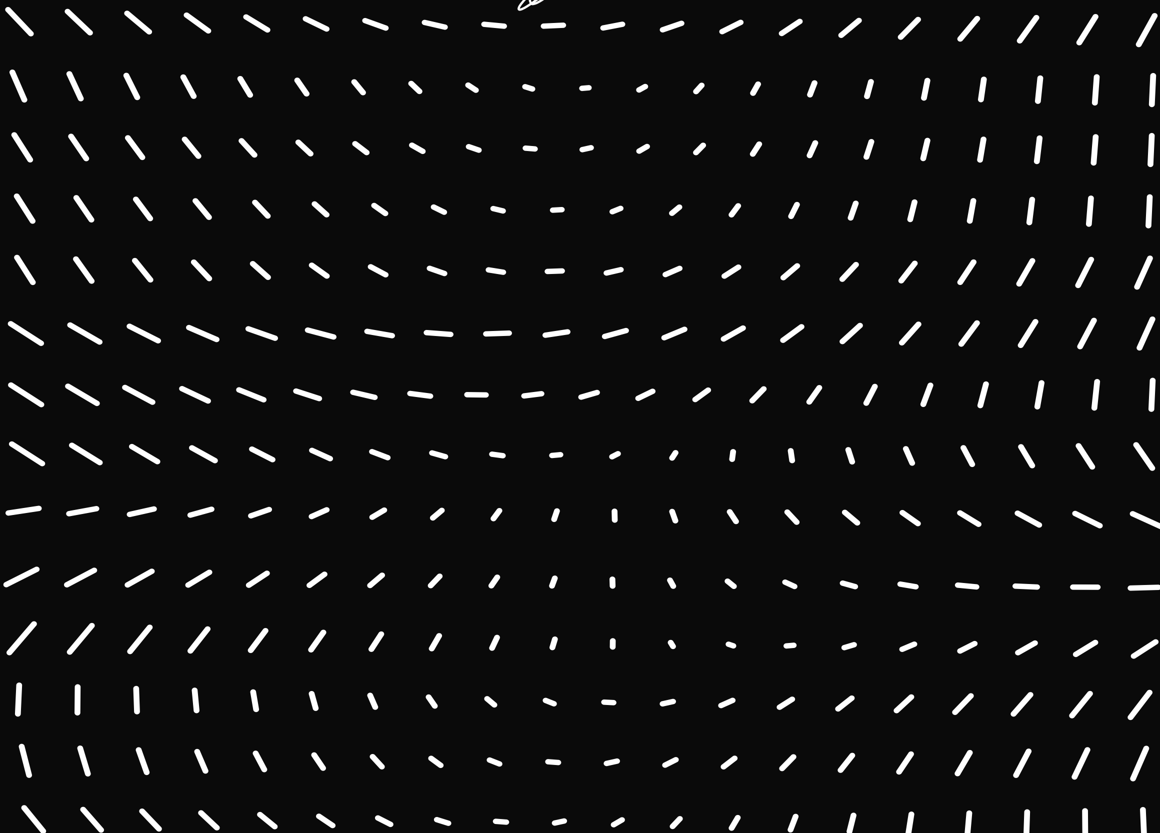
ANLP



Representation -

- Words, tokens, vectors, embeddings used to be similar to words

Chomsky view - computational linguistics
Shannon view - information theoretic ("encoding" & "decoding")

Splitting tokens used to be done at word/sentence level (look for space/period..)
Now, they are subword

Early Representations - one hot vector, tf idf, cooccurrence matrix with SVD
Limitations - Sparsity, ambiguity, lack of context

Deep Learning in NLP:
Word 2 Vec
↳ (CBOW) Continuous Bag of Words: Predict middle word from context
↳ Skip Gram (SG): Predict context from middle word

"Word Arithmetic" is now possible
E(King) - E(Man) + E(Woman) ≈ E(Queen)

Limitations: Ambiguity, Context (to a lesser extent)

Modelling

Language Modelling Problem:
Probability of a sequence (predict next word)
Before: Use conditional probability, with finite context (for practical reasons)

N-Gram Language Mode:
Sparse, doesn't exploit word similarity, finite context

Neural Network Language Modelling
Sparsity - solved, word similarity - solved, context - not solved, computational complexity

Recurrent Neural Networks

Used for time series data (stocks, weather...) & many NLP tasks.
Sequence to sequence (translation, speech recognition), classification tasks
Limitations: Long distance dependencies, vanishing & exploding gradients

LSTM - solved vanishing gradients but not long distance dependencies

2018, ELMo: Embeddings from Language Models Pre-Trained biLSTM for contextual embedding

Limitation of LSTM: Computationally complex (softmax), slow (ish), needs labelled data, transfer learning not possible

Attention:

Focus to individual components of sentences, interactions of words with other words

BERT: Bidirectional Encoder Representations from Transformers Pre-trained transformer encoder for sentence embedding → parallelization
↳ masked language modelling task

Today:

Representation: Embeddings from LLMs

Modelling: Encoder only: BERT, XML
Encoder-Decoder: T5, BART
Decoder only: GPT, LLaMA

NLP

1950s - 2010s: Symbolic → usually take in some input of symbols to do some task like classification.

Symbolic AI cares only about tokens presented as input & extracts relations & features

Ways to break text into elements:

- characters
- words (boundary: space)
- boundary: punctuation
- boundary: conjunctions

Phrasal vs discoursal boundaries matter. Ex: Ram and Shyam went out.
Cant split on this "and".

In images, two pixels are not related by themselves, but due to the real world entities they represent. However, in text, the grammar enforces a structure which gives inherent relations between elements.

In English, it is always actor verb object
Ex: The dog bit the man, but in many other languages, it can be actor object verb or object actor verb depending on the modifier.

Much more variation in text & language compared to images.
Some languages have no spaces
Ex: Mandarin, Thai

Language ≠ Text
Text loses some non verbal information like emphasis, intonation

"Translation is NLP-Complete" → needs all NLP tasks

① Words are complex - morph & encode some properties - Morphology
② Assign parts of speech
③ How are POS arranged? : Syntax (Depends on language)

Some languages are low morphology (Ex: Hindi), high morphology (Ex: Kannada). Hard to translate from low to high morphology languages

Noun Phrase

Specifier	Adjective	Noun
the	blue	house

English <u>requires</u> a subject. In stative sentences, a subject is artificially added. Ex: It's raining.

Pattern based learning ≠ Language Understanding