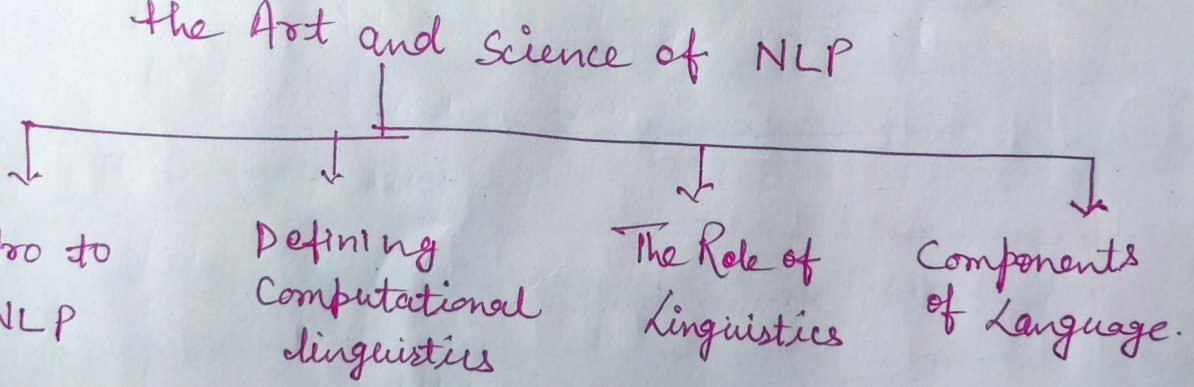
* NLP (Natural Language Processing)

→ NLP stands at the intersection of linguistics computer science, and artificial intelligence.

→ It enables machines to understand, interpret and respond to human language.

→ NLP's applications range from chatbots and language translation to sentiment-analysis and content generation.

→ It is a computational techniques for processing and analyzing human language.



→ Defining Computational Linguistics →

→ It combines linguistic theories with computational methods.

→ It envolves developing algorithms for automating language analysis, processing, and generation.

→ this field bridges the gap between language and machines. enabling meaningful interactions.

Role of Linguistics →

① Semantic Understanding →

Linguistic insights enable NLP models to comprehend word meanings, nuances, and contextual subtleties, enhancing accurate interpretation of text.

→ Syntax Analysis →

→ Linguistic theories guide NLP systems in structurally analyzing sentences, identifying grammatical components, and understanding relationships between words.

→ Named Entity Recognition (NER): →

Linguistic patterns assist NER algorithms in identifying and categorizing entities like names, dates and locations, aiding in information extraction.

* Components of Natural Language →

→ Human language comprises various components, including syntax (sentence structure), semantics (meaning), morphology (word forms), and phonetics (speech sounds).

→ NLP algorithms must decipher each of these components to comprehend and generate language accurately.

→ Benefits of Computational Linguistics →

Efficient Language Processing:

Computational linguistics steamlines the analysis, understanding, and generation of human language, enhancing the efficiency of various tasks.

Improved Communication :-

It bridges language barriers, enabling seamless communication between indivisuals speaking different languages.

Enhanced Data Insights: →

Computational linguistics extracts valuable insights from textual data, supporting informed

Innovation and Automation: →

By automating language-related tasks. it paves the way for innovative applications such as Chatbots, sentiment analysis, and content generation,

Applications of NLP: →

(1) Sentiment Analysis
(2) Language Translation
(3) Named Entity Recognition
(4) Deep learning for NLP.

History of NLP ⇒

Various NLP Tasks ⇒

Language Modeling → this is the task of predicting what the next word in a sentence will be based on the history of previous words.

→ It is useful for building solutions for a wide variety of problems, such as speech recognition, OCR, handwritting recognition, Machine translation.

Text classification →

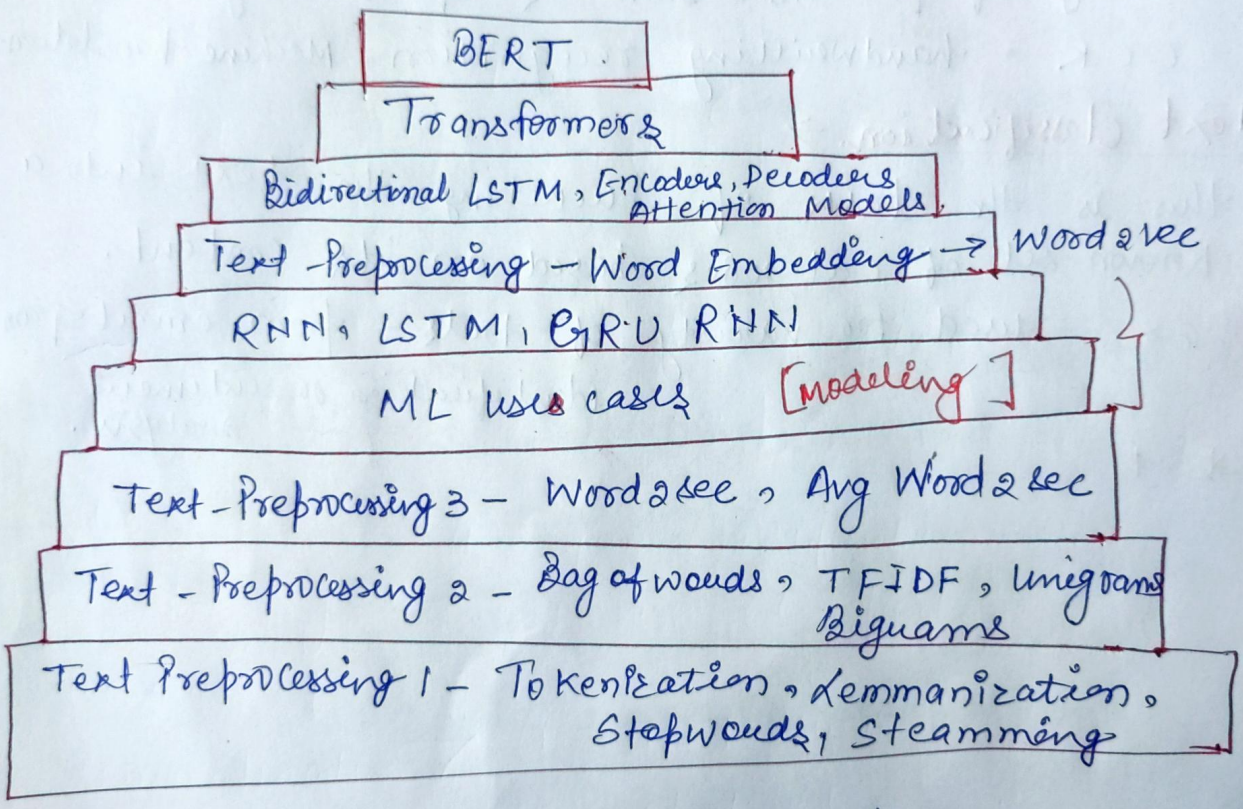
this is the task of bucketing the text into a known set of categories based on its content.

ex :- used for variety of tools - from email spam identification or sentiment analysis.

Why NLP? → NLP used on text based dataset.

→

Examples of NLP → • Chat bots
• Email Filters
• Sentiment Analysis
• Machine Translation



Libraries used in NLP → NLTK SPACY, TextBlob

(1) Tokenization => (Breaking text into words or phrases)

→ It is converting sentence into words.

→ It is the process of tokenizing or splitting a string, text into a list of tokens.

Stemming → It is the process of reducing a word to its word stem that affixes to suffixes and prefixes or the Roots.

Ex :- going, goes, gone → go (Meaningful word)

finally, final, finalized → fina (Meaning is gone)

Advantage → It is really fast.

Disadvantage → It is removing the meaning.

→ Stemming is a NLP technique that is used to reduce words to their base form as root.

* Lemmatization → (Reducing words to their base form)
It is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item.

history, historical → history

→ Lemmatization does morphological analysis of the words.

Ex :- better → Good
rocks → rock
Corpora → Corpus

Applications →

→ Used in compact indexing.
→ Used in comprehensive retrieval systems like search engines.

Advantage → Meaningful words

Disadvantage → It is slow.

Use cases in stemming → (1) Spam classification
(2) Review classification

Use cases of Lemmatization → (1) Text summerization
(2) chat bot
(3) Language Translation.

Text Preprocessing (1) =>

(1) Tokenization
(2) Stop words
(3) stemming
(4) Lemmatization

Stopwords Removal

↓

Removing common words with no significant meanings like "the", "and", "a".

Parts-of-speech Tagging

Identifying the syntactical properties of words.

Text preprocessing step 2 :→ [Feature engineering]

Convert words into vectors.

words ——→ vectors.

Techniques used in step 2 :

(2) Bag of words (3) TFIDF (4) Word2Vec.

(1) One hot Encoding

(Term Frequency - Inverse Document Frequency)

(5) Ngrams.

* Basic Terminologies used in NLP →

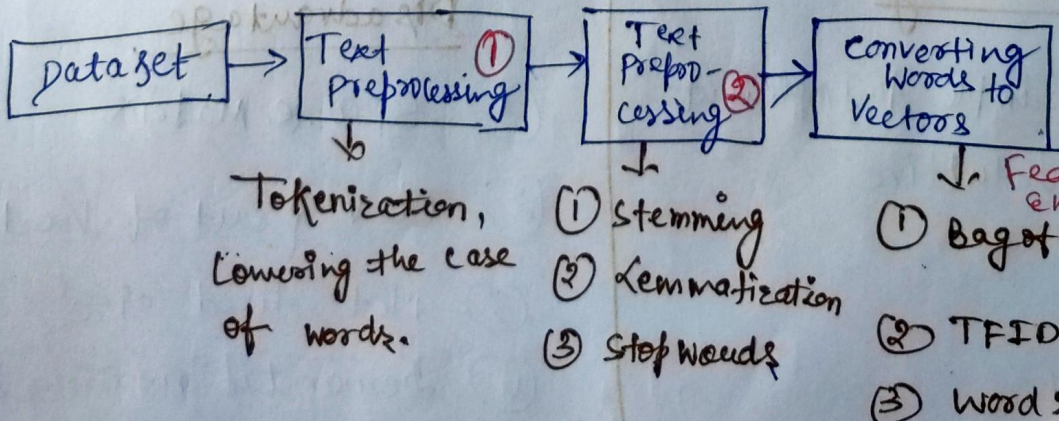
(1) CORPUS → Paragraph (Datapoints)

(2) Documents → It is like sentences

(3) Vocabulary → unique words in dictionary

(4) words.

(1) **One hot encoding** → output should be in binary (0/1) format.



* ① ~~Bag of words~~ (

* ① One hot encoding →

ex :- [A man eat food
Cat eat food
People watch KRISH YT] } Corpus

Paragraph (no. of words) = 9

A man eat food cat pepple watch KRISH YT

* Represent as one hot encode →

D1 [[1 0 0 0 0 0 0 0 0 0] (A), [0 1 0 0 0 0 0 0 0 0] (man)
[0 0 1 0 0 0 0 0 0 0] (eat), [0 0 0 1 0 0 0 0 0 0] (food)]

In this encode there are lots of issues —

① sparsity

② when vocab size decreases then can't train the model.

Advantage

① Simple to Implement

② Intuitive

Disadvantage

① Sparse Matrix

② OOV {out of Vocabulary}

③ Not fixed size.

④ Semantic meaning between word is not capture.

(2) Bag of words →

D_1 → He is a good → good boy

D_2 → She is a good girl → good girl

D_3 → Boys and girls are good → Boys girls good

Remove these words

Vocabulary	Frequency
good	→ 3 times
boy	→ 2 times
girl	→ 2 times

	f_1	f_2	f_3	
	→ good	boy	girl	o/p
D_1 →	1	1	0	→ not present 0
D_2 →	1	0	1	1
D_3 →	1	1	1	→ All present 1

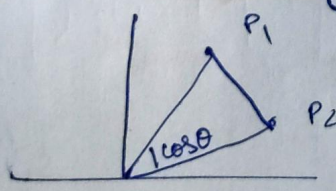
Doc 1 →
Doc 2 →
Doc 3 →

→ In BOW, we can make binary BOW.

Advantages

(1) It is simple and Intuitive

(2) Cosine Similarity



$\cos 45^\circ = 0.53$

$1 - 0.53 =$ Cos-Similarity

Disadvantages

(1) sparsity

(2) OOV

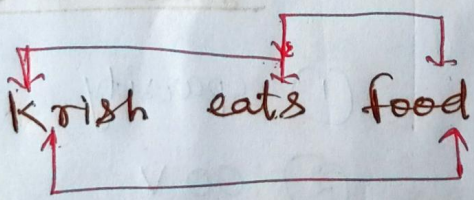
(3) Ordering of the words

* **Ngrams** → It helps to find semantic meanings.

	f_1 good	f_2 boy	f_3 girl
sentence 1	1	1	0
sent 2	1	0	1
set 3	1	1	1

Bigrams → Apart from using single features we will be using common features also.

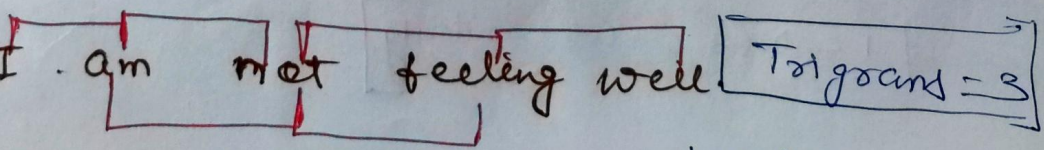
	f_4 Good boy	f_5 Good girl
S_1	1	0
S_2	0	1
S_3	0	0



practically

Krish eats Eats food

Trigrams →



I am not, am not feeling, not feeling well.

f_1	f_2	f_3	f_4	f_5
Krish	is	not	feeling	well

NLTK.sent_tokenize() →

Return a sentence-tokenized copy of "text" using NLTK's recommended sentence tokenizer.

re.sub(pattern, repl, string, count=0, flags=0) ⇒

Return the string obtained by replacing the leftmost non-overlapping occurances of the pattern in string

1. Q If we remove the words like a, the, is, an we use?

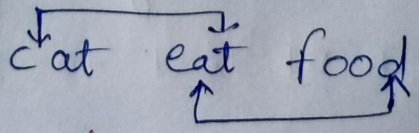
Ans → Stopword

2. Q Bag of words in text preprocessing is a –

Ans → Feature extraction Technique

3. Q. For the sentence "cat eat food", how many bigrams can be created?

Ans → 2



4. Q Collections of documents is called?

Ans → Corpus

5. Q tf-idf is used in?

→ Text preprocessing & Page ranking in search engines

① Bagwords → Text → Vectors

② stopwords

Sentence 1 → He is a good boy. → good boy

Sentence 2 → She is a good girl → good girl

Sentence 3 → Boy and girl are good → Boy girl good

② ↓ S-top words → (used to remove unnecessary words (. is, a, he, she, are)

Sent 1	good	boy
sent 2	good	girl
sent 3	boy	girl good

③ Frequency (Vocabulary) → ~~Frequency~~

good	3
boy	2
girl	2

④

	Good f_1	boy f_2	girl f_3
Sent 1	1	1	0
Sent 2	1	0	1
sent 3	1	1	1

Disadvantage

① out of vocabulary

② computation is high.

③ Semantic meaning is missing.

→ so TF-IDF used to remove (semantic meaning missing).

* TF - IDF (Term Friqueny - Inverse Document frequeney)

Sent1 : good boy
Sent2 : good girl
Sent3 : boy girl good

} which ever words are rarely present in the sentences we should give more weighted:

Term - Friqueny (TF) → (No. of repetations of words in sentence / No. of words in Sentence)

Inverse Document friqueny (IDF) → $\log_e$ (No. of sentences / No. of sentences containing the word)

Term friqueny (TF) →

↓	Sent 1	sent 2	sent 3
good	1/2	1/2	1/3
boy	1/2	0	1/3
girl	0	1/2	1/3

TF X IDF

*** Inverse Document Friqueny (IDF) →**

good $\log_e(3/3) = 0$
boy $\log_e(3/2)$
girl $\log_e(3/2)$

Mamta

TF * IDF

	f_1	f_2	f_3
	good	boy	girl
Sent 1	$\frac{1}{2} \times 0 = 0$	$\frac{1}{2} \times \log_e(3/2)$	$\frac{1}{2} 0 \times \log_e(3/2) = 0$
Sent 2	$\frac{1}{2} \times 0 = 0$	$0 \times \log_e(3/2) = 0$	$\frac{1}{2} * \log_e(3/2)$
sent 3	$\frac{1}{2} \times 0 = 0$	$\frac{1}{3} \log_e(3/2)$	$\frac{1}{3} \log_e(3/2)$

here, some amount of semantic information is present.

→ Words that are frequently present and words that are only present in a specific sentence.

→ Advantages

1. It is Intuitive
2. Word importance is getting captured

→ Overcome sparse matrix.
→ Sparsity problem removed

Disadvantage

1. ~~Sparsity~~
2. Out of Vocab
3. Not capture semantic meaning.

* Word Embeddings ⇒ [Words ⟶ Vectors]

→ It is a technique which converts words into vectors.

Word Embedding

- Count or Frequency
- Deep learning train models.

(1) Count or Frequency →

(1) Bag of words (BOW)

(2) TF-IDF

(3) One Hot Encoding

(2) Deep Learning Trained Model

(1) Word2 Vec → Hierarchical softmax

Continuous Bag of words (CBOW)

Skip Grams

(2) Word2 vec → [Feature Representation]

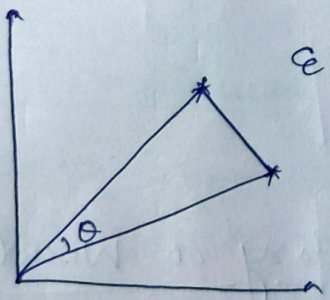
→ Every word is used to create vectors.

(1) → that vectors will be in limited dimensions.

(2) → Sparsity is reduced in word2vec.

(3) → Semantic meaning of vectors are maintained.

* Cosine Similarity →



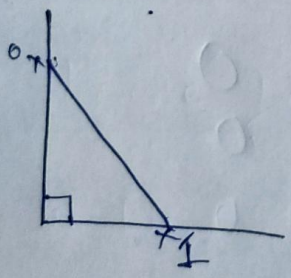
Euclidian Distance -

Distance = 1 - cosine similarity

1 - 0.7071 = 0.29

Cosine → Similarity = cosθ

cos45° = 0.7071 = $\frac{1}{\sqrt{2}}$



Cosine_Similarity = cos90° = 0

Distance = 1 - 0 = 1

CBOW (Continuous Bag of words) ⇒

→ CORPUS :⇒ Dataset → Training

Train word2vec model :⇒

[Krish channel is related to Data Science.]

context → context word

→ O/P Target (centered word)

① We have to decide window size

window size = 5

context word → related

centered word → is

Training Data

Independent Feature	O/P
→ Krish, channel, Related To	Is
→ Channel, Is, To, Data	Related
→ Is, Related, Data, Science	To

* Bag of words (BOW) ⇒

Vector representation of vocabulary or word →

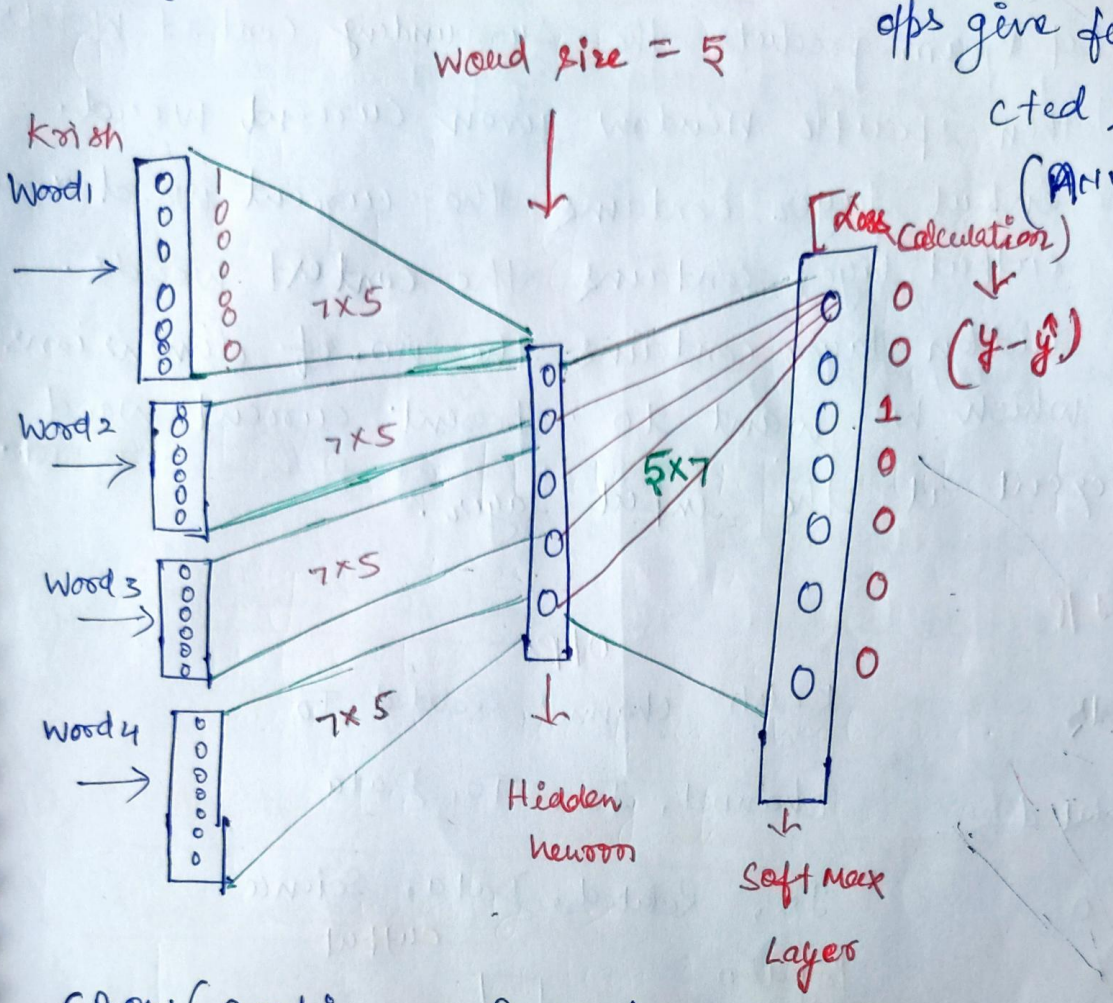
Krish → 1 0 0 0 0 0 0

channel → 0 1 0 0 0 0 0

is → 0 0 1 0 0 0 0

related → 0 0 0 1 0 0 0

⇒ Data representation, independent I/ps and o/ps give fully connected layer.



CBOW (Continuous Bag of Words) : CBOW model predicts the current word given context words within a specific window.

- the input layer contains the context words and the output layer contains the current word.

→ the hidden layer contains the no. of dimensions in which we want to represent the current word present at the output layer.

→ CBOW → [Predicts the target word based on its surrounding context words]

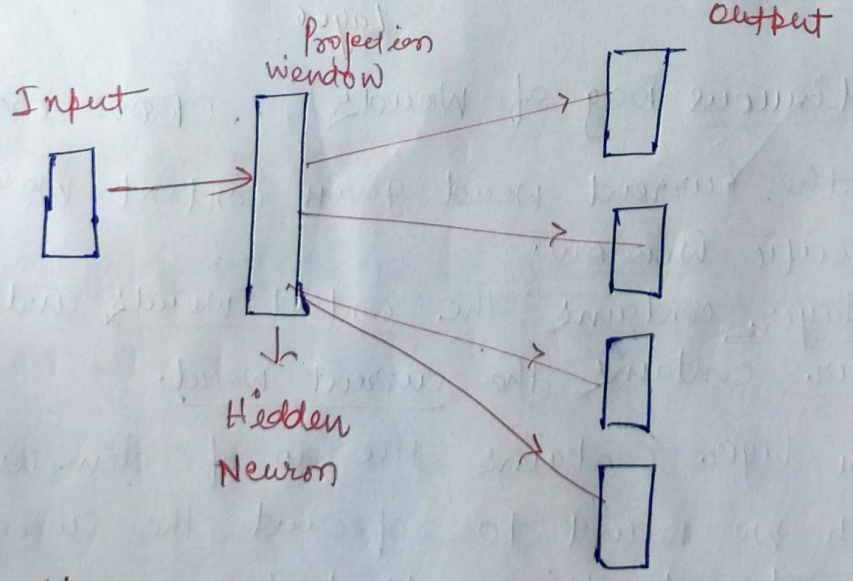
(2) Skip Gram ⇒ [Predicts the context words based on the target word]

→ Skip Gram predicts the surrounding context words within specific window given current word.

→ the input layer contains the current word and the output layer contains the context words.

→ the hidden layer contains the no. of dimensions in which we want to represent current word present at the input layer.

I/P	O/P
~~Krish~~	Krish, Channel, Related, To
Related	Channel, Is, To, Data
To	Is, Related, Data, Science



(3) Hierarchical Softmax →

→ Transforms the problem of predicting one word into a set of binary decision.

→ Word2vec Applications in NLP → [Enhancing Semantic Understanding]

(1) Semantic search → Matching synonyms, antonyms, and related concepts.

(2) Question answering → Providing answers to questions by understanding the meaning behind them.

(3) Chatbots → Teaching machines to understand human language better.

(4) Sentiment Analysis → Classifying text as positive, negative, or neutral based on the meaning of the words.

* Amplifying NLP with Word Embeddings:

Real-world use Cases →

(1) Amazon Recognition → Uses Word2vec and other algorithms to recognize objects and faces in images.

(2) Google Translate → Uses Word2vec to improve machine translation by understanding the meaning of words.

(3) Spotify → Uses Word2vec to recommend songs based on the meaning behind the lyrics.

(4) Google News → Uses Word2vec to group related articles based on the meaning of the words.

Word2vec's Limitations & challenges.

Limitations

1. Ignorance of concepts and semantic structure
2. Difficulty handling rare words or complex phrases
3. Lack of accounting for different word senses

Challenges

1. optimizing algo performance
   - Reducing the dimensionality of vectors.
   - Dealing with out-of-Vocab words and misspellings.