### **Prepared by:**

## 21PA1A54B1

## 21PA1A54A7

## 21PA1A54B7

## 21PA1A5472

## 21PA1A5489

## 21PA1A54C1

### 

**TEXTBOOKS:**

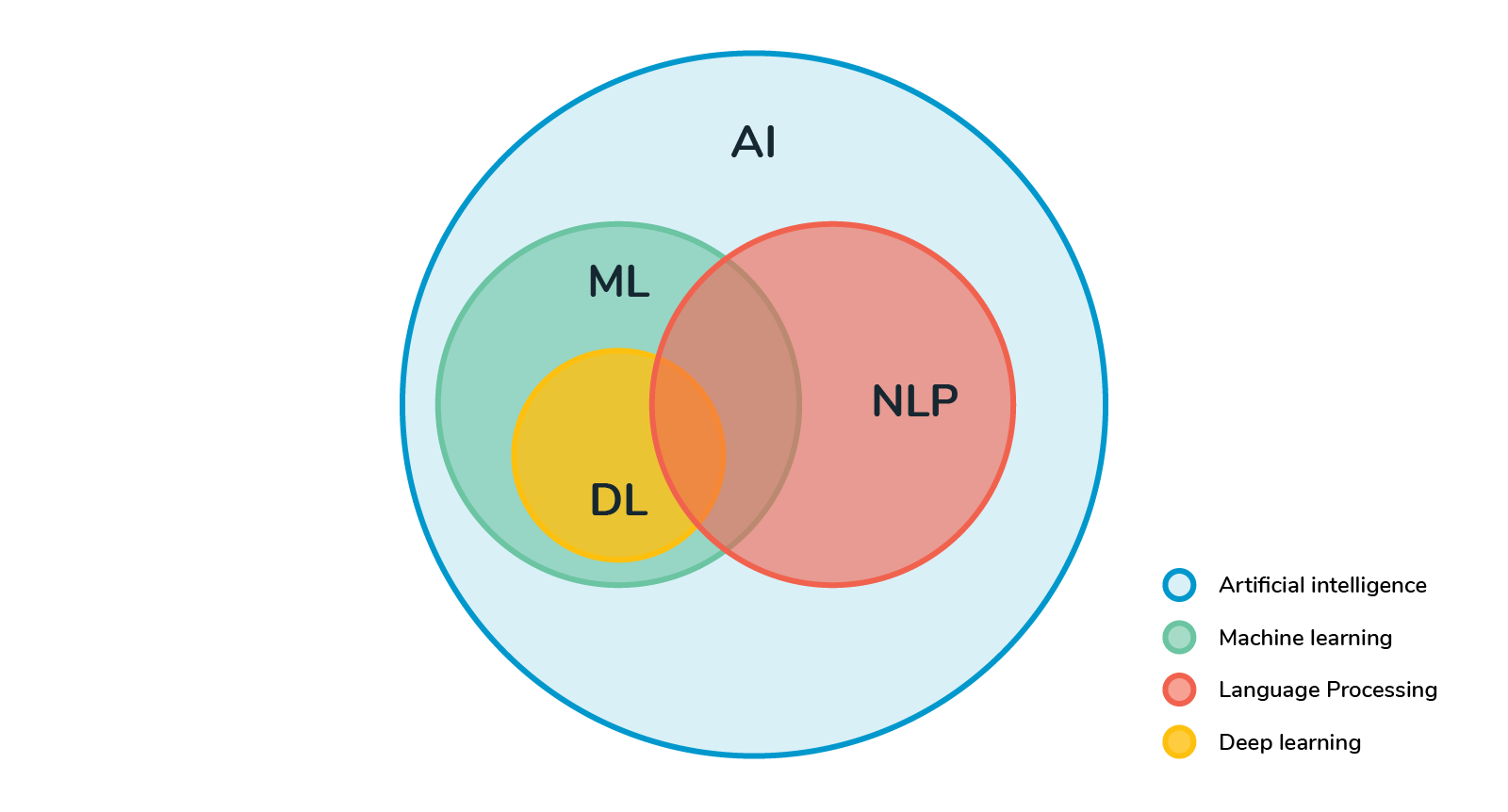
1. [Daniel Jurafsky, James H. Martin―Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication](https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf)
2. [Steven Bird, Ewan Klein and Edward Loper, ―Natural Language Processing with Python, First Edition, OReilly Media, 2009.](https://www.foo.be/cours/dess-20122013/b/Natural%20Language%20Processing%20with%20Python%20-%20O'Reilly2009.pdf)

### 

**UNIT - I**

****

# **What is Natural Language Processing**



Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modelling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarise large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

**NLP tasks:**

Several NLP tasks break down human text and voice data in ways that help the computer make sense of what it's ingesting. Some of these tasks include the following:

1. **Speech recognition**, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.
2. **Part of speech tagging,** also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies ‘make’ as a verb in ‘I can make a paper plane,’ and as a noun in ‘What make of car do you own?’
3. **Word sense disambiguation** is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that determines the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).
4. **Named entity recognition**, or NEM, identifies words or phrases as useful entities. NEM identifies ‘Kentucky’ as a location or ‘Fred’ as a man's name.
5. **Coreference** resolution is the task of identifying if and when two words refer to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., ‘she’ = ‘Mary’), but it can also involve identifying a metaphor or an idiom in the text (e.g., an instance in which 'bear' isn't an animal but a large hairy person).
6. **Sentiment analysis** attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.
7. **Natural language generation** is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

**NLP use cases**

Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples:

1. **Spam detection**: You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelt company names, and more. Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this doesn’t match your email experience).
2. **Machine translation:** Google Translate is an example of widely available NLP technology at work. Truly useful machine translation involves more than replacing words in one language with words of another. Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language. Machine translation tools are making good progress in terms of accuracy. A great way to test any machine translation tool is to translate text to one language and then back to the original. An oft-cited classic example: Not long ago, translating “*The spirit is willing but the flesh is weak”* from English to Russian and back yielded “*The vodka is good but the meat is rotten*.” Today, the result is “*The spirit desires, but the flesh is weak*,” which isn’t perfect, but inspires much more confidence in the English-to-Russian translation.
3. **Virtual agents and chatbots:** Virtual agents such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with appropriate action or helpful comments. Chatbots perform the same magic in response to typed text entries. The best of these also learn to recognize contextual clues about human requests and use them to provide even better responses or options over time. The next enhancement for these applications is question answering, the ability to respond to our questions—anticipated or not—with relevant and helpful answers in their own words.
4. **Social media sentiment analysis:** NLP has become an essential business tool for uncovering hidden data insights from social media channels. Sentiment analysis can analyse language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events–information companies can use in product designs, advertising campaigns, and more.
5. **Text summarization:** Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

## 

## 

# **Origins of NLP**

1. Neuro-Linguistic Programming (NLP) originated as a therapeutic approach but evolved for broader applications, including private change, interpersonal communications, persuasion, business communication, management training, sales, sports, coaching, team building, speechmaking, negotiation, and communication.
2. NLP training began in the early 1970s when Richard Bandler, a psychology student, met Dr. John Grinder, a linguistics professor. Bandler, influenced by programming and linguistics, aimed to understand and model successful therapeutic techniques.
3. Bandler modelled the work of therapists Virginia Satir and Fritz Perls, focusing on gestalt therapy principles and language structures. The goal was to define techniques and skills used by exceptional therapists.
4. Bandler and Grinder analysed the behaviour, writings, and recordings of Satir and Perls to identify patterns that led to excellence in therapy sessions.
5. NLP is characterised by its pragmatic approach—Bandler and Grinder focused on what worked and studied various influential communicators, including Gregory Bateson, Milton Erickson, and Noam Chomsky.
6. The early NLP books, "The Structure of Magic, Vol I & II," published in 1975 and 1976, identified language patterns and characteristics of effective therapists.
7. Bandler and Grinder expanded their studies to include Milton Erickson's techniques, particularly his conversational hypnosis, which became a central aspect of NLP known as the "Milton Model."
8. Other contributors, including Robert Dilts, Leslie Cameron Bandler, Judith DeLozier, and David Gordon, played crucial roles in expanding and developing NLP beyond the work of Bandler and Grinder.
9. Tony Robbins, a prominent figure in the personality development industry, began his career as an NLP Trainer, working with Richard Bandler.

# **Language and Knowledge**

#### **What in language……?**

* "Language" refers to human language, which is a system of communication used by humans to express thoughts, ideas, and emotions. NLP is a subfield of artificial intelligence (AI) that focuses on the interaction between computers and human language.
* The goal of NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant.
* Language in NLP encompasses various aspects, including syntax, semantics, and pragmatics:
* **Syntax**:
* “This refers to the structure of sentences and the rules governing the arrangement of words in a language.”
* **Semantics:**
* Semantics is concerned with the meaning of words, phrases, and sentences. NLP systems aim to understand the meaning of language and, in some cases, derive the intended meaning from context.
* **Pragmatics:**
* Pragmatics involves understanding language in context. It considers the social and cultural aspects of communication, as well as the implied meaning that may not be explicitly stated. Pragmatic understanding is essential for interpreting language in a way that aligns with human communication.

**2. What is Knowledge?**

**Knowledge in NLP:**

"Knowledge" refers to the understanding and representation of information extracted from text or other linguistic data.

1. **Phonetic and Phonological Knowledge**

* Phonetics is the study of language at the level of sounds while phonology is the study of the combination of sounds into organized units of speech.
* Phonetic and Phonological knowledge is essential for speech-based systems as they deal with how words are related to the sounds that realize them.

1. **Morphological Knowledge**

* Morphology concerns word-formation.
* It is a study of the patterns of formation of words by the combination of sounds into minimal distinctive units of meaning called morphemes.Morphological Knowledge concerns how words are constructed from morphemes.

1. **Syntactic Knowledge:**

* The syntax is the level at which we study how words combine to form phrases, phrases combine to form clauses and clauses join to make sentences.
* The syntactic analysis concerns sentence formation.It deals with how words can be put together to form correct sentences.

1. **Semantic Knowledge**

* It concerns the meaning of the words and sentences.
* Defining the meaning of a sentence is very difficult due to the ambiguities involved.

1. **Pragmatic Knowledge:**

* Pragmatics is the extension of the meanings or semantics.
* Pragmatics deals with the contextual aspects of meaning in particular situations.
* It concerns how sentences are used in different situations.

1. **Discourse Knowledge:**

* Discourse concerns connected sentences. It includes the study of chunks of language which are bigger than a single sentence.
* Discourse language concerns inter-sentential links that is how the immediately preceding sentences affect the interpretation of the next sentence.
* Discourse language is important for interpreting pronouns and temporal aspects of the information conveyed.

1. **Word Knowledge:**

* Word knowledge is nothing but everyday knowledge that all the speakers share about the world.
* It includes the general knowledge about the structure of the world and what each language user must know about the other user’s beliefs and goals.
* This is essential to make the language understanding much better.

# **The challenges of NLP**

Below are the steps involved and some challenges that are faced in NLP:

1. **Challenge: Breaking the sentence**

Formally referred to as “sentence boundary disambiguation”, this breaking process is no longer difficult to achieve, but is nonetheless, a critical process, especially in the case of highly unstructured data that includes structured information. A breaking application should be intelligent enough to separate paragraphs into their appropriate sentence units; Highly complex data might not always be available in easily recognizable sentence forms. This data may exist in the form of tables, graphics, notations, page breaks, etc., which need to be appropriately processed for the machine to derive meanings in the same way a human would approach interpreting text.

***Solution:*** Tagging the parts of speech (POS) and generating dependency graphs

1. **Challenge: Building the appropriate vocabulary**

Using these POS tags and dependency graphs, a powerful vocabulary can be generated and subsequently interpreted by the machine in a way comparable to human understanding.Sentences are generally simple enough to be parsed by a basic NLP program. But to be of real value, an algorithm should also be able to generate, at a minimum, the following vocabulary terms: Employees; Management of risk; Ultimate accountability…..etc.

***Solution:*** Unfortunately, most NLP software applications do not result in creating a sophisticated set of vocabulary.

1. **Challenge: Linking different components of vocabulary**

Recently, new approaches have been developed that can execute the extraction of the linkage between any two vocabulary terms generated from the document (or “corpus”).

***Solution:*** Word2vec, a vector-space based model, assigns vectors to each word in a corpus, those vectors ultimately capture each word’s relationship to closely occurring words or set of words. But statistical methods like Word2vec are not sufficient to capture either the linguistics or the semantic relationships between pairs of vocabulary terms.

1. **Challenge: Setting the context**

One of the most important and challenging tasks in the entire NLP process is to train a machine to derive context from a discussion within a document. Consider the following two sentences:

“I enjoy working in a bank.”

“I enjoy working near a river bank.”

1. **Language differences**

Different languages have not only vastly different sets of vocabulary, but also different types of phrasing, different modes of inflection, and different cultural expectations. You can resolve this issue with the help of “universal” models that can transfer at least some learning to other languages. However, you’ll still need to spend time retraining your NLP system for each language.

1. **Training data:**

At its core, NLP is all about analyzing language to better understand it. A human being must be immersed in a language constantly for a period of years to become fluent in it; even the best AI must also spend a significant amount of time reading, listening to, and utilizing a language. The abilities of an NLP system depend on the training data provided to it. If you feed the system bad or questionable data, it’s going to learn the wrong things, or learn in an inefficient way.

# **Language and Grammar**

**Different Types of Grammar in NLP**

* **Context-Free Grammar (CFG)**
* **Constituency Grammar (CG)**
* **Dependency Grammar (DG)**

## **What is Grammar?**

Grammar is defined as the rules for forming well-structured sentences. In simple words, Grammar denotes syntactical rules that are used for conversation in natural languages.

The theory of formal languages is not only applicable here but is also applicable in the fields of Computer Science mainly in programming languages and data structures.

**For Example,** in the ‘C’ programming language, the precise grammar rules state how functions are made with the help of lists and statements.

Mathematically, a grammar G can be written as a 4-tuple (N, T, S, P) where,

N or VN = set of non-terminal symbols, or variables.

T or ∑ = set of terminal symbols.

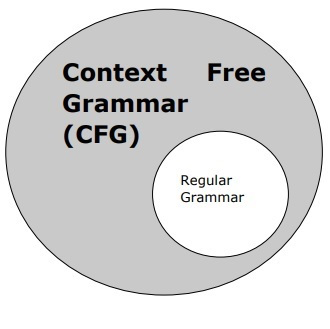
S = Start symbol where S ∈ N

P = Production rules for Terminals as well as Non-terminals.

It has the form α → β, where α and β are strings on VN ∪ ∑ and at least one symbol of α belongs to VN

**Context-Free Grammar (CFG)**

A context-free grammar, which is in short represented as CFG, is a notation used for describing the languages and it is a superset of Regular grammar which you can see from the following diagram:

****

**CFG consists of a finite set of grammar rules having the following four components**

* **Set of Non-Terminals**
* **Set of Terminals**
* **Set of Productions**
* **Start Symbol**

### **Set of Non-terminals**

It is represented by V. The non-terminals are syntactic variables that denote the sets of strings, which helps in defining the language that is generated with the help of grammar.

### **Set of Terminals**

It is also known as tokens and represented by Σ. Strings are formed with the help of the basic symbols of terminals.

### **Set of Productions**

It is represented by P. The set gives an idea about how the terminals and nonterminals can be combined. Every production consists of the following components:

* **Non-terminals,**
* **Arrow,**
* **Terminals (the sequence of terminals).**

The left side of production is called non-terminals while the right side of production is called terminals.

### **Start Symbol**

The production begins from the start symbol. It is represented by symbol S. Non-terminal symbols are always designated as start symbols.

## **Constituency Grammar (CG)**

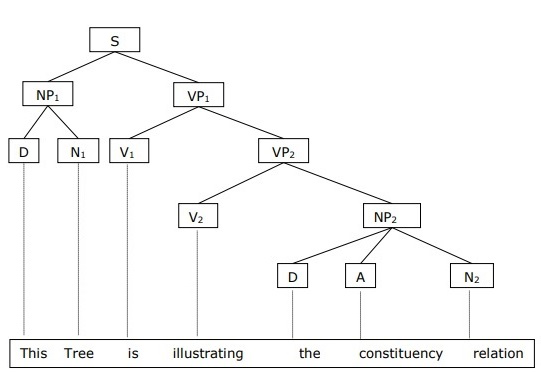
It is also known as Phrase structure grammar. It is called constituency Grammar as it is based on the constituency relation. It is the opposite of dependency grammar.

Before deep diving into the discussion of CG, let’s see some fundamental points about constituency grammar and constituency relation.

* All the related frameworks view the sentence structure in terms of constituency relation.
* To derive the constituency relation, we take the help of subject-predicate division of Latin as well as Greek grammar.
* Here we study the clause structure in terms of noun phrase NP and verb phrase VP.

**For Example,**

**Sentence: This tree is illustrating the constituency relation**

****

Now, Let’s deep dive into the discussion on Constituency Grammar:

In Constituency Grammar, the constituents can be any word, group of words, or phrases and the goal of constituency grammar is to organize any sentence into its constituents using their properties. To derive these properties we generally take the help of:

* Part of speech tagging,
* A noun or Verb phrase identification, etc

**For Example,** constituency grammar can organize any sentence into its three constituents- a subject, a context, and an object.

**Sentence: <subject> <context> <object>**

These three constituents can take different values and as a result, they can generate different sentences. For Example, If we have the following constituents, then

<subject> The horses / The dogs / They

<context> are running / are barking / are eating

<object> in the park / happily / since the morning

Example sentences that we can be generated with the help of the above constituents are:

“The dogs are barking in the park”

“They are eating happily”

“The horses are running since the morning”

Now, let’s look at another view of constituency grammar is to define their grammar in terms of their part of speech tags.

Say a grammar structure containing a

[determiner, noun] [ adjective, verb] [preposition, determiner, noun]

which corresponds to the same sentence – “The dogs are barking in the park”

Another view (Using Part of Speech)

< DT NN > < JJ VB > < PRP DT NN > -------------> The dogs are barking in the park

## **Dependency Grammar (DG)**

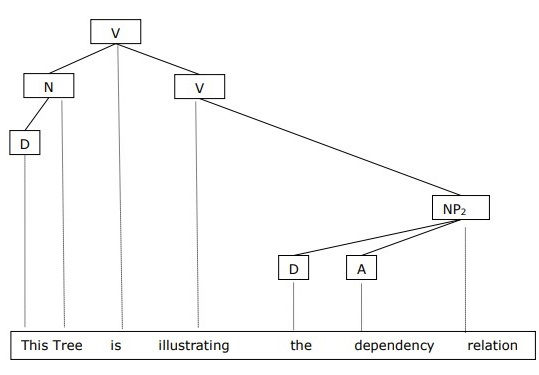
It is opposite to the constituency grammar and is based on the dependency relation. Dependency grammar (DG) is opposite to constituency grammar because it lacks phrasal nodes.

Before deep dive into the discussion of DG, let’s see some fundamental points about Dependency grammar and Dependency relation.

* In Dependency Grammar, the words are connected to each other by directed links.
* The verb is considered the center of the clause structure.
* Every other syntactic unit is connected to the verb in terms of directed link. These syntactic units are called dependencies.

**For Example,**

**Sentence: This tree is illustrating the dependency relation**

****

Now, Let’s deep dive into the discussion of Dependency Grammar:

1. Dependency Grammar states that words of a sentence are dependent upon other words of the sentence.

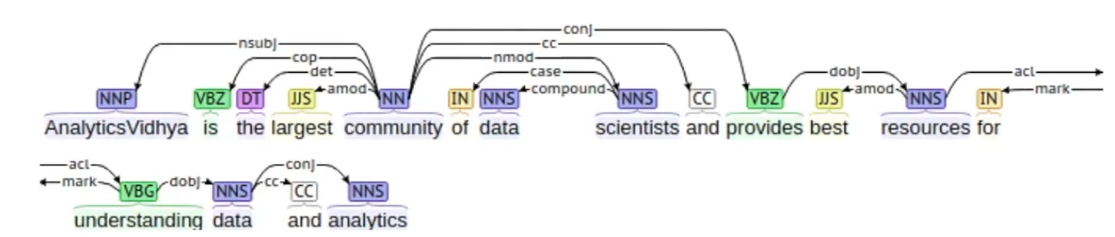
For Example, in the previous sentence which we discussed in CG, “barking dog” was mentioned and the dog was modified with the help of barking as the dependency adjective modifier exists between the two.

2. It organizes the words of a sentence according to their dependencies. One of the words in a sentence behaves as a root and all the other words except that word itself are linked directly or indirectly with the root using their dependencies. These dependencies represent relationships among the words in a sentence and dependency grammars are used to infer the structure and semantic dependencies between the words.

**For Example,** Consider the following sentence:

**Sentence: Analytics Vidhya is the largest community of data scientists and provides the best resources for understanding data and analytics**

**The dependency tree of the above sentence is shown below:**

****

In the above tree, the root word is “community” having NN as the part of speech tag and every other word of this tree is connected to root, directly or indirectly, with the help of dependency relation such as a direct object, direct subject, modifiers, etc.

These relationships define the roles and functions of each word in the sentence and how multiple words are connected together.

**We can represent every dependency in the form of a triplet which contains a governor, a relation, and a dependent,**

**Relation : ( Governor, Relation, Dependent )**

**which implies that a dependent is connected to the governor with the help of relation, or in other words, they are considered the subject, verb, and object respectively.**

**For Example,** Consider the following same sentence again:

**Sentence: Analytics Vidhya is the largest community of data scientists**

**Then, we separate the sentence in the following manner:**

**< Analytics vidhya> <is> <the largest community of data scientists>**

**Now, let’s identify different components in the above sentence:**

* **Subject: “Analytics Vidhya” is the subject and is playing the role of a governor.**
* **Verb: “is” is the verb and is playing the role of the relation.**
* **Object: “the largest community of data scientists” is the dependent or the object.**

**Some use cases of Dependency grammars are as follows:**

### **Named Entity Recognition**

It can be used to solve the problems related to named entity recognition (NER).

### **Question Answering System**

It can be used to understand the relational and structural aspects of question-answering systems.

### **Coreference Resolution**

It can also be used in coreference resolutions in which the task is to map the pronouns to the respective noun phrases.

### **Text summarization and Text classification**

It can also be used for text summarization problems and they are also used as features for text classification problems.

### 

# **Processing Indian Languages**

Natural language processing has the potential to broaden the online access for Indian citizens due to significant advancements in high computing GPU machines, high-speed internet availability and increased use of smartphones. According to a [survey](https://www.drift.com/blog/Chatbots-report/), the consumers pointed out the benefits of the chatbots, among which 55% of people thought getting answers to simple questions was one of the significant benefits. Still, when it comes to India, that’s challenging as languages in India aren’t that simple.

As Indian languages pose many challenges for NLP like ambiguity, complexity, language grammar, translation problems, and obtaining the correct data for the NLP algorithms, it creates a lot of opportunities for NLP projects in India.

### **Top NLP libraries for Indian Languages**

### **iNLTK (Natural Language Toolkit for Indic Languages)**

iNLTK provides support for various NLP applications in Indic languages. The languages supported are Hindi (hi), Punjabi (pa), Sanskrit (sa), Gujarati (gu), Kannada (kn), Malayalam (ml), Nepali (ne), Odia (or), Marathi (mr), Bengali (bn), Tamil (ta), Urdu (ur), English (en).

iNLTK is like the NLTK Python package. It provides the feature for NLP tasks such as tokenisation and vector embedding for input text with an easy API interface.

One has to first install:



Then next is installing iNLTK using pip:



### **Indic NLP Library:**

The Indian languages have some difficulties which come from sharing a lot of similarity in terms of script, phonology, language syntax, etc., and this library provides a general solution.

Indic NLP Library provides functionalities like text normalisation, script normalisation, tokenization, word segmentation, romanistion, indicisation, script conversion, transliteration and translation.

**Languages supported:**

* Indo-aryan:

Assamese (asm), Bengali (ben), Gujarati (guj), Hindi/Urdu (hin/urd), Marathi (mar), Nepali (nep), Odiaa (ori), Punjabi (pan).

* Dravidian:

Sindhi (snd), Sinhala (sin), Sanskrit (san), Konkani (kok), Kannada (kan), Malayalam (mal), Teugu (tel), Tami (tam).

* Others:

English (eng).

**Tasks handled:**

* It handles bilingual tasks like Script conversions for languages mentioned above except Urdu and English.
* Monolingual tasks
* This language supports languages like Konkani, Sindhi, Telugu and some others which aren’t supported by iNLTK library.
* Transliteration amongst the 18 above mentioned languages.
* Translation amongst ten languages.

The library needs Python 2.7+, [Indic NLP Resources](https://github.com/anoopkunchukuttan/indic_nlp_resources) (only for some modules) and [Morfessor 2.0 Python Library](http://www.cis.hut.fi/projects/morpho/morfessor2.shtml).

Installation:



Next, download the resources folder which contains the models for different languages.

### **Top datasets for NLP (Indian languages)**

* **Semantic Relations from Wikipedia:** [Contains](https://console.developers.google.com/storage/browser/wikipedia_multilingual_relations_v1/) automatically extracted semantic relations from multilingual Wikipedia corpus.
* **HC Corpora (Old Newspapers):** This [dataset](https://www.kaggle.com/alvations/old-newspapers) is a subset of HC Corpora newspapers containing around 16,806,041 sentences and paragraphs in 67 languages including Hindi.
* **Sentiment Lexicons for 81 Languages:** This [dataset](https://www.kaggle.com/rtatman/sentiment-lexicons-for-81-languages) contains positive and negative sentiment lexicons for 81 languages which also includes Hindi.
* **IIT Bombay English-Hindi Parallel Corpus:** This dataset contains parallel corpus for English-Hindi and monolingual Hindi corpus. This [dataset](http://www.cfilt.iitb.ac.in/iitb_parallel/) was developed ar the Center for Indian Language Technology.
* **Indic Languages Multilingual Parallel Corpus:** This [parallel corpus](http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/index.html) covers 7 Indic languages (in addition to English) like Bengali, Hindi, Malayalam, Tamil, Telugu, Sinhalese, Urdu.
* **Microsoft Speech Corpus (Indian languages)(Audio dataset)**: This [corpus contains](https://msropendata.com/datasets/7230b4b1-912d-400e-be58-f84e0512985e) conversational, phrasal training and test data for Telugu, Gujarati and Tamil.
* **Hindi Speech Recognition Corpus(Audio Dataset):** This is a [corpus collected](https://kingline.speechocean.com/exchange.php?act=view&id=16389) in India consisting of voices of 200 different speakers from different regions of the country. It also contains 100 pairs of daily spontaneous conversational speech data.

# **NLP Applications**

**1.** **Chatbots:**

* Chatbots are the bots designed for a particular use of interplay with human beings or different fellow machines using the strategies of AI.
* Chatbots are designed keeping in mind the human interaction. The use of Chatbots goes way back to 1966 when the first chatterbot named “ELIZA” was once designed at MIT.
* Eliza could hold the dialogue flowing with the human it interacted with, this led to the improvement of chatbots that may want to have a wonderful impact on human beings struggling from psychological issues.

**2. Text Classification:**

* Texts are a form of unstructured information that possesses very prosperous records inside them.
* Text Classifiers categorize and arrange exceptionally a great deal with any form of textual content that we use currently.
* With the methodologies of Deep Learning such as CNN and RNN the outcomes solely get better with the improved textual content information that we generate.

**3. Sentiment Analysis:**

* Feedback is one of the fundamental factors of true communication.
* Inspecting people’s sentiment in the direction of a product is necessary now greater than ever.
* The Bag of words(BOW) strategy where the authentic order of words is lost, however, the sentence below is decreased to the words that clearly make a contribution in figuring out the sentiment is pretty famous for sentiment analysis.

**4. Machine Translation:**

* Achieving multilingualism can frequently be a challenging mission to accomplish, so to make our lifestyles simpler at least in the factor of communication, Machine Translation comes to the rescue.
* Over the current years with the assets to put in force Neural networks, machine translation has drastically elevated in its high-quality such that translating between languages is as easy as urgent a button on the reachable smartphones or tablets.
* Google Translate helps with more than one hundred languages and can even translate language pictures from up to 37 languages.

**5. Virtual Assistants:**

* Virtual assistants are designed to engage with human beings in a very human way, most of their responses would be like the responses you would acquire from a pal or colleague.
* They are engineered to take delivery of the user’s voice instructions and operate the assignment entrusted with them.
* In addition to NLP virtual assistants additionally focuses on Natural Language Understanding so as to maintain up with the ever-growing slangs, sentiments, and intent at the back of the user’s input.

**6. Speech Recognition:**

* NLP can be used to recognize speech and convert it into text. This can be used for applications such as voice assistants, dictation software, and speech-to-text transcription.

**7. Text Summarization:**

* NLP can be used to summarise large volumes of text into a shorter, more manageable format. This can be useful for applications such as news articles, academic papers, and legal documents.

**8. Named Entity Recognition:**

* NLP can be used to identify and classify named entities, such as people, organisations, and locations. This can be used for applications such as search engines, chatbots, and recommendation systems.

**9. Question Answering:**

* NLP can be used to automatically answer questions posed in natural language. This can be used for applications such as customer service, chatbots, and search engines.

**10. Language Modelling:**

* NLP can be used to build models of natural language that can generate new text. This can be used for applications such as chatbots, virtual assistants, and creative writing.

# **Early NLP Systems**

**1950s**

**The Birth of NLP:** In the 1950s, computer scientists began to explore the possibilities of teaching machines to understand and generate human language. One prominent example from this era is the “Eliza” program developed by Joseph Weizenbaum in 1966. Eliza was a simple chatbot designed to simulate a conversation with a psychotherapist. While Eliza’s responses were pre-scripted, people found it surprisingly engaging and felt like they were interacting with an actual human.

**1960s-1970s**

**Rule-based Systems:** During the 1960s and 1970s, NLP research focused on rule-based systems. These systems used a set of predefined rules to analyse and process text. One notable example is the “SHRDLU” program developed by Terry Winograd in 1970. SHRDLU was a natural language understanding system that could manipulate blocks in a virtual world. Users could issue commands like “Move the red block onto the green block,” and SHRDLU would execute the task accordingly. This demonstration highlighted the potential of NLP in understanding and responding to complex instructions.

**1980s-1990s**

**Statistical Approaches and Machine Learning:** In the 1980s and 1990s, statistical approaches and machine learning techniques started gaining prominence in NLP. One groundbreaking example during this period is the development of Hidden Markov Models (HMMs) for speech recognition. HMMs allowed computers to convert spoken language into written text, leading to the development of speech-to-text systems. This breakthrough made it possible to dictate text automatically and have it transcribed, revolutionising fields like transcription services and voice assistants.

**2000s-2010s**

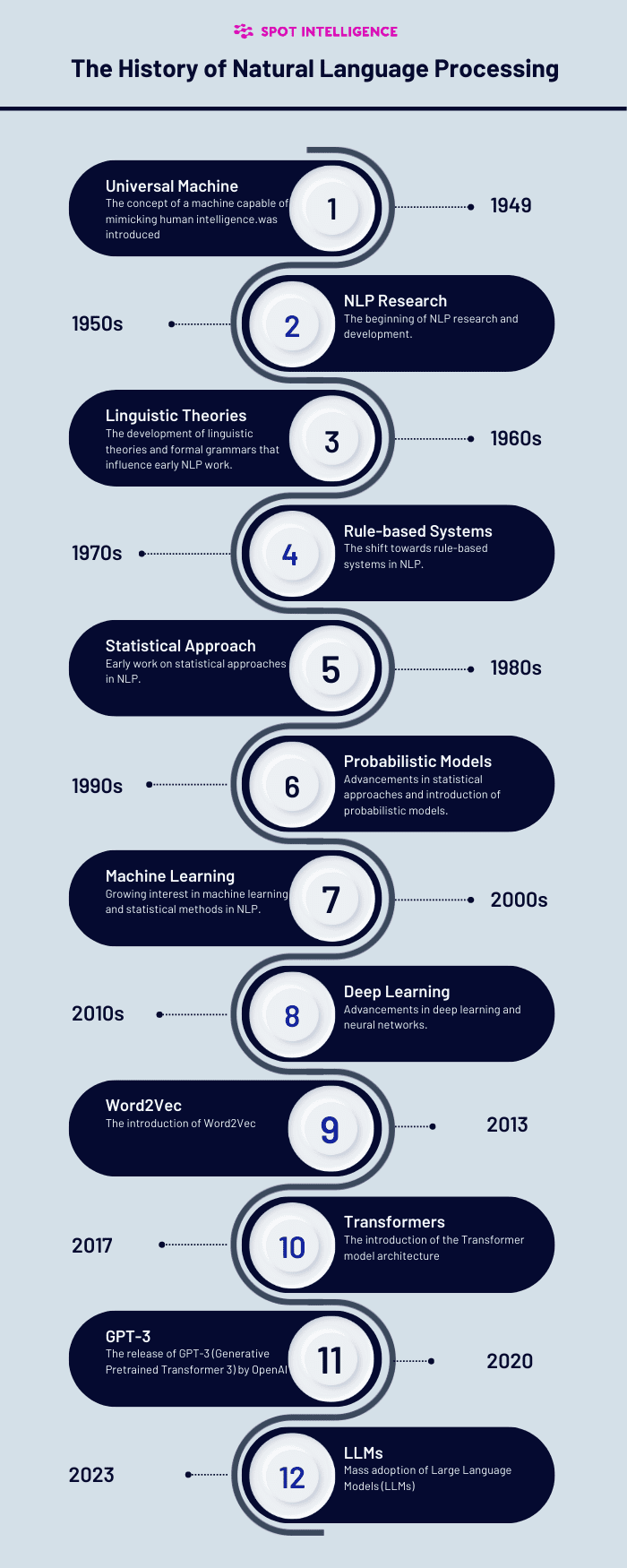
**Deep Learning and Neural Networks:** The 2000s and 2010s witnessed the rise of deep learning and neural networks, propelling NLP to new heights. One of the most significant breakthroughs was the development of word embeddings, such as Word2Vec and GloVe. These models represented words as dense vectors in a continuous vector space, capturing semantic relationships between words. For example, words like “king” and “queen” were represented as vectors that exhibited similar geometric patterns, showcasing their relational meaning.

**2017**

In 2017, Google introduced Google Translate’s neural machine translation (NMT) system, which used deep learning techniques to improve translation accuracy. The system provided more fluent and accurate translations compared to traditional rule-based approaches. This development made it easier for people to communicate and understand content across different languages.

**Present Day**

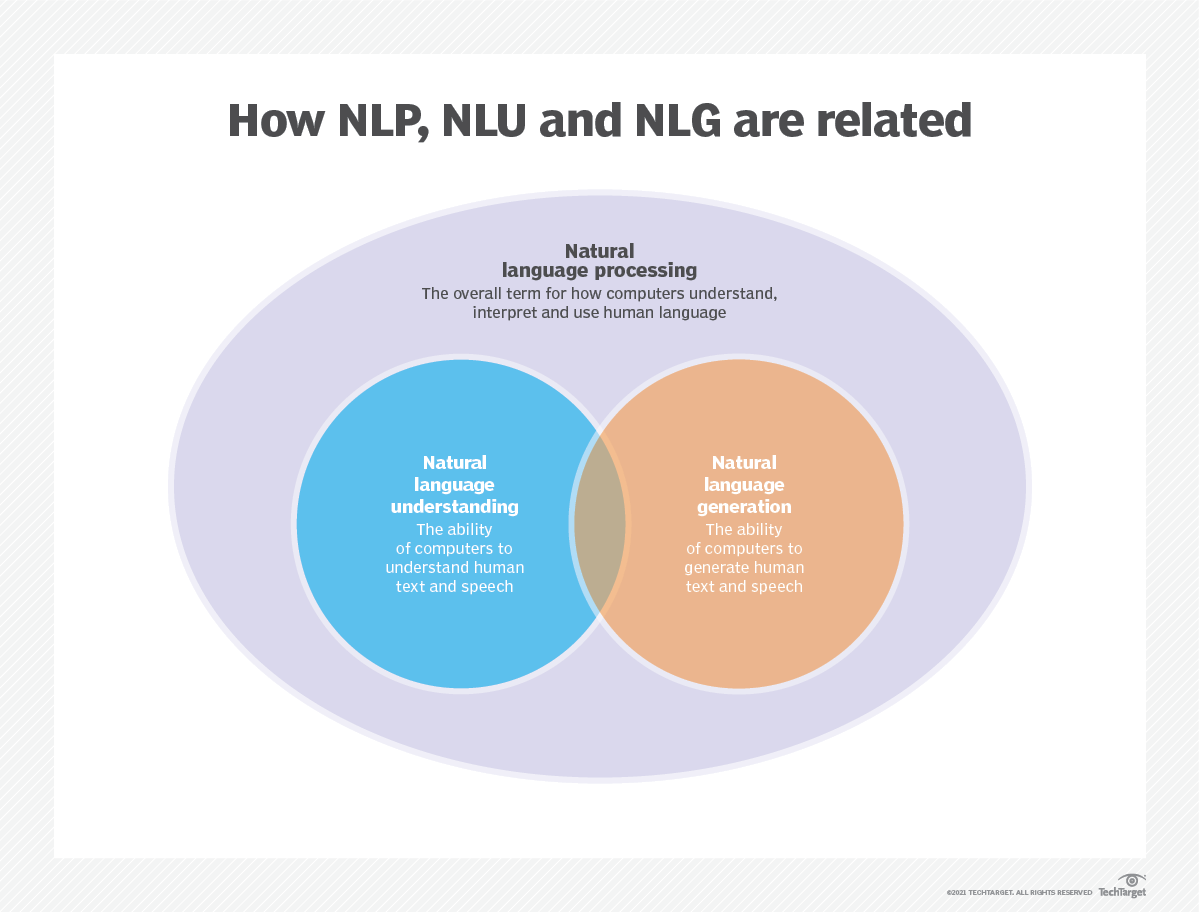
**Transformer Models and Large Language Models:** In recent years, transformer models like OpenAI’s GPT (Generative Pre-trained Transformer) have made significant strides in NLP. These models can process and generate human-like text by capturing the contextual dependencies within large amounts of training data. GPT-3, released in 2020, demonstrated the ability to generate coherent and contextually relevant text across various applications, from creative writing to customer support chatbots.



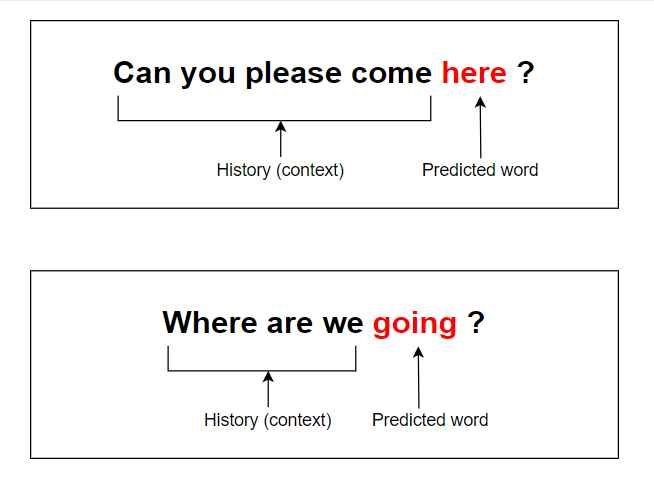
****

# **What is language modelling?**

Language modelling (LM) **analyses bodies to text** to provide a foundation for word prediction. These models use statistical and probabilistic techniques to determine the **probability of a particular word sequence** occurring in a sentence.

****

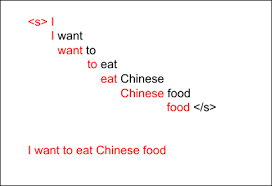
* In text generation, a language model completes a sentence by generating text based on the incomplete input sentence. This is the idea behind the autocomplete feature when texting on a phone or typing in a search engine. The model will give suggestions to complete the sentence based on the words it predicts with the highest probabilities.



# **Various Grammar-based Language Models**

There are two categories that Language Models fall under:

1. **Statistical Language Models:** These models use traditional statistical techniques like N-grams, Hidden Markov Models (HMM), and established linguistic rules to learn the probability distribution of words. Statistical Language Modeling involves the development of probabilistic models that can predict the next word in the sequence given the words that precede it.



1. **Neural Language Models:** These models are new players in the NLP world and have surpassed the statistical language models in their effectiveness.They use different kinds of Neural Networks to model language. The use of neural networks in the development of language models has become so popular that it is now the preferred approach for challenging tasks like speech recognition and machine translation

# **STATISTICAL LANGUAGE MODELS**

**N-gram Language Models:**

The n-gram model is a probabilistic language model that can predict the next item in a sequence using the (n − 1)–order Markov model. Let’s understand that better with an example. Consider the following sentence:

“I love reading blogs on Educative to learn new concepts”

A **1-gram** is a one-word sequence. For the above sentence, the unigrams would simply be: “I”, “love”, “reading”, “blogs”, “on”, “Educative”, “and”, “learn”, “new”, “concepts”.

A **2-gram** (or bigram) is a two-word sequence of words, like “I love”, “love reading”, “on Educative” or “new concepts”.

A **3-gram** (or trigram) is a three-word sequence of words, like “I love reading”, “blogs on Educative”, or “learn new concepts”.

An N-gram language model predicts the probability of a given N-gram within any sequence of words in the language. If we have a good N-gram model, we can predict *p*(*w*∣*h*), or the probability of seeing the word w given a history of previous words h, where the history contains n-1 words.

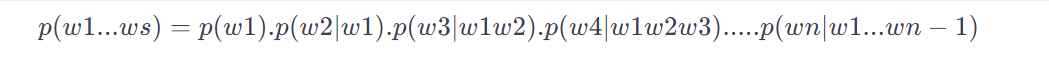
**Example:** “I love reading \_\_\_”. Here, we want to predict what word will fill the dash based on the probabilities of the previous words.

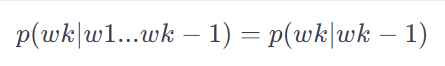
We must estimate this probability to construct an N-gram model. We compute this probability in two steps:

Apply the chain rule of probability

We then apply a very strong simplification assumption to allow us to compute p(w1…ws) in an easy manner.

The chain rule of probability is:

What is the chain rule? It tells us how to compute the joint probability of a sequence by using the conditional probability of a word given previous words.

Here, we do not have access to these conditional probabilities with complex conditions of up to n-1 words.So, how do we proceed? This is where we introduce a simplification assumption. We can assume for all conditions, that:

Here, we approximate the history (the context) of the word wk by looking only at the last word of the context.This assumption is called the Markov assumption. It is an example of the Bigram model. The same concept can be enhanced further for example for trigram model the formula will be:**

These models have a basic problem: they give the probability to zero if an unknown word is seen, so the concept of smoothing is used. In smoothing we assign some probability to the unseen words. There are different types of smoothing techniques such as Laplace smoothing, Good Turing, Kneser-ney smoothing.

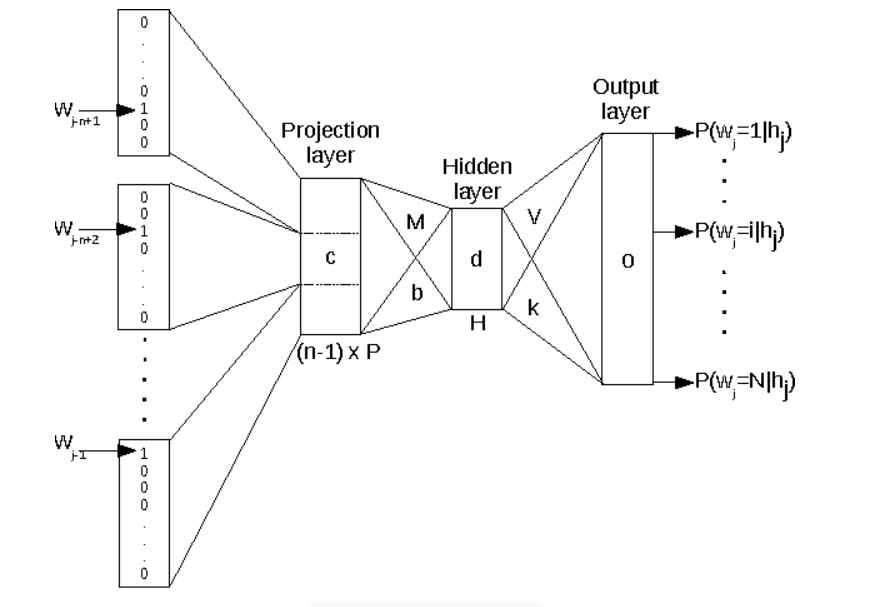
## **Introduction to Neural language models**

Neural language models have some advantages over probabilistic models. For example, they don’t need smoothing, they can handle much longer histories, and they can generalise over contexts of similar words

For a training set of a given size, a neural language model has much higher predictive accuracy than an n-gram language model.

On the other hand, there is a cost for this improved performance: neural net language models are strikingly slower to train than traditional language models, and so for many tasks an N-gram language model is still the right tool.

In neural language models, the prior context is represented by embeddings of the previous words. This allows neural language models to generalise unseen data much better than N-gram language models.

****

**Word embeddings** are a type of word representation that allow words with similar meaning to have a similar representation. Word embeddings are, in fact, a class of techniques where individual words are represented as real-valued vectors in a predefined vector space.

Each word is mapped to one vector, and the vector values are learned in a way that resembles a neural network. Each word is represented by a real-valued vector, often tens or hundreds of dimensions.

The Neural language models were first based on RNNs and word embeddings. Then the concept of LSTMs, GRUs and Encoder-Decoder came along. The recent advancement is the discovery of Transformers, which has changed the field of Language Modelling drastically.



The RNNs were then stacked and used bidirectionally, but they were unable to capture long term dependencies. LSTMs and GRUs were introduced to counter this drawback.

The transformers form the basic building blocks of the new neural language models. The concept of transfer learning was introduced which was a major breakthrough. The models were pre-trained using large datasets.

For example, BERT is trained on the entire English Wikipedia. Unsupervised learning was used for training of the models. GPT-2 is trained on a set of 8 million web pages. These models are then fine-tuned to perform different NLP tasks.