

Chapter 1: Introduction to NLP Concepts and Applications

Unit: Advanced Deep Learning



Introduction

- Natural Language Processing (NLP) is a field of artificial intelligence
- It focuses on the interaction between computers and humans through natural language
- NLP enables computers to understand, interpret, and generate Human language.
- It combines, computer science, and deep learning to process and analyze large amounts of natural language data, facilitating better communication between humans and machines.



Why Natural Language Processing?

What do we use language for?

- We **communicate** using language
- We **think** (partly) with language
- We **tell stories** in language
- We build **Scientific Theories** with language
- We make friends/build **relationships**

Why NLP ?

- **Access Knowledge** (search engine, recommender system...)
- **Communicate** (e.g. Translation)
- **Linguistics** and **Cognitive Sciences** (Analyse Languages themselves)



Why Natural Language Processing?

- **Amount of online textual data...**
 - 70 billion web-pages online (1.9 billion websites)
 - 55 million Wikipedia articles
- **Growing at a fast pace**
 - 9000 tweets/second
 - 3 million mail / second (60% spam)



Why Natural Language Processing?

- **Potential Users of Natural Language Processing**

- 7.9 billion people use some sort of language (January 2022)
- 4.7 billion internet users (January 2021) (~59%)
- 4.2 billion social media users (January 2021) (~54%)



Why Natural Language Processing?


• **What Products ?**

- Search: +2 billion Google users, 700 millions Baidu users
- Social Media: +3 billion users of Social media (Facebook, Instagram, WeChat, Twitter...)
- Voice assistant: +100 million users (Alexa, Siri, Google Assistant)
- Machine Translation: 500M users for google translate



Why is Language Hard to Model?

1. Productivity
2. Ambiguous
3. Variability
4. Diversity
5. Sparsity



Productivity

Definition

“property of the language-system which enables native speakers to construct and understand an indefinitely large number of utterances, including utterances that they have never previously encountered.” (Lyons, 1977)

→ New words, senses, structure are introduced in languages all the time

Examples: staycation and social distance were added to the Oxford Dictionary in 2021



Ambiguous

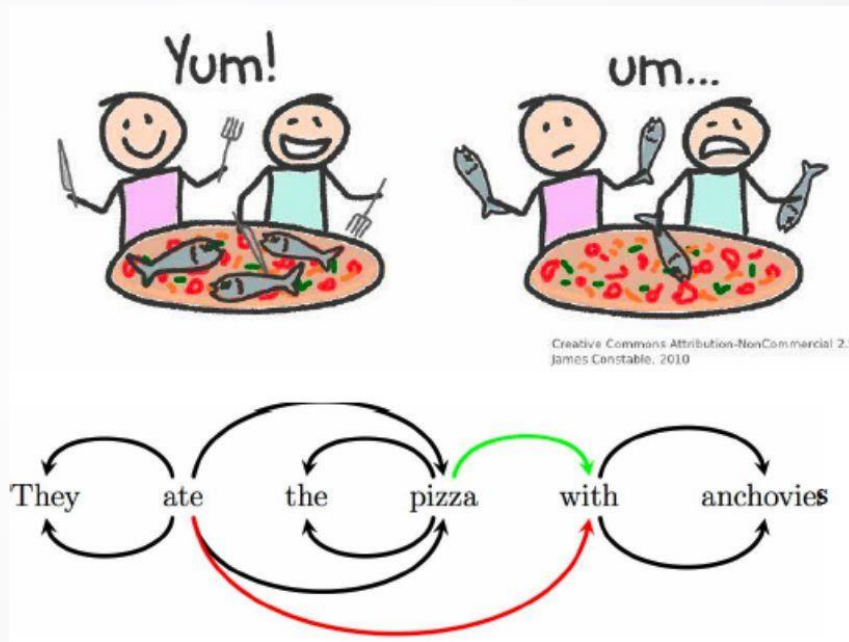
Most linguistic observations (speech, text) are open to **several interpretations**

We (Humans) disambiguate -i.e. **find the correct interpretation** - using all kind of signals (linguistic and extra linguistic)

Ambiguity can appear at all levels (phonology, graphemics, morphology, syntax, semantics)

Ambiguous

Syntactic Ambiguity






Ambiguous

Pragmatic Ambiguity

*Two Soviet ships collide, **one dies***

*Dealers will hear **car talk** at noon*



Variation

Language Varies at all levels

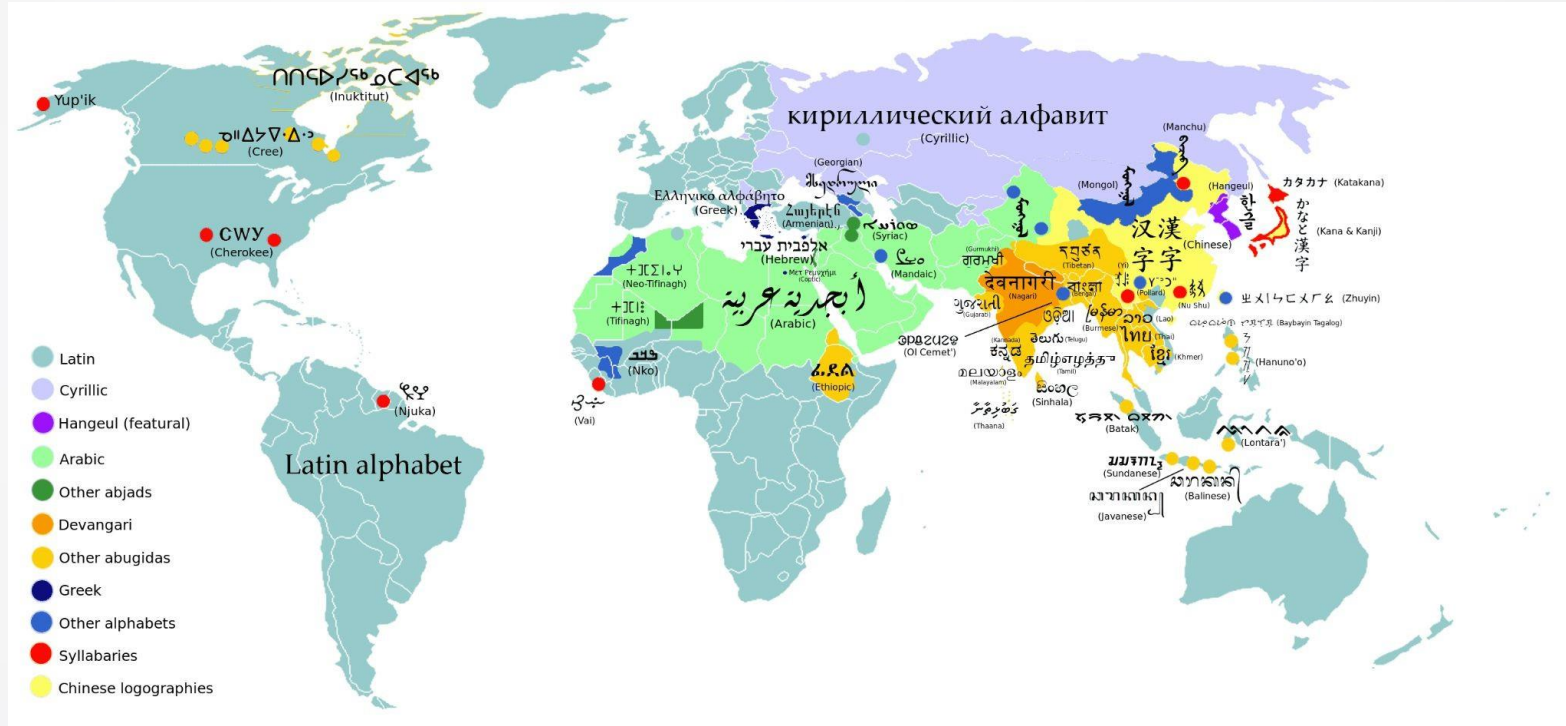
- Phonetic (accent)
- Morphological, Lexical (spelling)
- Syntactic
- Semantic



Diversity

- About **7000 languages** spoken in the world
- About **60%** are found in the **written form** (cf. Omniglot)

Graphemic Diversity





Syntactic Diversity

A key characteristics of the syntax of a given language is **the word order**

- **Word order differs** across languages
- **Word order degree of freedom** also differs across languages
- We characterize word orders with: **Subject (S) Verb (V) Object (O) order**



Word Order Freedom And Morphology



- Word orders freedom and morphology are usually related
- **The more freedom in word orders**
 - the less information is conveyed by word positions
 - the more information is carried by each word
 - **the richer the morphology**

French : *Je mange souvent du pain.*

English : *I often eat bread.*



Statistical Description of a Corpus

We describe statistically a corpus of 800 scientific articles

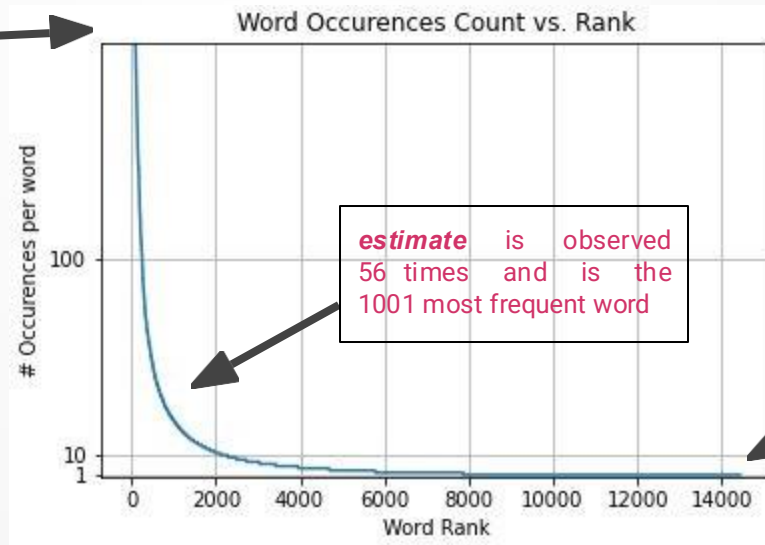
Question: If we plot the number of occurrences of each word vs. the rank, what will we observe?

Statistical Description of a Corpus



We describe statistically a corpus of 800 scientific articles

the is the most observed (rank 1) word with 8119 occurrences



About 6000 Words are observed only 1 time in the dataset (e.g. *stakeholders*, *pending*, *score*...)

Statistical Description of a Corpus



We describe statistically a corpus of 800 scientific articles

→ In a large enough corpus, word distributions follows *a Zipf Law* ie:

f_w frequency of entity w
 k frequency rank of entity w

$$f_w(k) \propto \frac{1}{k^\theta}$$

- Zipf law is a Power relation between the rank and frequency
*The most frequent entities are **much more frequent** than the less frequent ones*



Statistical Description of Language

Zipf Distributions are observed not only for words but with many other units of language (sounds, syntactic structure, name entities...)

Consequence

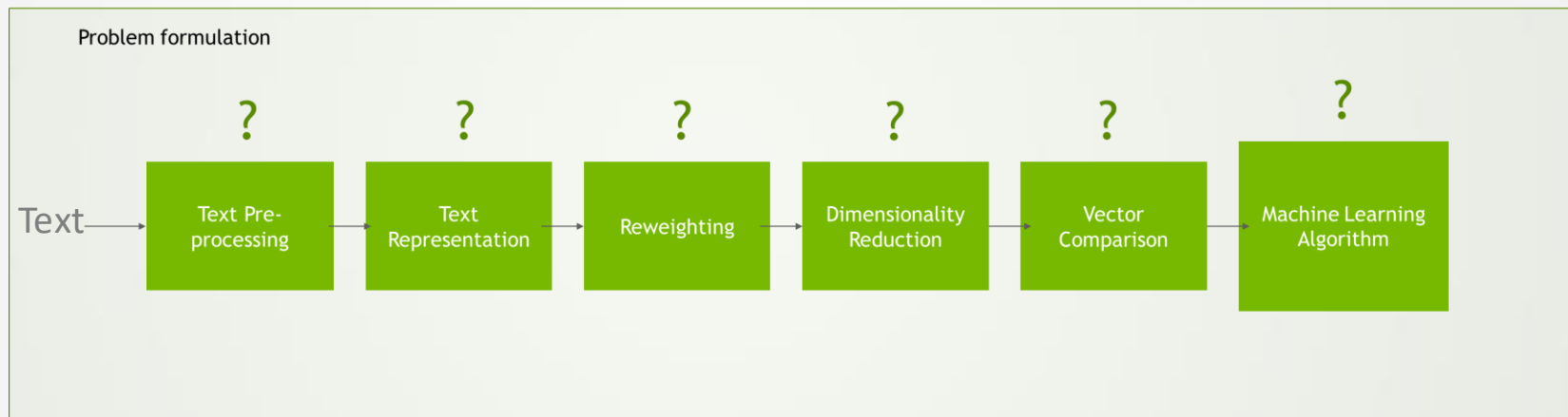
➡ A large number of units are observed in language with very low frequency i.e. **Sparsity**

➡ **Very challenging for NLP**

► Problem Formulation



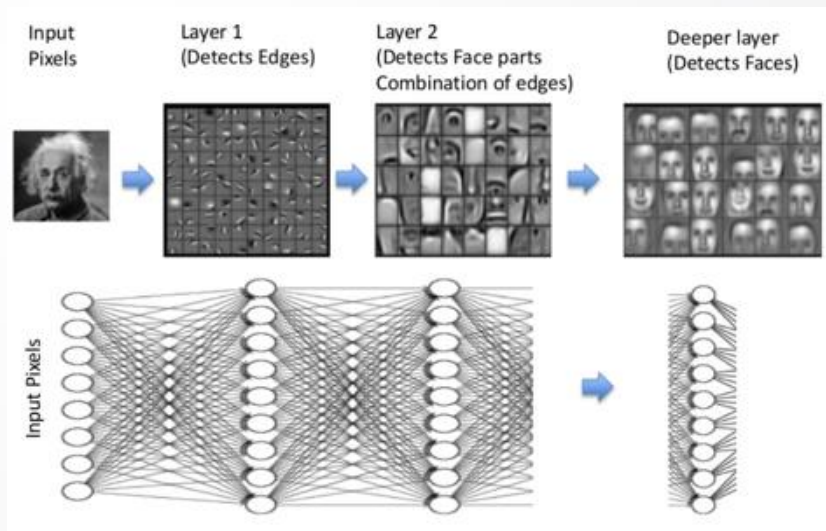
?



▶ Deep Learning Models : ANN



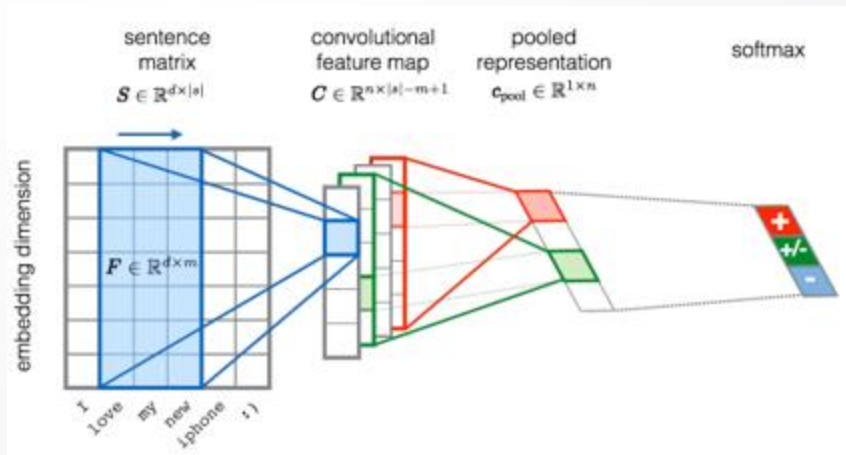
- **Challenges** : fixed input size.
- they often struggle with contextual understanding and require extensive training data.



► Deep Learning Models : CNN



- Convolutional Neural Networks are primarily designed for image processing, which limits their effectiveness in NLP tasks.
- Their local receptive fields can miss long-range dependencies in text, and they often require complex architectures to capture semantic meaning.





CNN limitations



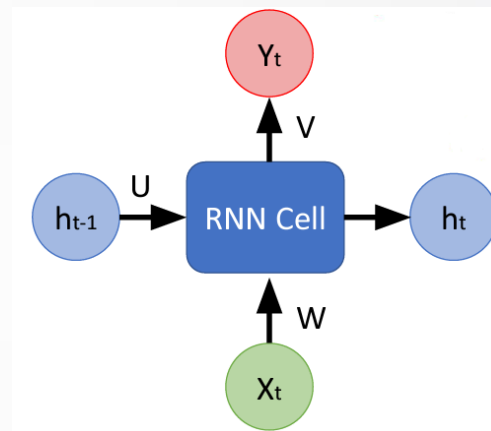
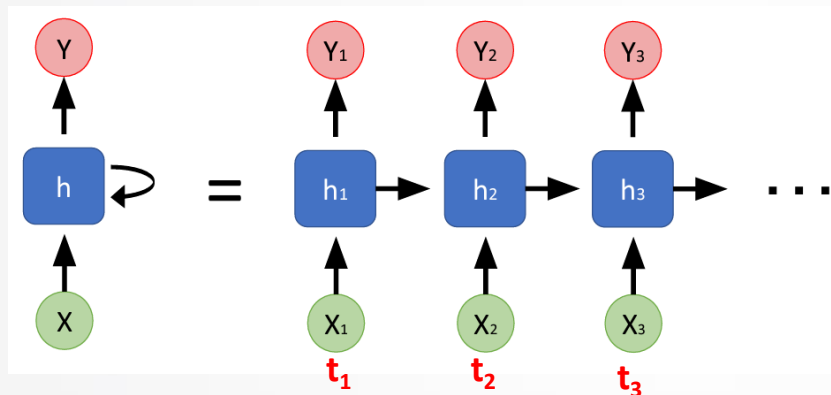
- **Fixed input size:** CNNs require a fixed input size. This can be a problem when dealing with sequences of varying length, such as in natural language processing or speech recognition.
- **Lack of memory:** CNNs have no memory of previous inputs, they can't easily capture temporal dependencies in sequential data.
- **Order invariance:** CNNs are order-invariant, they treat all inputs equally regardless of their position in the sequence. This makes it difficult for the network to capture the order-dependent patterns that are often present in sequential data.



Recurrent Neural network : RNN

- RNNs are designed to handle sequences of **varying length**
- They have memory that allows them to capture **temporal dependencies** in the data.
- They also have the ability to process inputs in a sequential order, which allows them to capture **order-dependent patterns**.
- RNNs have become the standard approach for processing sequential data such as speech, text, and time series data.

Graphical Representation of RNN



- X_i : Input at t_i
- h_i : hidden state at t_i
- Y_i : output at t_i

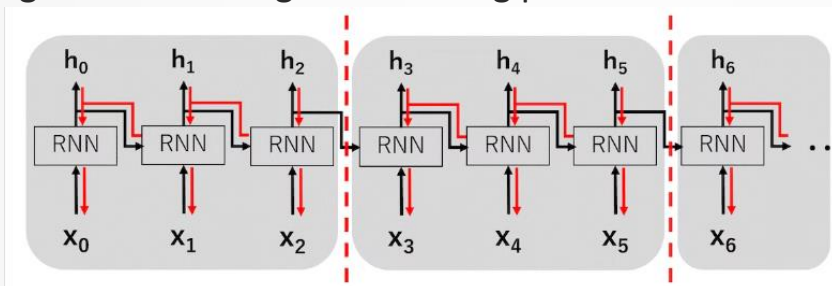
- $h_t = f(W \cdot x_t + U \cdot h_{t-1})$
- $y_t = f1(V \cdot h_t)$

at every time step t , the same set of weight parameters W , V , and U is used.

▶ Truncate BPTT



- Moving window through the training process



Forward propagation

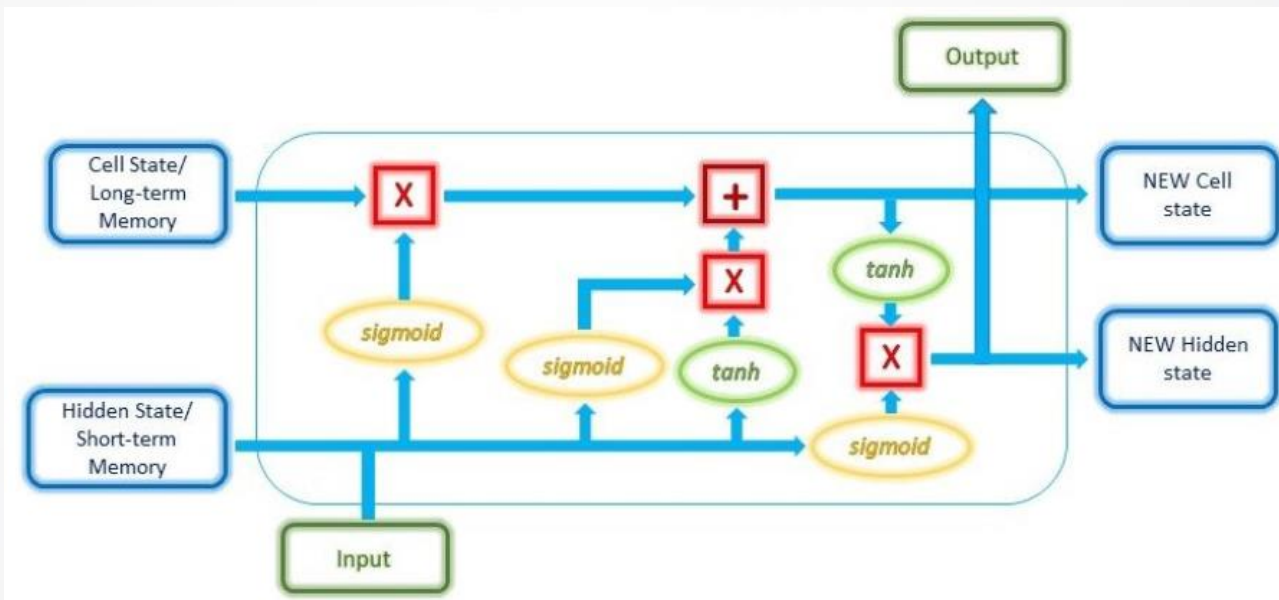


Backward propagation



- Advantages:
 - Help to avoid exploding/vanishing gradient
 - Much faster than the simple BPTT, and also less complex
- disadvantage:
 - dependencies of longer than the chunk length, are not taught during the training process.

▶ Long Short-Term Memory (LSTM)





Bidirectional RNNs

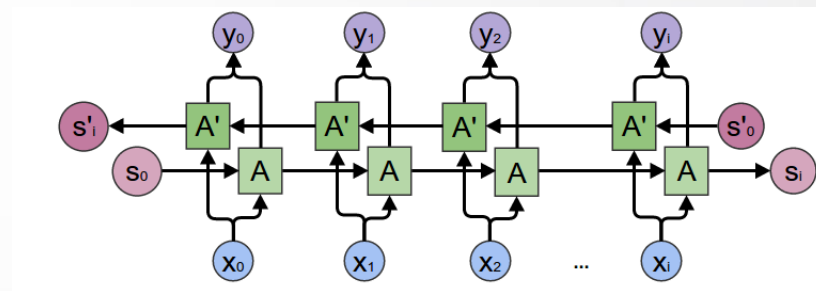
- **Bi-RNNs** are a type of recurrent neural network (RNN) that process input sequences **in both directions: forward and backward**.
- This allows Bi-RNNs to capture information from both the past and future context of a sequence, making them particularly useful for tasks that require understanding the entire sequence rather than just the preceding elements

Bidirectional RNNs

- To enable straight (past) and reverse traversal of input (future), Bidirectional RNNs, or BRNNs, are used. A BRNN is a combination of two RNNs

- one RNN moves forward, beginning from the start of the data sequence, and the other, moves backward, beginning from the end of the data sequence.

-The network blocks in a BRNN can either be simple RNNs, GRUs, or LSTMs.



RNN & LSTM limits



- Vanishing and Exploding Gradients
- Difficulty in Capturing Long-Term Dependencies
- Limited Parallelization
- Lack of Explicit Position Encoding
- ...

