrieval (IR)



- It is clear from the above diagram that a user who needs information will have to formulate a request in the form of query in natural language.
- Then the IR system will respond by retrieving the relevant output, in the form of documents, about the required information.

Basics of Information Retrieval (IR)

- A user who needs information will have to formulate a request in the form of a query in natural language. After that, the IR system will return output by retrieving the relevant output, in the form of documents, about the required information.
- ***The step by step procedure of these systems are as follows:***
  - Indexing the collection of documents.
  - Transforming the query in the same way as the document content is represented.
  - Comparing the description of each document with that of the query.
  - Listing the results in order of relevancy.

Basics of Information Retrieval (IR)

- ***Retrieval Systems consist of mainly two processes:***
  - Indexing
  - Matching

What is Indexing?

•Indexing involves parsing and storing data from web pages in a structured format. When a search engine crawls web pages, it indexes their content to make it searchable.

•The search engine scans the content of web pages, extracts key information (like words, phrases, and metadata), and creates an index. This index is a data structure that maps keywords to the web pages where they appear.

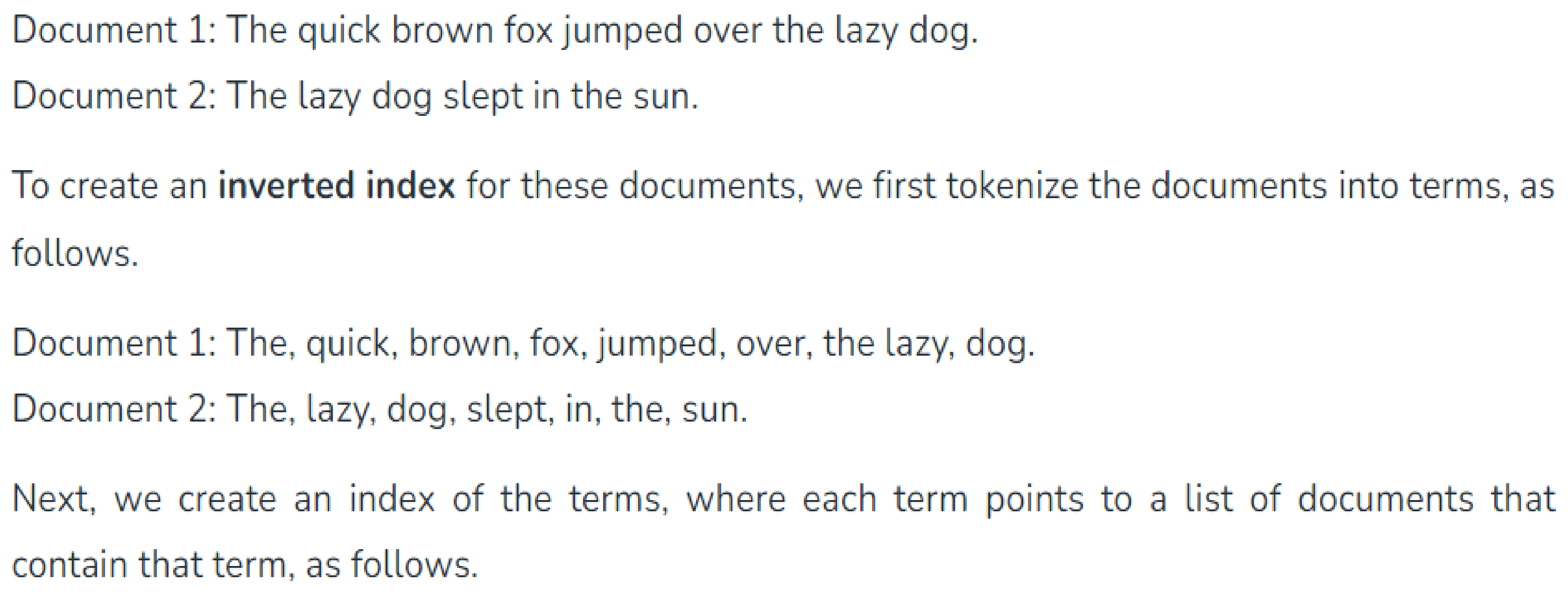
•This allows the search engine to quickly retrieve relevant pages when a user performs a search query. Instead of **searching the entire internet, the engine looks up the index to find matching pages.**

Indexing

- Indexing is the ***process of preparing and organizing the collection of documents*** or data to make it efficient for retrieval.
- **This involves several tasks:**
  1. ***Tokenization:***
     - Breaking down text into individual words or tokens.
     - For example, the sentence "I love programming" would be tokenized into three tokens: "I," "love," and "programming."
  2. ***Removing Frequent Words:***
     - Common words that don't contribute much to the meaning of the text, such as "the," "and," "in," are often removed. These are known as stop words.
  3. ***Stemming:***
     - Reducing words to their root or base form. For example, "running" and "ran" would both be stemmed to "run."
  4. ***Creating an Inverted Index:***
     - This is a data structure that maps words or terms to the documents in which they appear. It speeds up the retrieval process by allowing the system to quickly locate documents containing specific terms.

Inverted Index

- An Inverted Index is a data structure used in information retrieval systems to efficiently retrieve documents or web pages containing a specific term or set of terms. In an inverted index, the index is organized by terms (words), **It lists words (terms) and then the documents where these terms appear:**
- ***Example:***



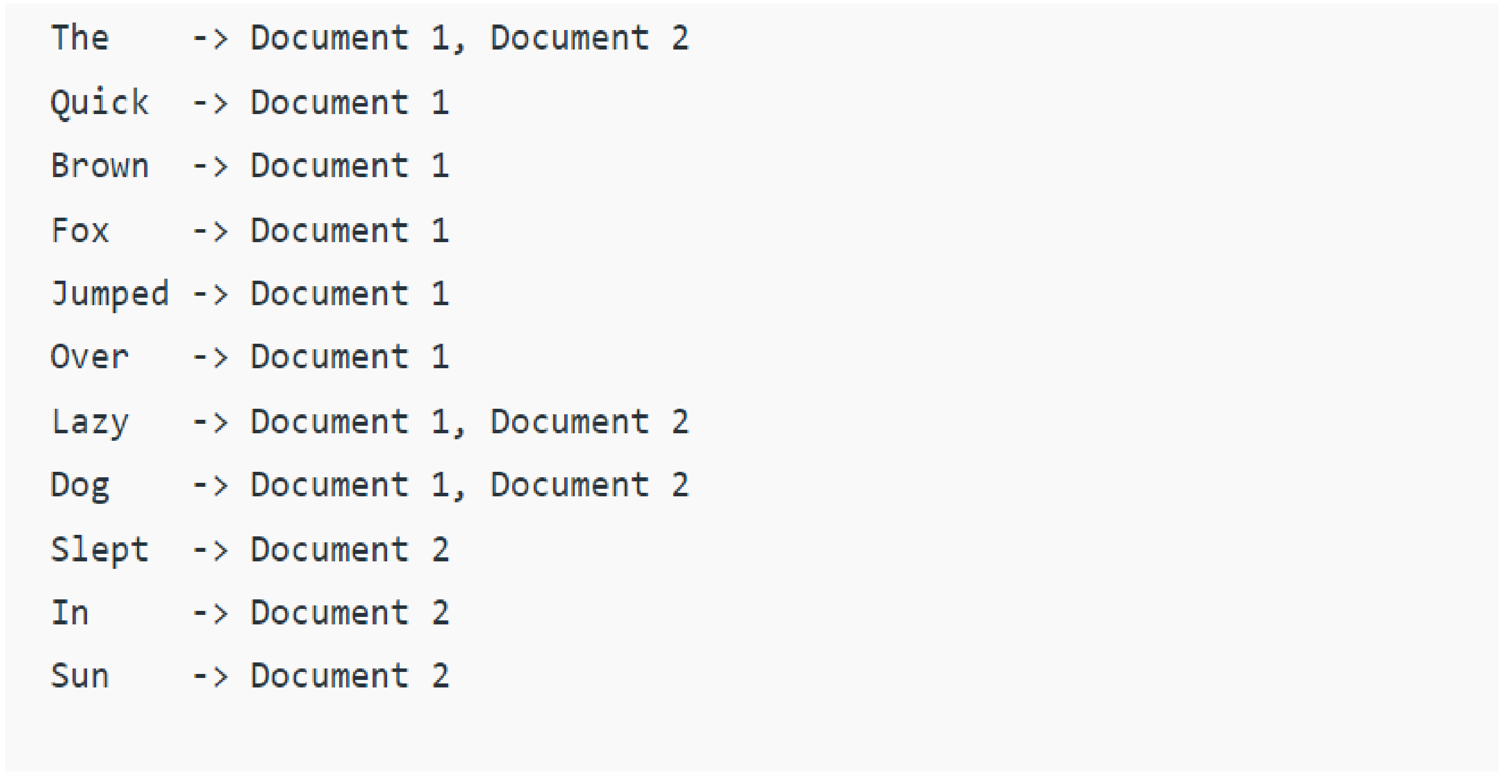
Document 1: The quick brown fox jumped over the lazy dog.
Document 2: The lazy dog slept in the sun.

To create an **inverted index** for these documents, we first tokenize the documents into terms, as follows.

Document 1: The, quick, brown, fox, jumped, over, the lazy, dog.
Document 2: The, lazy, dog, slept, in, the, sun.

Next, we create an index of the terms, where each term points to a list of documents that contain that term, as follows.

Inverted Index



```
The    -> Document 1, Document 2
Quick  -> Document 1
Brown  -> Document 1
Fox    -> Document 1
Jumped -> Document 1
Over   -> Document 1
Lazy   -> Document 1, Document 2
Dog    -> Document 1, Document 2
Slept  -> Document 2
In     -> Document 2
Sun    -> Document 2
```

- To search for documents containing a particular term or set of terms, the ***search engine queries the inverted index*** for those terms and retrieves the list of documents associated with each term.
- The search engine can then use this information to **rank the documents** based on relevance to the query and present them to the user in order of importance.

Matching

- Matching is the process of ***finding a measure of similarity between two text representations***. Once the indexing is complete, the retrieval system can match user queries to the indexed documents.

- **This involves the following steps:**

  1. ***Transforming the user query:*** The user's query is processed in a similar way to how the documents were indexed. This includes tokenizing the query, removing stop words, and stemming.
  2. ***Comparing with documents:*** The system then ***compares the processed query with the indexed documents.*** It identifies documents that contain terms matching those in the query.
  3. ***Ranking:*** The retrieved documents are ranked by ***relevance.*** Various algorithms, such as ***TF-IDF (Term Frequency-Inverse Document Frequency) or BM25***, are used to assign scores to documents based on how well they match the query.
  4. ***Presenting results:*** Finally, the system presents the ranked list of documents to the user, with the most relevant ones at the top.

Relevance

- Relevance refers to ***how effectively a document or collection of documents that has been retrieved satisfies the user's information needs.***
- Concerns about timeliness, authority, or uniqueness of the outcome are examples of relevant considerations.

TF-IDF

- The relevance of a document is computed based on the following parameters using the TF-IDF method:
- ***TF:*** It stands for ***Term Frequency*** which is simply the number of times a given term appears in that document.

TF (i, j) = (count of ith term in jth document)/(total terms in jth document)

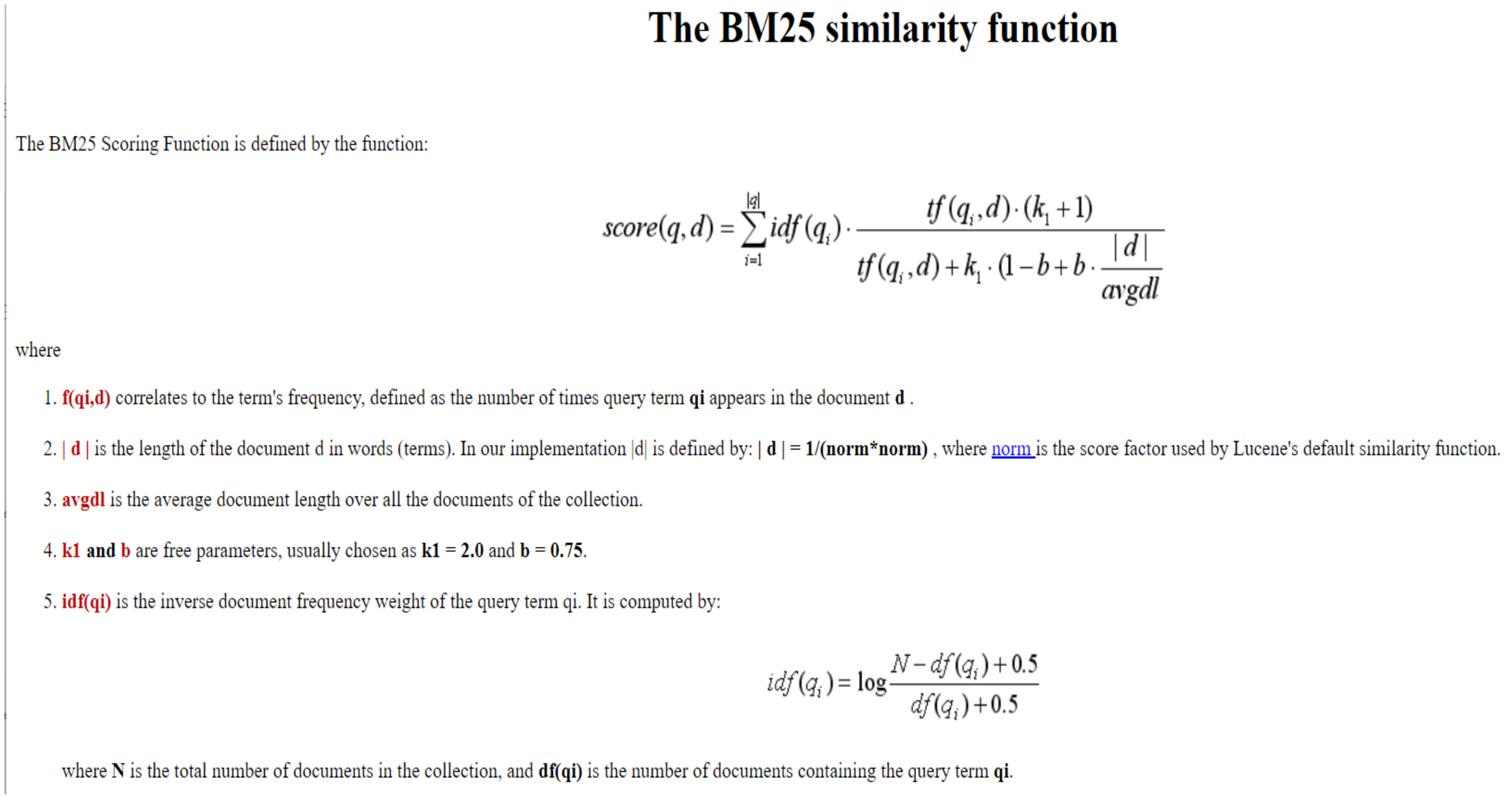
- ***IDF:*** It stands for ***Inverse Document Frequency*** which is a measure of the general importance of the term.

IDF (i) = (total no. of documents)/(no. of documents containing ith term)

TF-IDF Score (i, j) = TF * IDF

BM25

- BM25 is a ranking function that ranks a set of documents based on the query terms appearing in each document



The BM25 similarity function

The BM25 Scoring Function is defined by the function:

$$score(q,d)=\sum_{i=1}^{|q|} idf(q_i)\cdot\frac{tf(q_i,d)\cdot(k_1+1)}{tf(q_i,d)+k_1\cdot(1-b+b\cdot\frac{|d|}{avgdl}}$$

where

1. **f(qi,d)** correlates to the term's frequency, defined as the number of times query term **qi** appears in the document **d** .
2. | **d** | is the length of the document d in words (terms). In our implementation |d| is defined by: | **d** | = **1/(norm*norm)** , where norm is the score factor used by Lucene's default similarity function.
3. **avgdl** is the average document length over all the documents of the collection.
4. **k1 and b** are free parameters, usually chosen as **k1** = **2.0** and **b** = **0.75**.
5. **idf(qi)** is the inverse document frequency weight of the query term qi. It is computed by:

$$idf(q_i)=\log\frac{N-df(q_i)+0.5}{df(q_i)+0.5}$$

where **N** is the total number of documents in the collection, and **df(qi)** is the number of documents containing the query term **qi**.

IR models

- Information retrieval models predict and explain what a user will find in relevance to the given query. These are basically a pattern that defines the retrieval procedure and consists of the following:
- Mathematically, a ***retrieval model consists of the following components:***
  - D: Representation for documents.
  - R: Representation for queries.
  - F: The modeling framework for D, Q along with the relationship between them.
    - F represents the modeling framework that ***combines the document and query representations and captures the relationship between them***. This modeling framework could be a machine learning model, deep learning architecture, or any other algorithm that processes the representations of documents and queries to generate relevant rankings or similarity scores.
  - R (q, di): A ranking or similarity function that orders the documents with respect to the query.

Types of IR models

- ***Classic IR Model:*** This is described as the most basic and straightforward IR model, grounded in mathematical information. The three traditional models mentioned are ***Boolean, Vector, and Probabilistic***.
- ***Non-Classic IR Model:*** In contrast to the traditional models, non-classic IR models are based on different concepts apart from ***probability, similarity, and Boolean operations***. Examples include situation theory models, information logic models, and interaction models.
- ***Alternative IR Model:*** These models are improvements to traditional IR models and incorporate unique approaches from other domains. Alternative IR models mentioned include ***fuzzy models, cluster models, and latent semantic indexing (LSI) models.***

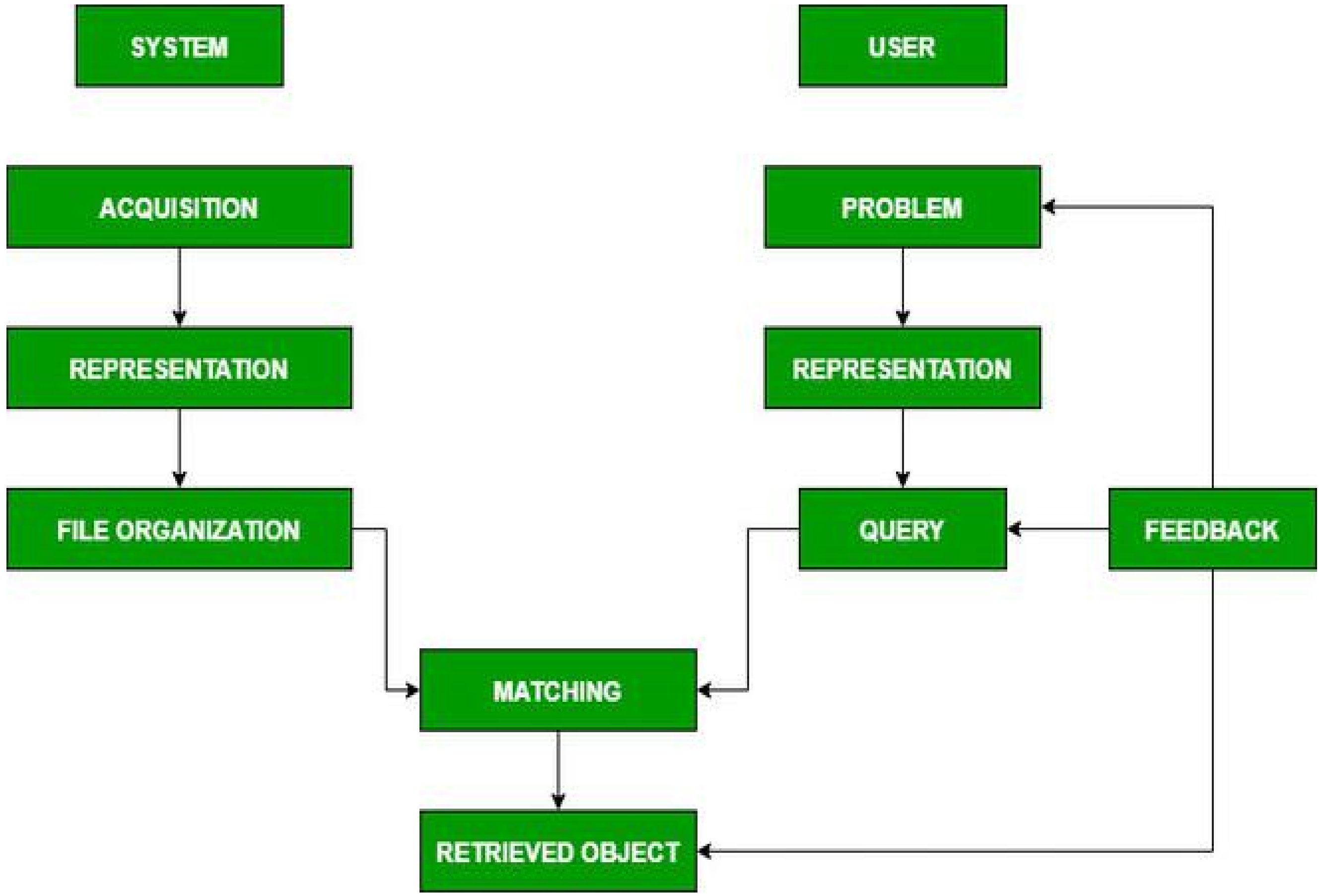
Classical problem in IR systems

- ***What is Ad-hoc Retrieval?***
  - In ad-hoc retrieval, it means that a ***search or retrieval process is conducted on-the-fly***, in response to a specific user query or request, with the goal of finding information that is relevant to that particular query.
  - It's not a pre-arranged or pre-organized search but rather a flexible and tailored retrieval process designed to meet the immediate needs of the user.

Classical problem in IR systems

- ***Ad-hoc Retrieval:*** Ad-hoc retrieval is described as a ***classical problem*** within the information retrieval paradigm.
  - It involves presenting a natural language query to retrieve relevant information. In other words, it's the process of searching for information based on a user's query.
  - Handling Irrelevant Information: After the user submits a query, the retrieved information includes both relevant and non-relevant results. ***The non-relevant results, in this context, are referred to as "ad hoc retrieval difficulties."*** These are documents or pieces of information that don't satisfy the user's search criteria.
  - Real-World Example: You provide a practical example to illustrate the concept. If someone searches for something on the Internet and receives a list of specific websites that are relevant to their search, ***there may also be some results that are not related to their query. These unrelated results are a manifestation of the ad-hoc retrieval issue.***

Components of IR systems



Components of IR systems

- ***Acquisition:*** This step involves collecting data from various web resources, specifically text-based documents. It is carried out by ***web crawlers***, which are automated programs that navigate the web, visit websites, and collect the necessary data. The collected data is then stored in a ***database for further processing***.
- ***Representation:*** In this step, the acquired data is organized and prepared for efficient retrieval. It includes ***indexing, which involves creating data structures that facilitate quick access to information.*** Representation techniques can include:
- Free-Text Terms: These are the actual words or phrases found in the documents.
- Controlled Vocabulary: This refers to a predefined set of terms used to categorize and describe documents.
- Manual and Automatic Techniques: Both manual human input and automated algorithms can be used to enhance the representation of data.
- Examples: Abstracting, which involves summarizing documents, and Bibliographic description, which includes details like author, title, source, date, and metadata.

Boolean Model

- Boolean Model is the oldest model for Information Retrieval (IR). These models are based on set theory and Boolean algebra, where
  - Documents: Sets of terms
  - Queries: Boolean expressions on terms
- As a response to the query, the set of documents that satisfied the Boolean expression are retrieved.
- The boolean model can be defined as:
  - D: It represents a set of words, i.e, the indexing terms present in a document. Here, each term is either present (1) or absent (0) in the document.
  - Q: It represents a Boolean expression, where terms are the index terms and operators are logical products such as:
  - AND,
  - Logical sum − OR,
  - Logical difference − NOT.
  - F: It represents a Boolean algebra over sets of terms as well as over sets of documents.

Boolean Model

- If we talk about the relevance feedback, then in the Boolean IR model the Relevance prediction can be defined as follows:
- R: A document is predicted as relevant to the query expression if and only if it satisfies the query expression as −
- $((text \vee information) \wedge rerieval \wedge \tilde{}\ theory)$
- We can explain this model by a query term as an unambiguous definition of a set of documents.

Semantic Search

Semantic Search

- Semantic search is an advanced approach to information retrieval that goes beyond matching keywords in search queries. Its primary goal is to understand the meaning and context of the words and phrases used in a query.
- By comprehending the user's intent and the context of their search, semantic search aims to provide search results that are more relevant to the user's needs.
- It goes beyond traditional keyword-based search by considering the meaning and relationships between words and concepts.

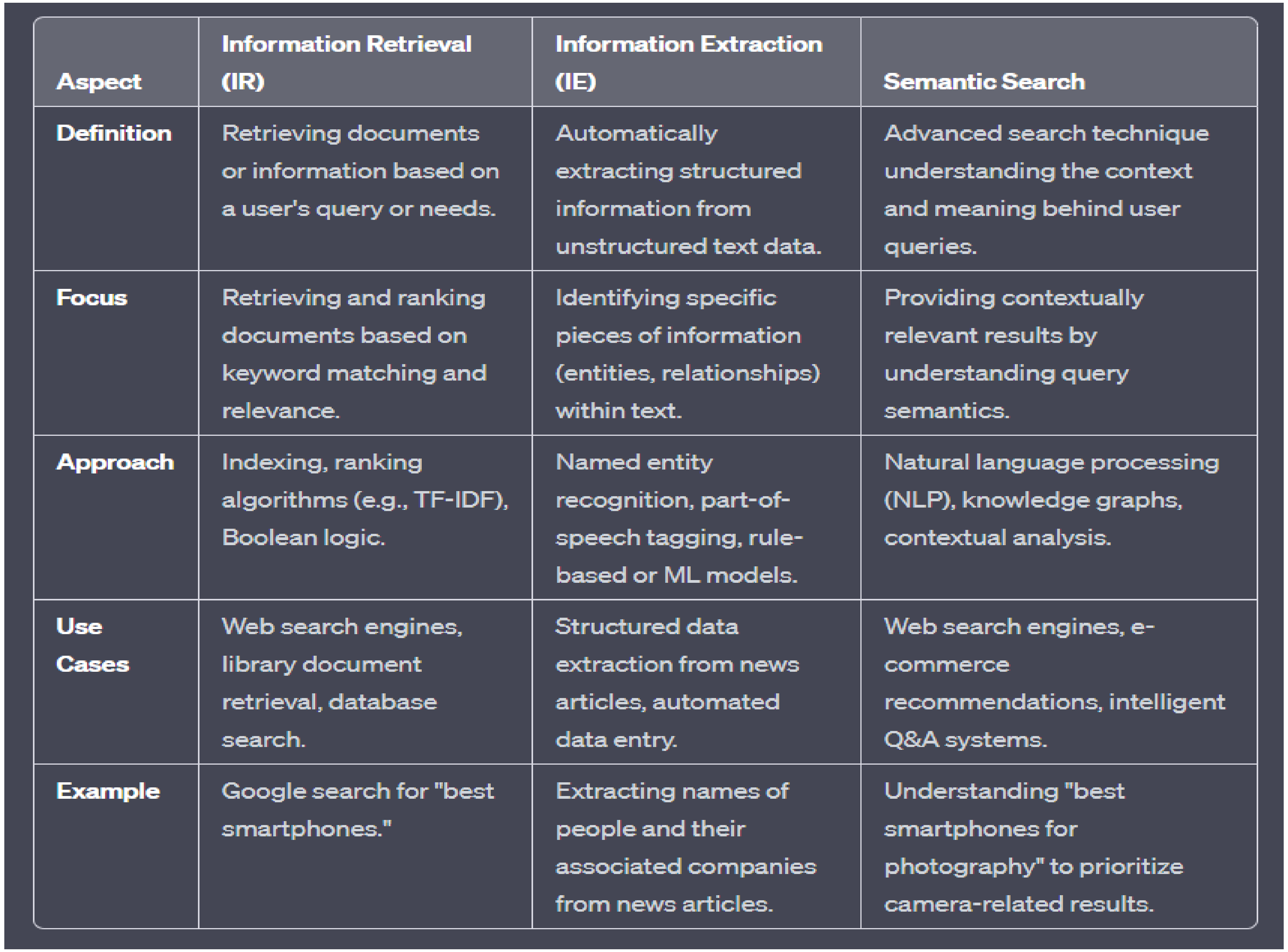
Semantic Search

- Semantic search involves analyzing the words in a search query to understand not only their individual definitions but also how they relate to each other in the given context. For example, if you search for "apple," Google's semantic search considers whether you mean the fruit or the technology company based on the context of your search.
- The ultimate goal of semantic search is to offer more relevant search results to users. By understanding the meaning and context of a search query, Google can better match user intent, delivering results that are more likely to answer the user's questions or meet their needs.

Semantic Search

- ***The Importance of Semantics search:*** The emphasis on semantics is an evolution in search technology that has greatly improved the quality of search results. It's especially valuable in understanding ambiguous queries, natural language, and complex questions.

IR vs IE vs Semantic Search



Aspect	Information Retrieval (IR)	Information Extraction (IE)	Semantic Search
Definition	Retrieving documents or information based on a user's query or needs.	Automatically extracting structured information from unstructured text data.	Advanced search technique understanding the context and meaning behind user queries.
Focus	Retrieving and ranking documents based on keyword matching and relevance.	Identifying specific pieces of information (entities, relationships) within text.	Providing contextually relevant results by understanding query semantics.
Approach	Indexing, ranking algorithms (e.g., TF-IDF), Boolean logic.	Named entity recognition, part-of-speech tagging, rule-based or ML models.	Natural language processing (NLP), knowledge graphs, contextual analysis.
Use Cases	Web search engines, library document retrieval, database search.	Structured data extraction from news articles, automated data entry.	Web search engines, e-commerce recommendations, intelligent Q&A systems.
Example	Google search for "best smartphones."	Extracting names of people and their associated companies from news articles.	Understanding "best smartphones for photography" to prioritize camera-related results.

Semantic Search Example

When you search for "pizza" on Google:

1. ***Local Search Results:*** Google often shows results related to nearby pizzerias and restaurants because many people search for "pizza" when they want to order it.
2. ***User Behavior:*** Google's search results are influenced by what most people are searching for. If most people search for "pizza" to order, Google will prioritize these results.
3. ***Personalization:*** If you've been searching for pizza recipes, Google will use this information to personalize your results. You'll see more pizza recipe-related content because it aligns with your interests.

- In a nutshell, Google adapts its results based on ***what people commonly search (user behaviour)*** for and also takes into ***account your individual search history (personalization)*** to show you more relevant content. This can result in different search results for different people.

Semantic Search Example

- Google performs semantic search, which is a technique used to understand the meaning and context of words in search queries, providing more relevant search results.

How does semantic search work?

1. ***Query Preprocessing:*** Before the query is transformed into embeddings, there might be preprocessing steps, such as tokenization, removal of stop words, and stemming or lemmatization to standardize and clean the query text.
2. ***Embedding Model Selection:*** The choice of embedding model (e.g., Word2Vec, GloVe, BERT) and the pre-trained embeddings can significantly impact the quality of embeddings. The search system might involve selecting the most suitable model for the task.
3. ***Vector Indexing:*** In order to efficiently perform vector searches, the system may employ indexing structures (e.g., KD-trees or Annoy) to organize the vectors of existing documents, making it faster to find nearest neighbors.

How does semantic search work?

4. ***Similarity Metric:*** When matching query vectors to document vectors, a similarity metric, such as cosine similarity or Euclidean distance, is typically used to quantify the similarity between vectors.
5. ***Threshold Selection:*** Depending on the search system, there may be a threshold set to determine which results are considered relevant. Documents or vectors that surpass a certain similarity threshold are returned as results.
6. ***Post-processing:*** After the k-nearest neighbors are found, the system may apply additional post-processing techniques, like reranking results based on other factors (e.g., recency or popularity), filtering out duplicates, or improving the presentation of results.
7. ***User Feedback Integration***: In some systems, user interactions, such as clicks on search results, can be used to refine future search results. The system might employ machine learning techniques to adapt to user preferences and improve result relevance over time.

How does semantic search work?

- Semantic search is powered by **vector search / embedding**, which enables semantic search to deliver and rank content based on context relevance and intent relevance.
- Vector search is a technology that encodes information in a particular way. It takes details about searchable information and represents them as vectors.
- Vectors are like points in multi-dimensional space, and they represent related terms or items. Each term or item has a specific position within this multi-dimensional space.
- The key role of vector search is to compare these vectors to determine how similar they are to each other. By assessing the similarity of vectors, search engines can identify content that closely aligns with the user's context and intent.

https://www.elastic.co/what-is/semantic-search

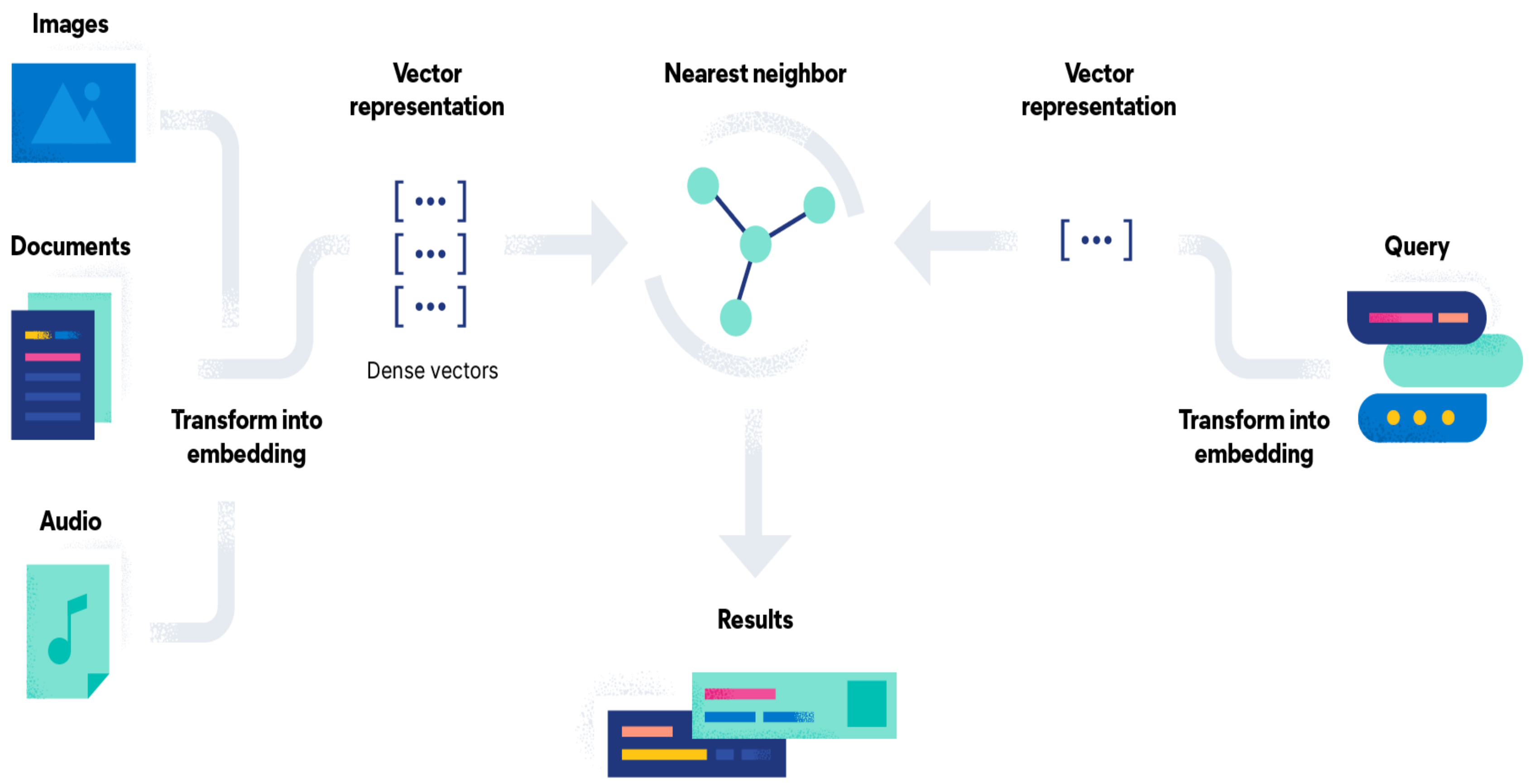
How does semantic search work?

- The search engine can deliver and rank search results based on ***two important factors***: ***Context Relevance and Intent Relevance.***
- ***Context Relevance:*** Context relevance refers to how well the content in search results fits into the larger context of the user's query. It's about understanding the user's needs beyond the specific keywords they've typed. Here's a more detailed explanation:
  - Imagine a user is searching for "Apple." Context relevance takes into account various factors:
  - User's Location: If the user is in New York City, context relevance might mean showing results related to the Apple Store in Manhattan.
  - User's Search History: If the user has been previously searching for tech-related topics, context relevance might lead to results related to Apple Inc. and its products.
  - User's Recent Behavior: If the user has recently clicked on articles about the latest iPhone model, context relevance might prioritize results about Apple's latest product releases.
- In essence, context relevance considers the broader picture – the user's location, preferences, and recent interactions – to ensure the search results are more meaningful in the given context.

How does semantic search work?

- ***Intent Relevance:*** Intent relevance is all about understanding and aligning with the specific purpose or intent behind the user's search. It ensures that the search results directly address what the user is trying to achieve. Here's a more detailed explanation:
    - Consider a user searching for "Apple." Intent relevance aims to grasp what the user wants to do with that information. It could be:
    - Purchase Intent: If the user is looking to buy an Apple product, intent relevance would prioritize showing links to online or local stores where Apple products are available.
    - Research Intent: If the user wants to learn more about Apple's history and innovations, intent relevance would show articles, Wikipedia pages, and informative resources about Apple Inc.
    - Technical Support Intent: If the user needs help with an Apple device or service, intent relevance might provide links to Apple's official support pages or forums.
- In this way, intent relevance ensures that the search results align with the specific goal or intention of the user's search. It helps deliver results that best serve the user's needs and objectives.

How does semantic search work?



Semantic search vs. keyword search

- The difference between semantic search and keyword search is that keyword search returns results that match words to words, words to synonyms, or words to similar words.
- Semantic search looks to match the meaning of the words in the query. In some cases, semantic search might not generate results with direct word matches, but it will match the user's intent.
- Keyword search engines use query expansion or relaxation tools like synonyms or word omission. They also use natural language processing and understanding tools like typo tolerance, tokenization, and normalization. On the other hand, semantic search is able to return query results that match meaning through the use of vector search.
- Consider "chocolate milk." A semantic search engine will distinguish between "chocolate milk" and "milk chocolate." Though the keywords in the query are the same, the order in which they are written affects the meaning. As humans, we understand that milk chocolate refers to a variety of chocolate, whereas chocolate milk is chocolate-flavored milk.

Semantic search vs. keyword search

- Semantic search offers results based on the ***user's geographical context, the user's past search history, and user intent.***
- Personalization uses the searcher's previous searches and interactions to determine response relevance and rank. Semantic search can also rerank results based on how other users have interacted with the responses it has pulled. For example, when you type "restaurants" into your search engine, it will produce results that are in your area.
- With a better understanding of user intent, semantic search can respond to a query like "Creuset vs. Staub dutch ovens" with content that prioritizes product comparisons because that is the user's intent. Semantic search will recognize the intent behind "best Staub deals" or "Creuset discounts" as intent to purchase and offer responses accordingly.
- Another example is ***predictive text.*** As you type a query into a search bar, it uses semantic search to complete your query and ***suggest relevant search*** terms based on context, common searches, and past search history.

Benefits of semantic search

- **Easier to use for customers**

  Customers might not remember jargon, or recall specific product names. Semantic search enables customers to input vague search queries and get specific results. Customers can also search using a description to discover its name. For example, you can discover a song by searching for the lyrics that you know and find the title.

- Because semantic search interprets meaning by considering intent and context, the experience on the customer side feels more like human interaction.

Benefits of semantic search

- **Concepts are more robust than keywords**

  By matching concepts rather than keywords, semantic search produces more accurate results. Through dimensional embeddings, a vector represents a word as a concept. "Car" is no longer only matched to "car" or "cars," it is also matched to "driver," "insurance," "tires," "electric," "hybrid," and so on because those words are connected to the vector of "car."

- So, semantic search that is vector search-powered expands on the concept of simply matching keywords represented by tokens.

Benefits of semantic search

- **Better for business**

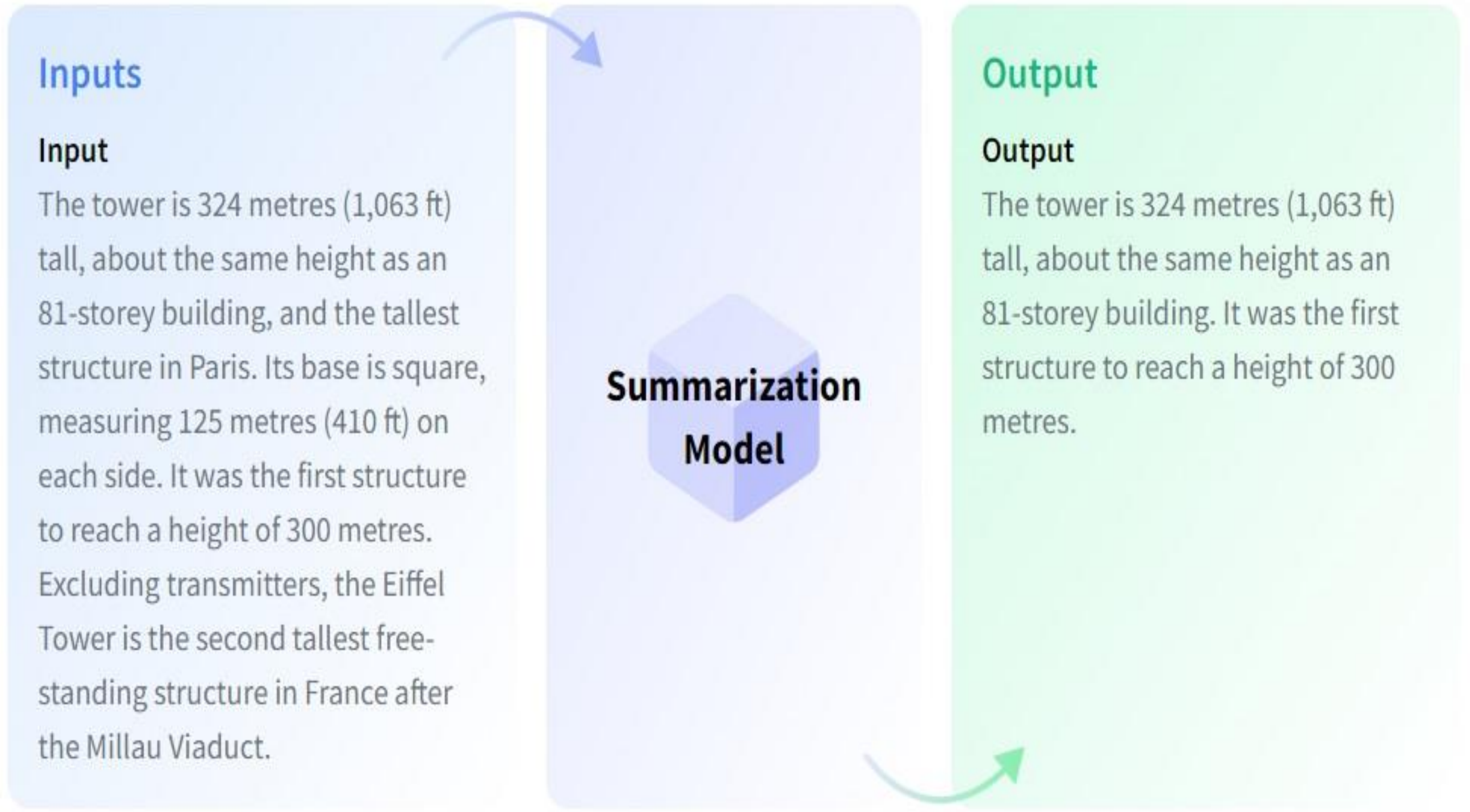
  By understanding user intent, semantic search can boost sales and customer satisfaction. User intent can be informational, transactional, navigational, or commercial. Understanding intent allows a search engine to better meet customer needs. This improves the customer’s relationship with the brand, which is better for business.

Summarization

Summarization

- Text summarization is the technique of creating a short, accurate, and fluent summary of a longer text document.
- Summarization attempts to reduce a section of text to a smaller amount.
- It aims to either remove redundant or irrelevant information, or to draw attention immediately to the most relevant part of a large document.
- Summaries are useful for displaying in results lists.

Ex of Summarization



Inputs

Input

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building, and the tallest structure in Paris. Its base is square, measuring 125 metres (410 ft) on each side. It was the first structure to reach a height of 300 metres. Excluding transmitters, the Eiffel Tower is the second tallest free-standing structure in France after the Millau Viaduct.

Output

Output

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building. It was the first structure to reach a height of 300 metres.

Why automatic text summarization?

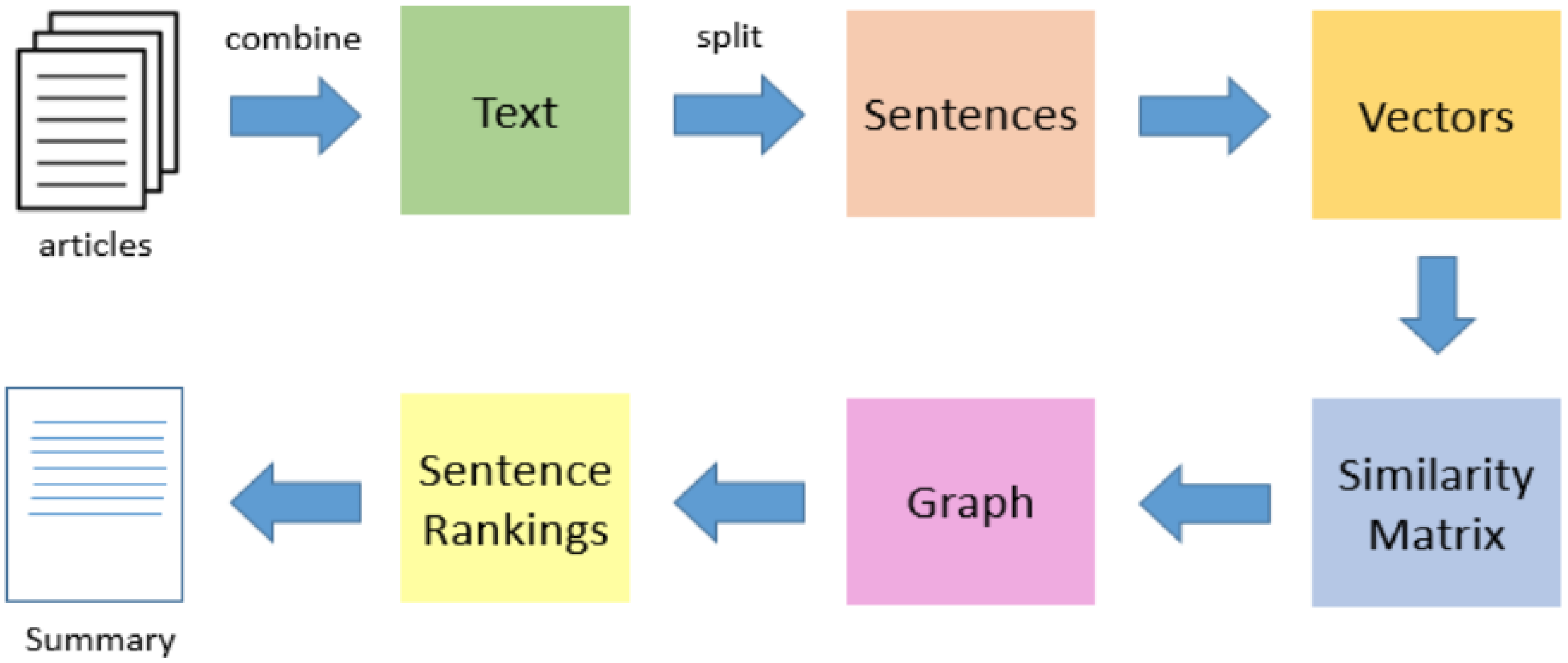
1. Summaries reduce reading time.
2. When researching documents, summaries make the selection process easier.
3. Automatic summarization improves the effectiveness of indexing.
4. Automatice summarization algorithms are less biased than human summarization.
5. Personalized summaries are useful in question-answering systems as they provied personalized information.
6. Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of text documents they are able to process.

Types of Summarization

Summary	Description
Conceptual summary	Sentences that are typical of the document content, which can be from different parts of the document. Use this type of summary to give a general idea of what the document is about
Contextual summary	A conceptual summary, biased to include sentences that are particularly ***relevant to the query terms.*** Use this type of summary to show the sections of the document that are most relevant to the query
Quick summary	The first few sentences of the document. Use this type of summary to give a brief introduction to the document

Text Summarization

- In this approach we build algorithms or programs which will reduce the text size and create a summary of our text data.
- This is called automatic text summarization in machine learning. Text summarization is the process of creating shorter text without removing the semantic structure of text.
-



How does this text summarizer work?

- Trained by machine learning, text summarizer uses the concept of abstractive summarization to summarize a book, an article, or a research paper.
- This summarize tool uses NLP to create novel sentences and generates a summary in which the main idea remains intact.
- It is an advanced-level tool that uses AI for its work. Therefore, the summary produced by this article summarizer tool appears to be flawless and inflow.

Demo - Text summarizer

https://colab.research.google.com/drive/1owXJLMJW1ZKbcthCqy5DKMwiTWQtrMW0?usp=sharing

Extractive Vs Abstractive, Summarization

Auto summarization

- Auto summarization is the process of generating a concise and coherent summary of a longer document or set of documents automatically.
- There are several approaches to auto summarization, including
  - Extractive Summarization
  - Abstractive Summarization:

Approaches in auto summarization

- **Extraction Summarization:**

This approach entails the method to ***extract keywords and phrases from sentences*** and then ***joining*** them to produce a compact meaningful summary.

- **Abstractive Summarization:**

In this summary generator, algorithms are developed in such a way to ***reproduce a long text into a shorter one*** by NLP. It retains its meaning but ***changes the structure of sentences.***

Extractive Vs Abstractive Summarization

ASPECT	EXTRACTIVE SUMMARIZATION	ABSTRACTIVE SUMMARIZATION
Definition	Selects and compiles important sentences or phrases from the original text	Generates a summary using natural language generation techniques
Content	Reproduces content from the original text	May include original content not present in the original text
Output	Sentence or phrase-based	More fluent, human-like language
Input	Limited to the content of the original text	Can incorporate external knowledge or information
Difficulty	Easier to implement	More challenging due to the need for natural language generation
Accuracy	May miss important information	May include irrelevant or incorrect information
Use case	Suitable for summarizing news articles, scientific papers, and other informative texts	Useful for creating headlines, marketing copy, and other creative content

Algorithms used for Extractive Summarization

1. ***Frequency method:*** It find the frequency of all the words in our text data and store the text data and its frequency in a dictionary. After that, we tokenize our text data. The sentences which contain more high frequency words will be kept in our final summary data.

   - Word Frequency Analysis: The technique counts the frequency of each word in the source text. It may use a simple dictionary or data structure to keep track of word occurrences. This step can be implemented with a basic algorithm that iterates through the text and updates the word frequencies.

Algorithms used for Extractive Summarization

2. **TextRank:** An ***unsupervised graph-based algorithm*** that ranks sentences based on their importance using ***PageRank algorithm.***

3. **LexRank:** Another unsupervised graph-based algorithm that measures the centrality of each sentence using ***cosine similarity***.

4. **LSA:** ***Latent Semantic Analysis (LSA)*** is a statistical algorithm that identifies important sentences by analyzing the underlying ***latent semantic structure*** of the text.

Method to calculate cosine similarity

1. ***Vector Representation:*** To apply cosine similarity, you first need to represent documents as ***vectors.*** This is often done using techniques like ***TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings.***

2. ***Vector Normalization:*** Normalize the vectors. ***This means dividing each vector by its magnitude*** (also called the vector's length). This step ensures that the length of the vectors does not affect the cosine similarity calculation. ***Normalized vectors have a length of 1.***

Method to calculate cosine similarity

3. ***Cosine Similarity Calculation:*** To calculate the cosine similarity between two vectors A and B, use the following formula:

4. Cosine Similarity ***(A, B) = (A • B) / (||A|| * ||B||)***

   - (A • B) is the dot product of vectors A and B, which is the sum of the pairwise products of their components.
   - ||A|| and ||B|| represent the magnitudes (or lengths) of vectors A and B, respectively. These magnitudes are the square roots of the sum of the squares of the individual vector components.
   - ***Interpretation:*** The resulting cosine similarity value ranges from -1 (perfectly dissimilar) to 1 (perfectly similar). ***A value of 0 indicates no similarity.***

Algorithms used for Abstractive Summarization

1. ***Sequence-to-Sequence models***
2. ***Pointer-Generator Networks***
3. ***Transformer-based models***

Sequence-to-Sequence models

- A Seq2Seq model is a type of neural network architecture used NLP and other sequence-related tasks. It is designed to handle sequences of data, where the length of the input and output sequences can vary. Sequence-to-sequence models, often implemented using ***RNNs*** or more modern variants like ***LSTMs and GRUs***, utilize an encoder-decoder architecture.

- The key characteristic of a Seq2Seq model is its encoder-decoder structure:

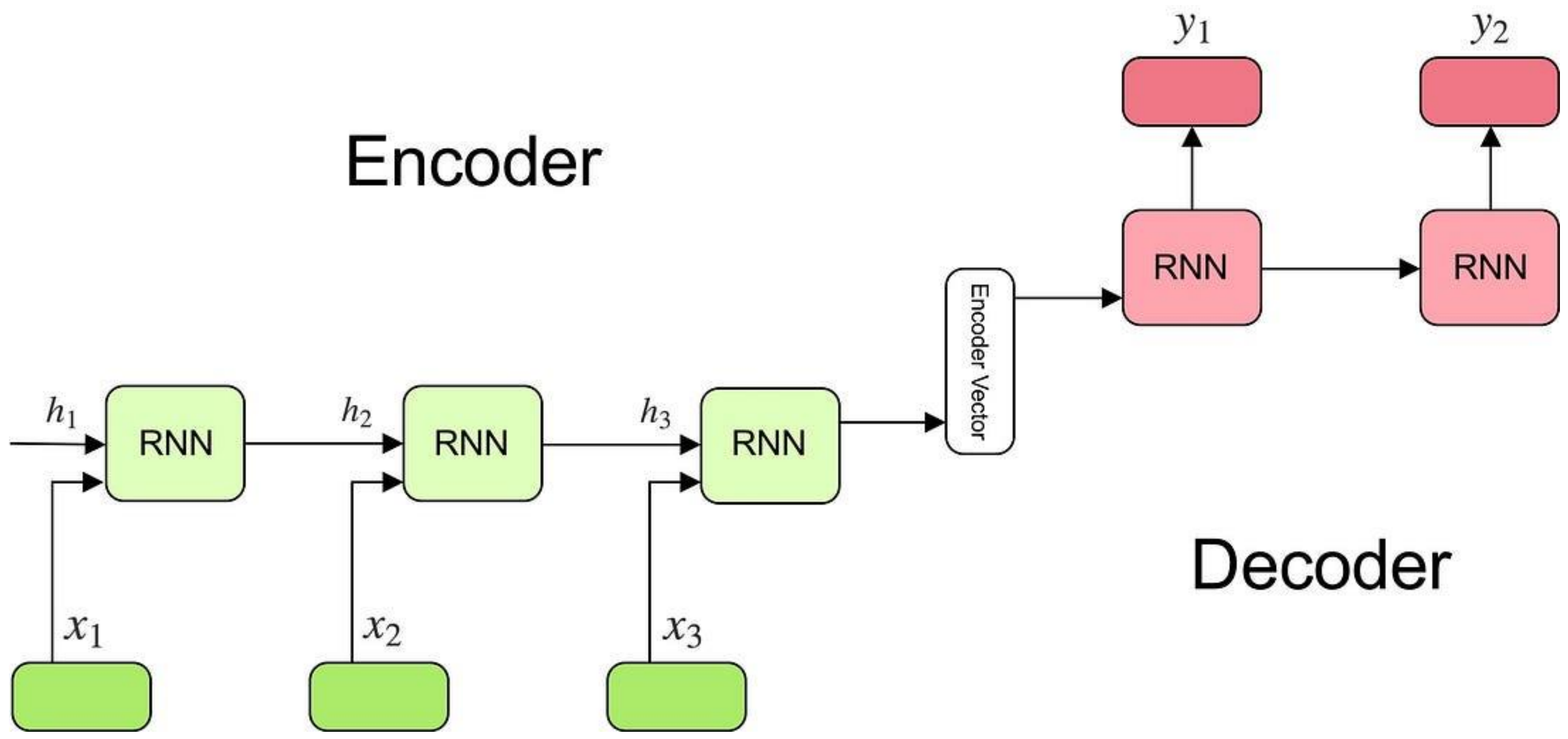
  1. ***Encoder (process inputs):*** The encoder component takes an input sequence and processes it step by step, usually from left to right. It transforms the input sequence into a fixed-length vector called the "context" or "thought" vector. This context vector represents the entire input sequence in a compressed form and captures the important information from the input.

  2. ***Decoder (generate output):*** The decoder component takes the context vector produced by the encoder and generates an output sequence, often one step at a time. The decoder can be autoregressive, meaning it generates each output token while conditioning on the previously generated tokens.

Sequence-to-Sequence models

- Seq2Seq models are particularly versatile and have been used for a wide range of ***sequence-based tasks***, including machine translation, text summarization, and speech recognition.
- ***Machine Translation:*** Seq2Seq models are known for their effectiveness in translating text from one language to another. The encoder processes the input text in the source language, and the decoder generates the translated text in the target language.
- ***Text Summarization:*** In abstractive text summarization, the encoder processes the input document, and the decoder generates a concise summary of the content.
- ***Speech Recognition:*** Seq2Seq models can convert spoken language (audio) into text. The encoder processes the audio waveform, and the decoder generates a corresponding textual representation of the spoken words.

Encoder-decoder sequence to sequence model



https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346
https://www.cloudskillsboost.google/course_sessions/5521175/video/383848

Pointer-Generator Networks

- Pointer-Pointer-Generator Networks are a neural architecture, built on ***RNNs or transformers***, that processes input and generates output.

- Key characteristics and components of Pointer-Generator Networks:

  1. ***Hybrid Approach:*** Pointer-Generator Networks combine the characteristics of both ***extractive and abstractive summarization.*** They have the ability to "copy" words or phrases directly from the source text (extractive) and also generate novel words and phrases to form a coherent and informative summary (abstractive).

  2. ***Encoder-Decoder Architecture:*** Like many other sequence-to-sequence models, Pointer-Generator Networks consist of an encoder and a decoder. The encoder processes the input text, and the decoder generates the summary. ***The key difference lies in how the decoder operates.***

  3. ***Pointer Mechanism:*** The decoder includes a ***pointer mechanism,*** which allows it to select and copy words from the input text into the summary. This is especially valuable for retaining specific details, named entities, or rare words from the source text, ensuring that the summary remains faithful to the source.

Pointer-Generator Networks

4. ***Coverage Mechanism:*** Pointer-Generator Networks often include a coverage mechanism that keeps track of which words in the ***source text have been attended to by the decoder.*** This helps prevent repetition of words and encourages the generation of more diverse and coherent summaries.

5. ***Word Generation:*** In addition to copying words from the source text, the decoder generates new words and phrases when necessary to create a coherent summary. It does so by learning how to predict which words to generate based on the context.

6. ***Training Data:*** Training Pointer-Generator Networks typically requires access to parallel data where the source text is matched with its corresponding summary. This allows the model to learn both extraction and generation behaviors.

Pointer-Generator Networks have demonstrated success in various NLP tasks, especially in text summarization, where they can generate informative and coherent summaries that include both extracted content from the source text and abstractive content that captures the essence of the text. This hybrid approach has helped o

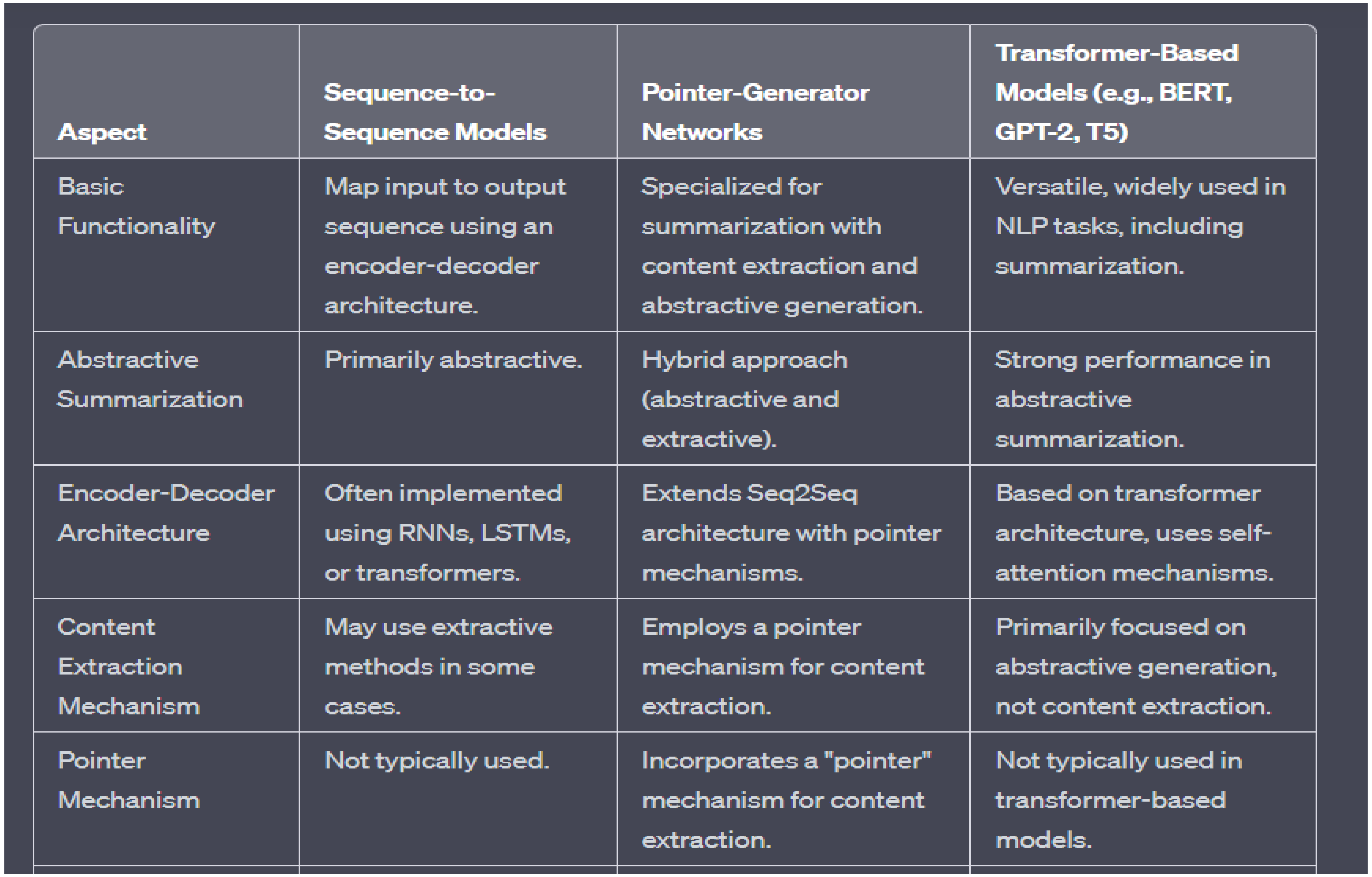
Transformer based Networks

- Transformer is a neural network architecture.
- Unlike traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs), transformers rely on ***self-attention mechanisms*** to process sequences of data.
- These models have shown promising results for abstractive summarization tasks, such as ***BERT, GPT-2, T5***, etc.
- ***Self-Attention Mechanism:*** The self-attention mechanism allows the model to consider the ***relationships and dependencies between all words or tokens in a sequence simultaneously.*** This enables transformers to capture complex contextual information, including long-range dependencies, which is crucial for understanding the meaning of text.
- ***Contextual Relationships:*** Transformers can effectively capture contextual relationships between words in the text. For example, when generating a summary, they consider not only the current word but also the relevant words that provide context and meaning to the current word. This results in more coherent and contextually relevant summaries.

Transformer based Networks

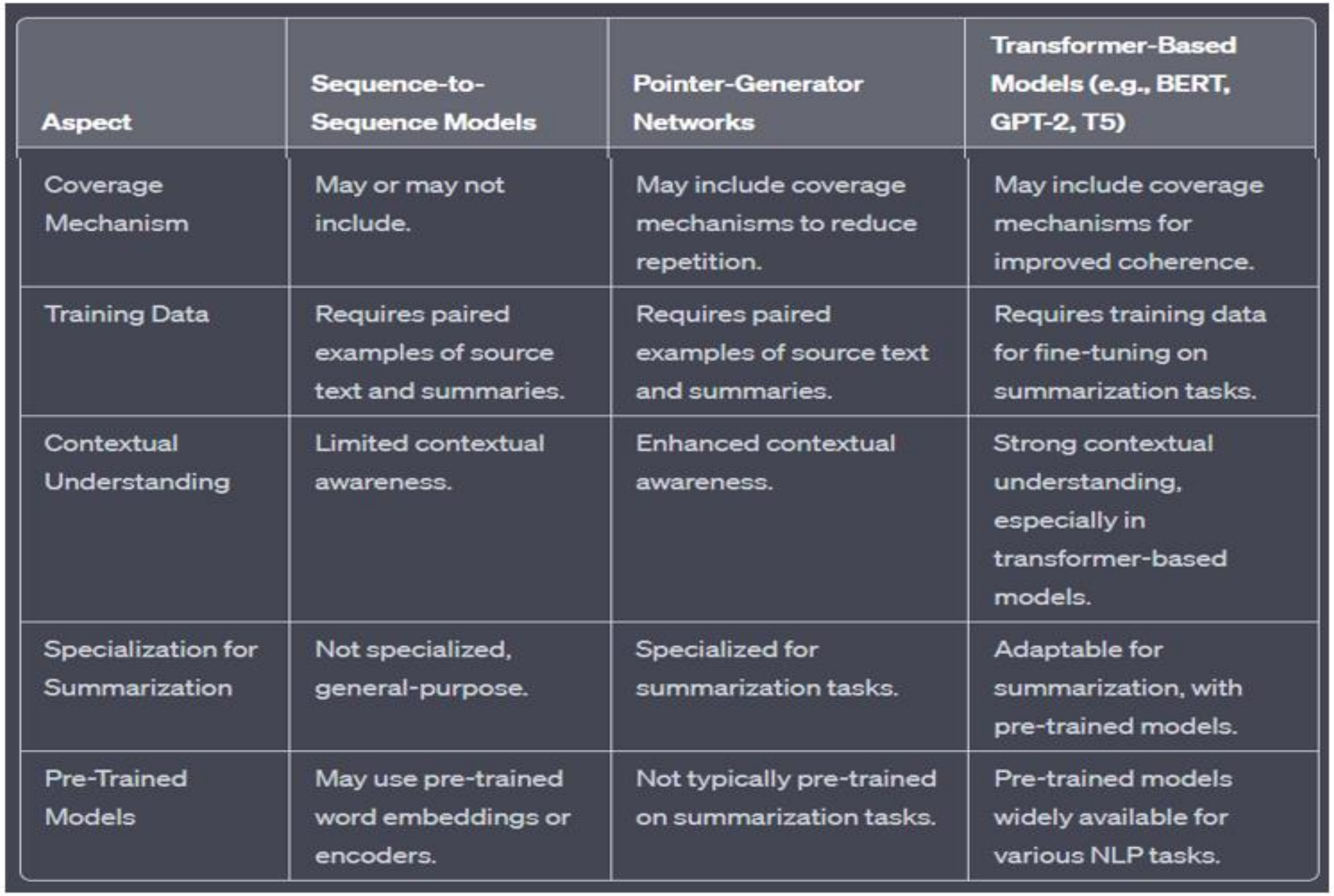
- Transformers, such as BERT, GPT-2, and T5, excel in abstractive summarization due to their advanced self-attention mechanisms and deep learning capabilities.
- ***BERT (Bidirectional Encoder Representations from Transformers):*** BERT is a pre-trained transformer model that captures bidirectional contextual information. It's used as a foundation for various NLP tasks, including abstractive summarization. BERT-based models can understand the context of words and generate summaries that are contextually relevant.
- ***GPT-2 (Generative Pre-trained Transformer 2):*** GPT-2 is a generative model that uses transformers to generate human-like text. It's capable of abstractive summarization by generating coherent and contextually relevant summaries. GPT-2 has been applied to a wide range of NLP tasks, including summarization.
- ***T5 (Text-to-Text Transfer Transformer):*** T5 is a model that frames all NLP tasks as text-to-text tasks. It has shown remarkable results in various tasks, including summarization. By treating summarization as a text-to-text task, T5 can generate abstractive summaries effectively.

Sequence-to-Sequence vs Pointer-Generator vs Transformer-based models



Aspect	Sequence-to-Sequence Models	Pointer-Generator Networks	Transformer-Based Models (e.g., BERT, GPT-2, T5)
Basic Functionality	Map input to output sequence using an encoder-decoder architecture.	Specialized for summarization with content extraction and abstractive generation.	Versatile, widely used in NLP tasks, including summarization.
Abstractive Summarization	Primarily abstractive.	Hybrid approach (abstractive and extractive).	Strong performance in abstractive summarization.
Encoder-Decoder Architecture	Often implemented using RNNs, LSTMs, or transformers.	Extends Seq2Seq architecture with pointer mechanisms.	Based on transformer architecture, uses self-attention mechanisms.
Content Extraction Mechanism	May use extractive methods in some cases.	Employs a pointer mechanism for content extraction.	Primarily focused on abstractive generation, not content extraction.
Pointer Mechanism	Not typically used.	Incorporates a "pointer" mechanism for content extraction.	Not typically used in transformer-based models.

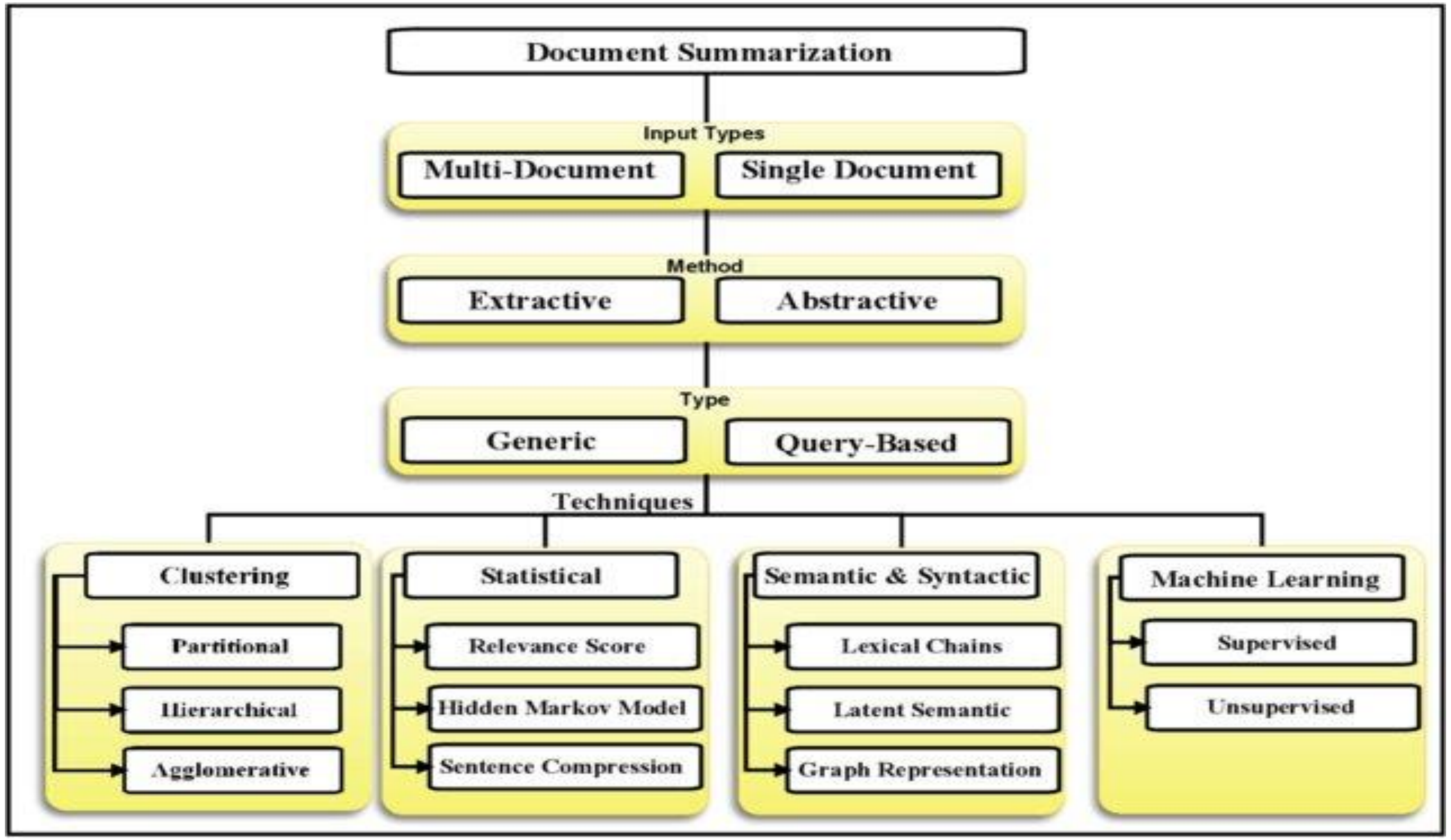
Sequence-to-Sequence vs Pointer-Generator vs Transformer-based models



Aspect	Sequence-to-Sequence Models	Pointer-Generator Networks	Transformer-Based Models (e.g., BERT, GPT-2, T5)
Coverage Mechanism	May or may not include.	May include coverage mechanisms to reduce repetition.	May include coverage mechanisms for improved coherence.
Training Data	Requires paired examples of source text and summaries.	Requires paired examples of source text and summaries.	Requires training data for fine-tuning on summarization tasks.
Contextual Understanding	Limited contextual awareness.	Enhanced contextual awareness.	Strong contextual understanding, especially in transformer-based models.
Specialization for Summarization	Not specialized, general-purpose.	Specialized for summarization tasks.	Adaptable for summarization, with pre-trained models.
Pre-Trained Models	May use pre-trained word embeddings or encoders.	Not typically pre-trained on summarization tasks.	Pre-trained models widely available for various NLP tasks.

Single and Multi-document Summarization – Question Answering

Single and Multi-document Summarization

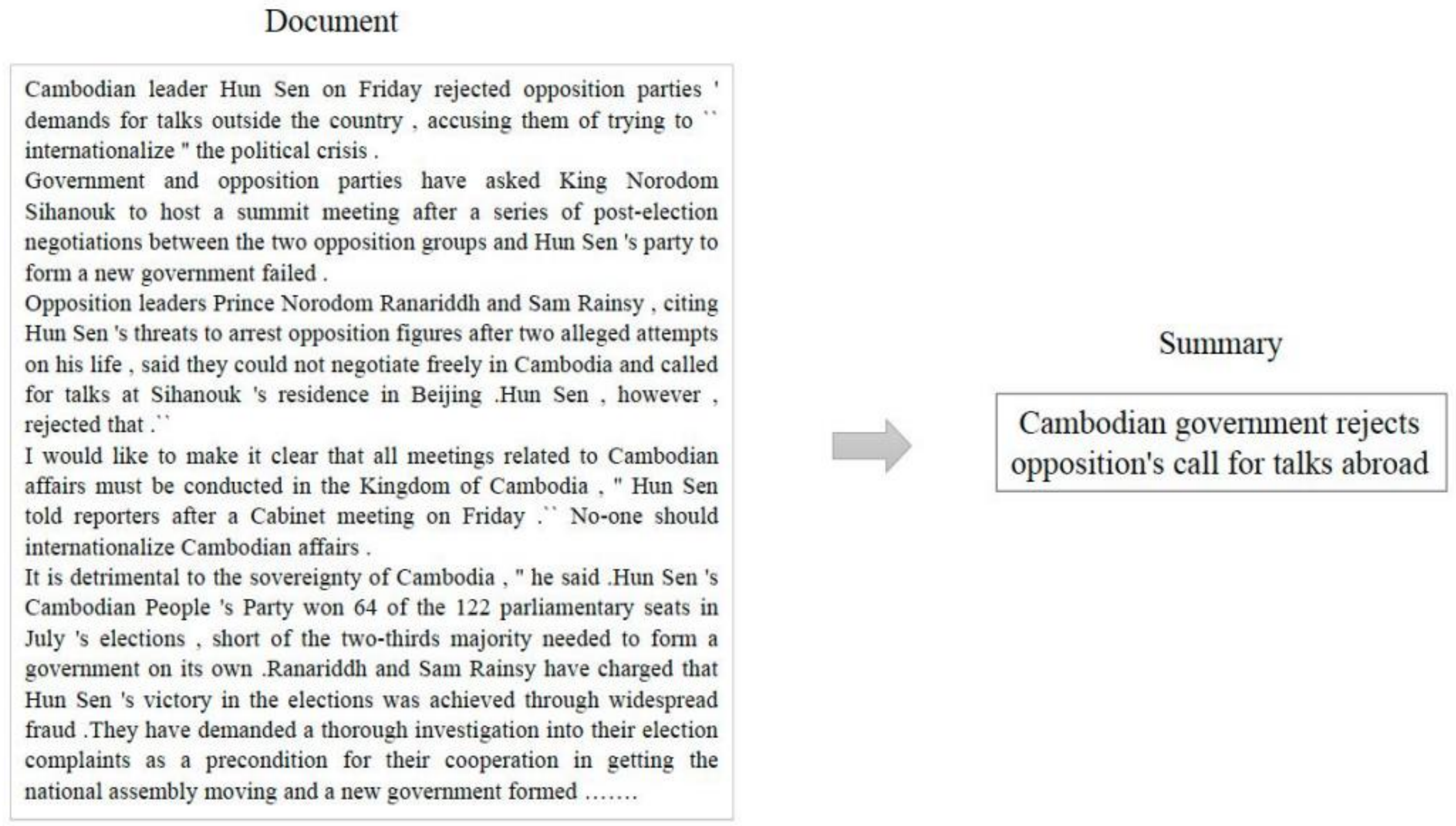


Single and Multi-document Summarization

- ***Single-Document Summarization:***
- Single-document summarization, as the name suggests, focuses on summarizing the content of a single document or text. It aims to distill the **most important information or key points from a single source document.** This technique is commonly used for various purposes:

1. **Information Retrieval:** Single-document summarization helps users quickly grasp the main ideas of an article, news story, research paper, or any other single text document.
2. **Content Abstraction:** It generates a concise summary of the source text, often in a few sentences or paragraphs, while retaining the core meaning.
3. **Content Presentation:** Single-document summarization is employed to create abstracts, bullet points, or concise introductions for documents to make them more accessible and informative.
4. **Search Engine Snippets:** Search engines often use single-document summarization to provide snippet summaries in search results, helping users decide which pages to visit.

Ex of Single document Summarization


Document

Cambodian leader Hun Sen on Friday rejected opposition parties ' demands for talks outside the country , accusing them of trying to `` internationalize " the political crisis .
Government and opposition parties have asked King Norodom Sihanouk to host a summit meeting after a series of post-election negotiations between the two opposition groups and Hun Sen 's party to form a new government failed .
Opposition leaders Prince Norodom Ranariddh and Sam Rainsy , citing Hun Sen 's threats to arrest opposition figures after two alleged attempts on his life , said they could not negotiate freely in Cambodia and called for talks at Sihanouk 's residence in Beijing .Hun Sen , however , rejected that .``
I would like to make it clear that all meetings related to Cambodian affairs must be conducted in the Kingdom of Cambodia , " Hun Sen told reporters after a Cabinet meeting on Friday .`` No-one should internationalize Cambodian affairs .
It is detrimental to the sovereignty of Cambodia , " he said .Hun Sen 's Cambodian People 's Party won 64 of the 122 parliamentary seats in July 's elections , short of the two-thirds majority needed to form a government on its own .Ranariddh and Sam Rainsy have charged that Hun Sen 's victory in the elections was achieved through widespread fraud .They have demanded a thorough investigation into their election complaints as a precondition for their cooperation in getting the national assembly moving and a new government formed ....

Summary

Cambodian government rejects opposition's call for talks abroad

Single document summarization

- Single document summarization involves generating a condensed version of a single text document while retaining its main ideas and key details. This is commonly used in news articles, blog posts, and research papers.
- ***Extractive Summarization:***
  - Example: Given a news article about a recent political event, an extractive summarization model might select sentences like "The event took place on Monday" and "The president addressed the nation".
- ***Abstractive Summarization:***
  - Example: Using abstractive summarization, the model could generate a summary like "The president addressed the nation on Monday about the recent political event".
- ***Hybrid Approaches:***
  - Example: A hybrid model might extract the sentences "The event took place on Monday" and then rephrase it as "On Monday, the event occurred".

CHALLENGES

- ***Preservation of Meaning:*** Abstractive summarization may struggle with ensuring that the generated summary accurately conveys the original meaning.

- ***Fluency and Coherence:*** Achieving natural-sounding summaries that flow well can be a challenge for abstractive methods.

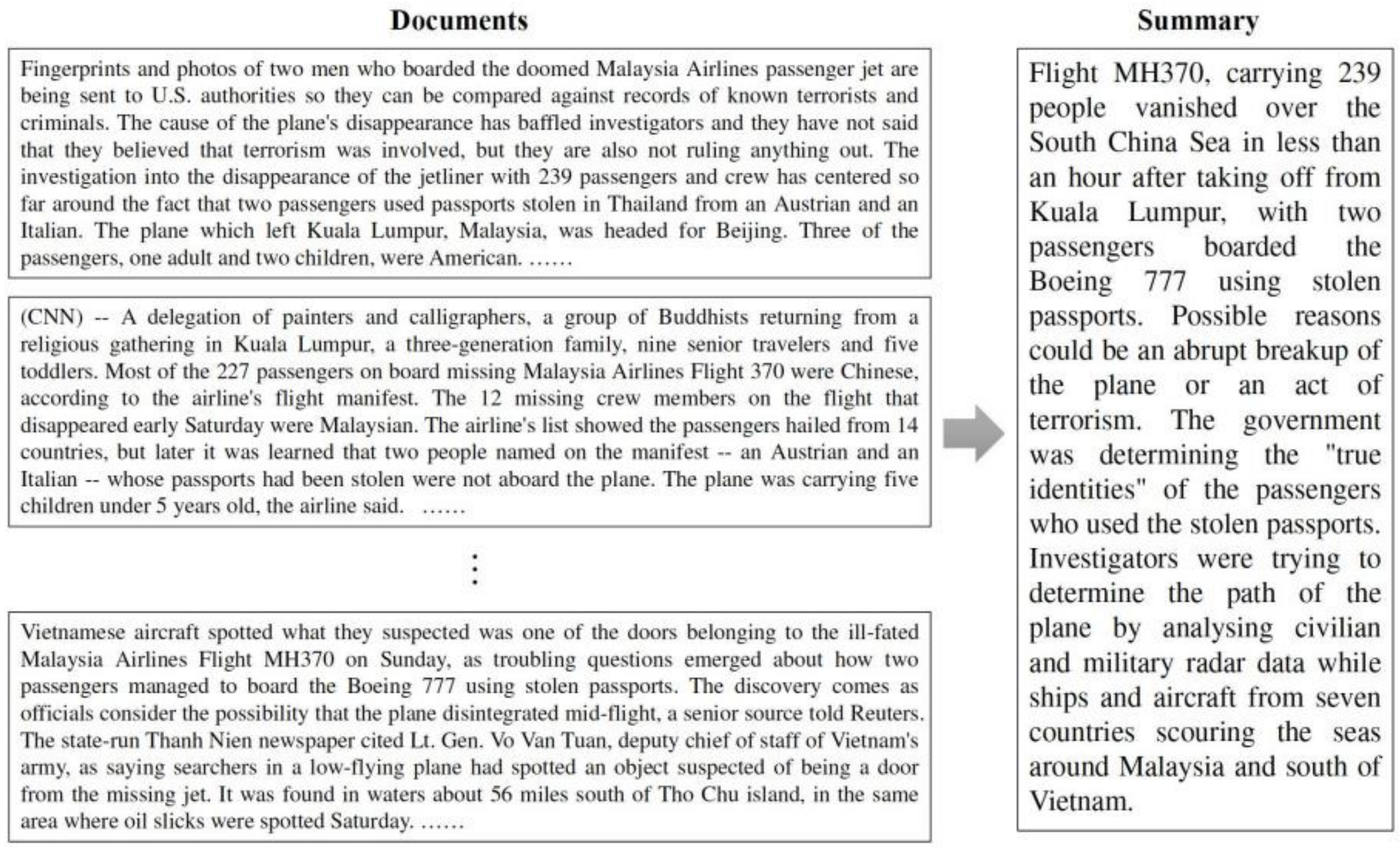
- ***Handling Specific Domains:*** In specialized domains, adapting summarization techniques to understand and use domain-specific terminology is complex.

Multi-document Summarization

- ***Multi-Document Summarization:***
- Multi-document summarization, in contrast, involves summarizing the content of multiple source documents. The objective is to **aggregate, distill, and merge information** from a collection of texts, such as a set of articles on a particular topic, multiple news reports about the same event, or a corpus of documents related to a specific subject. Multi-document summarization is used for several purposes:

1. **Aggregating Information:** It aims to provide a comprehensive summary that captures the most important details from multiple source documents, ensuring that the main points from all documents are included.
2. **Redundancy Reduction:** This technique addresses the issue of redundancy, ensuring that repetitive information found in multiple documents is summarized efficiently.
3. **Content Fusion:** Information from different sources is often merged to create a coherent and informative summary that provides a more comprehensive view of the topic.
4. **Research and Analysis:** Multi-document summarization is valuable for researchers, journalists, analysts, and anyone who needs to quickly understand and compare information from multiple sources.

Ex of Multi document Summarization



Documents

Fingerprints and photos of two men who boarded the doomed Malaysia Airlines passenger jet are being sent to U.S. authorities so they can be compared against records of known terrorists and criminals. The cause of the plane's disappearance has baffled investigators and they have not said that they believed that terrorism was involved, but they are also not ruling anything out. The investigation into the disappearance of the jetliner with 239 passengers and crew has centered so far around the fact that two passengers used passports stolen in Thailand from an Austrian and an Italian. The plane which left Kuala Lumpur, Malaysia, was headed for Beijing. Three of the passengers, one adult and two children, were American. ……

(CNN) -- A delegation of painters and calligraphers, a group of Buddhists returning from a religious gathering in Kuala Lumpur, a three-generation family, nine senior travelers and five toddlers. Most of the 227 passengers on board missing Malaysia Airlines Flight 370 were Chinese, according to the airline's flight manifest. The 12 missing crew members on the flight that disappeared early Saturday were Malaysian. The airline's list showed the passengers hailed from 14 countries, but later it was learned that two people named on the manifest -- an Austrian and an Italian -- whose passports had been stolen were not aboard the plane. The plane was carrying five children under 5 years old, the airline said. ……

⋮

Vietnamese aircraft spotted what they suspected was one of the doors belonging to the ill-fated Malaysia Airlines Flight MH370 on Sunday, as troubling questions emerged about how two passengers managed to board the Boeing 777 using stolen passports. The discovery comes as officials consider the possibility that the plane disintegrated mid-flight, a senior source told Reuters. The state-run Thanh Nien newspaper cited Lt. Gen. Vo Van Tuan, deputy chief of staff of Vietnam's army, as saying searchers in a low-flying plane had spotted an object suspected of being a door from the missing jet. It was found in waters about 56 miles south of Tho Chu island, in the same area where oil slicks were spotted Saturday. ……

Summary

Flight MH370, carrying 239 people vanished over the South China Sea in less than an hour after taking off from Kuala Lumpur, with two passengers boarded the Boeing 777 using stolen passports. Possible reasons could be an abrupt breakup of the plane or an act of terrorism. The government was determining the "true identities" of the passengers who used the stolen passports. Investigators were trying to determine the path of the plane by analysing civilian and military radar data while ships and aircraft from seven countries scouring the seas around Malaysia and south of Vietnam.

Figure 2: Multi-document summarization for the topic "Malaysia Airlines Disappearance".

Multiple Document Summarization

- Multiple document summarization involves condensing information from a collection of related documents. This is valuable for scenarios like news coverage of a major event, where multiple articles provide different perspectives.
- ***Cluster-based Summarization:***
  - Example: Given a set of news articles about a sports event, the system clusters articles by teams involved. It then generates summaries for each cluster, offering various viewpoints.
- ***Graph-based Summarization:***
  - Example: Documents are represented as nodes in a graph, with edges denoting similarity. Sentences with high centrality in the graph are selected for the summary, ensuring coverage of the main points.
- ***Centroid-based Summarization:***
  - Example: For a collection of product reviews, the centroid (average) of TF-IDF vectors is calculated. Sentences closest to this centroid, representing common sentiments, are chosen for the summary.

Question and Answering

Question answering (QA) is a subfield of Natural Language Processing (NLP) that focuses on developing systems capable of understanding and generating responses to questions posed in natural language. QA systems can be categorized into two main types:

Retrieval-based QA:

- In retrieval-based QA, the system searches for predefined answers from a database or a set of documents. When a question is asked, the system matches it with similar questions or extracts relevant information from the database to provide an answer.

• **Example**: A frequently asked questions (FAQ) chatbot that matches user questions to predefined answers based on keywords or similarity.

Generative-based QA:

- Generative-based QA models generate answers from scratch based on the understanding of the question. These models employ techniques like natural language understanding, context analysis, and language generation to provide responses.

• **Example**: A chatbot that uses a language model to understand the intent of the question and generate a response in natural language.

Example

User: "What is the capital of France?"

Retrieval-based QA:

System: "The capital of France is Paris."

Generative-based QA:

System: "The capital of France is Paris."

QA systems can be applied in various domains, including customer support, information retrieval, educational platforms, and more. They can range from simple rule-based systems to advanced machine learning models like transformers.

QA systems play a crucial role in automating responses to common queries, enhancing user experiences, and improving the efficiency of customer service operations in a wide range of applications.

Machine Translation

Machine Translation (MT)

- ***Machine translation or MT, automatically translates text from one natural language into another.***
- One of the most notable advantages of machine translation is its ***ability to rapidly translate large volumes of text.***
- Many of us were introduced to machine translation when Google launched its translation service.
- However, this technology has come a long way since the mid-20th century.
- Research and development in machine translation, or MT, began as early as the 1950s, primarily in the United States."

How does it works?

- ***Machine translation works by using computer software to translate the text from one source language to another target language.***
- At its core, machine translation is a process that replaces individual words or phrases in one natural language with their equivalents in another language. This fundamental approach is based on creating a mapping between words, phrases, or sentences in the source language and their corresponding translations in the target language.
- Initially, MT systems performed straightforward word-to-word replacements. For example, if the source language has the word "apple," it might be replaced with the word "pomme" in French. This simple substitution of atomic words formed the foundation of early MT systems.

How does it works?

- As the field advanced, more sophisticated techniques emerged. Machine translation systems started to use large corpora (collections of text) to analyze and understand patterns in language. This allowed for more complex translations that considered contextual factors such as idiomatic expressions, sentence structure, and word order.
- MT systems have evolved to handle linguistic challenges, including variations in phonetic typology (sound patterns), recognizing and translating idioms (phrases with non-literal meanings), and addressing linguistic anomalies or exceptions. These improvements have led to more accurate and contextually appropriate translations.

How does it works?

- While MT has made significant progress, there is still a gap between the capabilities of machines and human translators. Human translators possess cultural and contextual knowledge that can be challenging for machines to replicate accurately. However, ongoing advancements in artificial intelligence and machine learning suggest that MT systems will continue to improve, possibly reaching a level where they can perform translations that closely resemble human translation in quality and nuance.

4 Types of Machine Translation

1. Statistical Machine Translation or SMT

- It works by alluding to statistical models that depend on the investigation of huge volumes of bilingual content. It expects to decide the correspondence between a word from the source language and a word from the objective language. A genuine illustration of this is Google Translate.
- Presently, SMT is extraordinary for basic translation, however its most noteworthy disadvantage is that it doesn't factor in context, which implies translation can regularly be wrong or you can say, don't expect great quality translation. There are several types of statistical-based machine translation models which are: Hierarchical phrase-based translation, Syntax-based translation, Phrase-based translation, Word-based translation.

- https://www.analyticssteps.com/blogs/4-types-machine-translation-nlp

4 Types of Machine Translation

2. *Rule based MT*

- RBMT basically translates the basics of grammatical rules. It directs a grammatical examination of the source language and the objective language to create the translated sentence. But, RBMT requires broad editing, and its substantial reliance on dictionaries implies that proficiency is accomplished after a significant period.

3. *Hybrid Machine Translation*

- HMT, as the term demonstrates, is a mix of RBMT and SMT. It uses a translation memory, making it unquestionably more successful regarding quality. Nevertheless, even HMT has a lot of downsides, the biggest of which is the requirement for enormous editing, and human translators will also be needed. There are several approaches to HMT like multi-engine, statistical rule generation, multi-pass, and confidence-based.

4 Types of Machine Translation

4. Neural Machine Translation or NMT

- NMT is a type of machine translation that relies upon neural network models (based on the human brain) to build statistical models with the end goal of translation. The essential advantage of NMT is that it gives a solitary system that can be prepared to unravel the source and target text. Subsequently, it doesn't rely upon specific systems that are regular to other machine translation systems, particularly SMT.