60004198006

Sandeep Parmar

BE-B

①

Assignment 2

NLP

Q1. Steps involved in Summarization.

-> Text Summarization is the practice of breaking down long publications into manageable paragraphs or sentences. The procedure extracts important information while also ensuring that the paragraph's sense is preserved. This shortens the time it takes to comprehend long materials like research articles while without omiting critical information.

The process of constructing a concise, cohesive, and fluent summary of a lengthier text document, which includes highlighting text's important point, is known as text summarization.

types of Summarisation:-

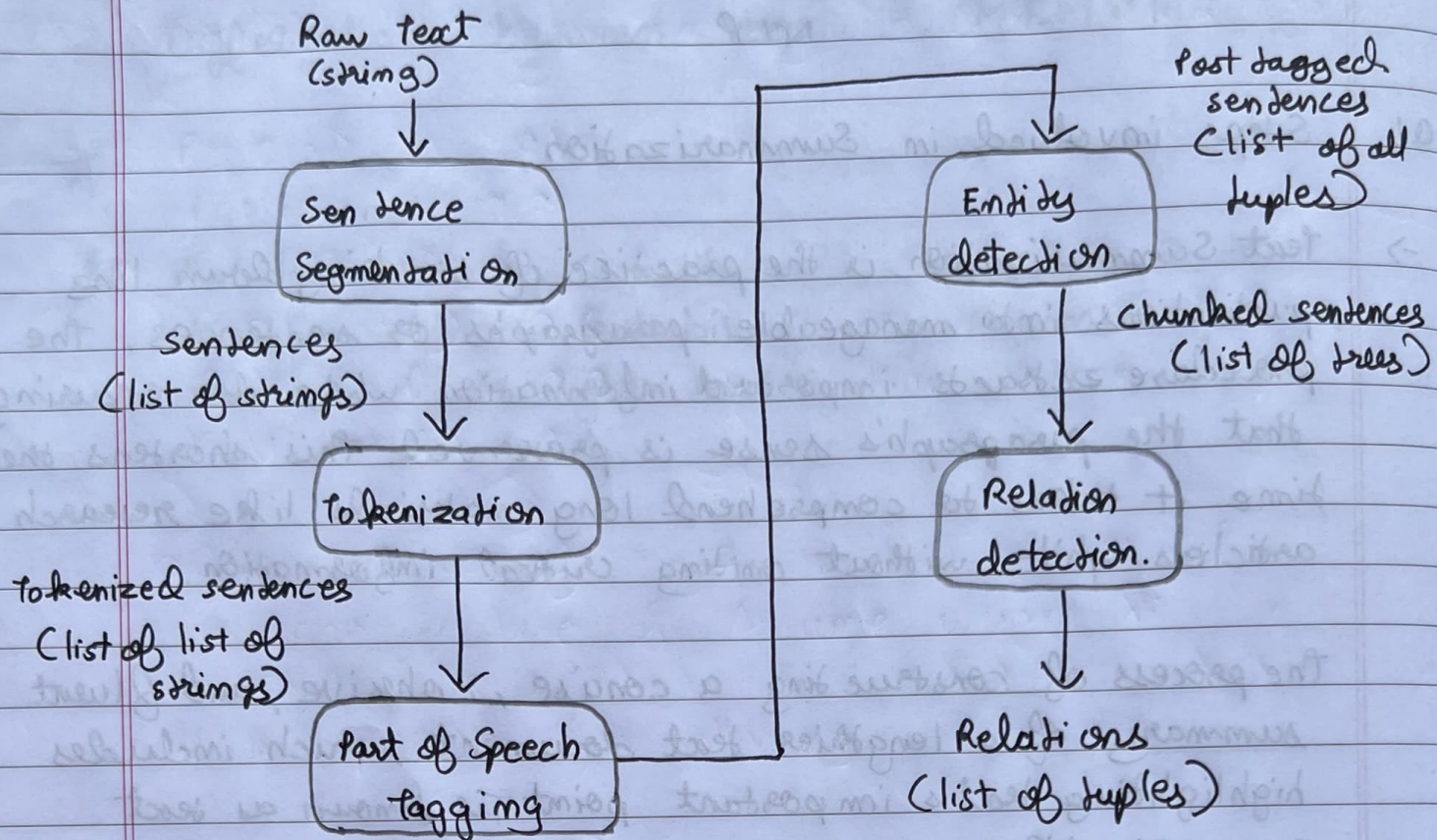
1. Extractive Summarisation:-

Extractive summaries are created by reusing portions of the input text document. The system extracts text from entire collection, without modifying the text document.

2. Abstractive summaries:-

Requires deep understanding and reasoning over the text. It provides own summary over input text without using same word or sentence in the input text. Determines the actual and short meaning of each element, such as words, sentences and paragraphs

(2)



Pipeline architecture of an Information extraction process.

→ Linguistic Preprocessing for Automatic Summarization.

→ Sentence Segmentation

Converts raw text into sentences - List of strings
- Sentence tokenizes

Input text: John own a car. It is a Toyota

Output: Segm1: John owns a car.

Segm2: It is a Toyota

→ ~~Part of Speech (P~~ Tokenization

Indentifies the word tokens from given sentence
- Provides a list of tokens as output - Word tokenize

Input: John owns a car.

Output: [[John], [owns], [a], [car], [.]]

classmate
Date
Page

→ Part of Speech [POS] tagging

Assigns appropriate part of speech tag to each word - POS is useful in extraction of nouns, adverbs, adjective, which provides some meaningful information about text - Generates a list of tuples with POS annotation.

Input: [[John], [owns], [a], [car], [.]]

Output: (NP (NNP John)), (VP (VBZ owns) (NP (DT a) (NN car))) (..)

→ Entity Detection

Identification of Predefined categories such as person, location, quantities, organizations etc - NER provides the entity detection for linguistic processing - NER system uses linguistic grammar based techniques and also statistical model to identify the entity.

Output: John → Person.

→ Relation Detection

Identifies the possible relation between two or more chunked sentences - Co-reference chain provides a relation between two or more sentences - Provides the link between pronouns and its corresponding nouns - Replacement of the pronouns with proper nouns.

Input text: John owns a car. It is a Toyota (In form of parse tree)

Output: "a car" → "a Toyota";
"it" → "a Toyota"

Q2. Consider Maximum Entropy Model for POS Tagging

P(D|a) = 0.9
P(N|man) = 0.9
P(V|sleeps) = 0.9
P(D|word) = 0.6
P(N|word) = 0.3
P(V|word) = 0.1

} For any word other than a, man, sleeps

Define features of Maximum Entropy Model
For each feature f_i Assume weight λ_i

Write expression for following probabilities.

i) P(D|cat)
ii) P(N | laughs)

→ Maximum Entropy Modelling for POS Tagging

$$P_\lambda(y|x) = \frac{1}{Z_\lambda(x)} \exp\left[\Sigma_i \lambda_i f_i(x, y)\right]$$

Where,
$Z_\lambda(x)$ is a normalising constant given by

$$Z_\lambda(x) = \Sigma_y \exp\left(\Sigma_i \lambda_i f_i(x, y)\right)$$

tag → D, N, V V = a, man, sleeps.

$$V' = V - \{a, man, sleeps\}$$

(5)

Features are as follows

b_1 } 1, word "a" and tag D
0, otherwise

b_2 } 1, word "man" and tag N
0, otherwise

b_3 } 1, word "sleeps" and tag V
0, otherwise

b_4 } 1, word ends with "ly" and tag D
0, otherwise

b_5 } 1, word → V and tag N.
0, otherwise

b_6 } 1, word → "V" and tag V
0, otherwise.

i) $P(D|cat) = \frac{e^{\sum \lambda_i f(x,y)}}{Z}$

$= \frac{e^{\sum \lambda_i b_i}}{Z}$

$\sum \lambda_i b_i = \lambda_1 0 + \lambda_2 0 + \lambda_3 0 + \lambda_4 1 + \lambda_5 0 + \lambda_6 0$

$= \underline{\lambda_4}$

(b)

$$P(D|cat) = \frac{\lambda_4}{z} = \frac{e^{\lambda_4}}{z}$$

$$z = \lambda_4 + \lambda_5 + \lambda_6$$

$$\therefore P(D|cat) = \frac{e^{\lambda_4}}{e^{\lambda_4} + e^{\lambda_5} + e^{\lambda_6}}$$

ii) $P(N|laughs) = \frac{e^{\Sigma \lambda_i f_i}}{z}$

$$\Sigma \lambda_i f_i = \lambda_1 0 + \lambda_2 0 + \lambda_3 0 + \lambda_4 0 + \lambda_5 1 + \lambda_6 0$$
$$= \lambda_5$$

$$P(N|laughs) = \frac{e^{\lambda_5}}{z}$$

$$z = \lambda_4 + \lambda_5 + \lambda_6$$

$$\therefore P(N|laughs) = \frac{e^{\lambda_5}}{e^{\lambda_4} + e^{\lambda_5} + e^{\lambda_6}}$$

Q3 Differentiate between HMM & MEMM.

→

HMM	MEMM
i) HMM is a Hidden Markovian Model	i) Maximum Entropy Markovian Model.
ii) HMM is a generative model	ii) MEMM is a discriminative model.
iii) HMM uses Join probability	iii) MEMM uses discrete probability
iv) In HMM probabilities are obtained by training on a text corpus	iv) MEMM builds a distribution by adding features, which can be hand picked or picked out by training.
v) HMM is less flexible	v) MEMM is more flexible.

Q4 Use Bean Search Algorithm with Bean size 2 to identify Highest Probability Tag Sequence for the Sentence "the light book"

→ the : [Det, Noun]
light : [Verb, Adj]
book : [Verb, Noun].

Assume $\lambda_i = 1$

→

"the"	"light"	"book"
Det = 1/2	Verb = e/e^3+e+2	Verb = $1/e^2+e+2$
Noun = 1/2	Adj = e^2/e^3+e+2	Noun = e^2/e^2+e+2

S1 $P(D|the) = \frac{e^{\lambda_5 f_7}}{z} = \frac{e^1}{z}$

$P(N|the) = \frac{e^{\lambda_8 f_8}}{z} = \frac{e^1}{z}$

Here, $z = e^{\lambda_7 f_7} + e^{\lambda_8 f_8}$
$= e^1 + e^1$
$= 2e$

$\therefore P(D|the) = \frac{e}{2e} = \frac{1}{2}$

$\therefore P(N|the) = \frac{e}{2e} = \frac{1}{2}$

S2 For word "light".

$$P(\text{verb} \mid D) = \frac{e^0}{z} = \frac{1}{z}$$

$$P(\text{verb} \mid N) = \frac{e^{\lambda_2 f_2}}{z} = \frac{e^1}{z}$$

$$P(\text{Adj} \mid D) = \frac{e^{\lambda_1 f_1 + \lambda_4 f_4 + \lambda_5 f_5}}{z} = \frac{e^3}{z}$$

$$P(\text{Adj} \mid N) = \frac{e^0}{z} = \frac{1}{z}$$

Here
$z = 1 + e^1 + e^3 + 1$
$= e^3 + e + 2$

$\therefore$ $P(\text{verb} \mid D) = \frac{1}{e^3 + e + 2}$

$P(\text{verb} \mid N) = \frac{e}{e^3 + e + 2}$

$P(\text{Adj} \mid D) = \frac{e^3}{e^3 + e + 2}$

$P(\text{Adj} \mid N) = \frac{1}{e^3 + e + 2}$

S3 for word "book"

$P(\text{verb} \mid \text{verb}) = \frac{e^0}{z} = \frac{1}{z}$

$P(\text{verb} \mid \text{Adj}) = \frac{e^0}{z} = \frac{1}{z}$

(10)

$$P(\text{Noun} \mid \text{Verb}) = \frac{e^{\lambda_6 \lambda_6}}{z} = \frac{e^1}{z}$$

$$P(\text{Noun} \mid \text{Adj}) = \frac{e^{\lambda_3 f_3 + \lambda_6 f_6}}{z} = \frac{e^2}{z}$$

Here $z = 1 + 1 + e^1 + e^2$

$= e^2 + e + 2$

$$\therefore P(\text{verb} \mid \text{verb}) = \frac{1}{e^2+e+2}$$

$$P(\text{verb} \mid \text{adj}) = \frac{1}{e^2+e+2}$$

$$P(\text{Noun} \mid \text{verb}) = \frac{e}{e^2+e+2}$$

$$P(\text{Noun} \mid \text{Adj}) = \frac{e^2}{e^2+e+2}$$

∴ Comparing the probabilities in table, the final tag sequence is

" the	light	book "
↓	↓	↓
Det	Adj	Noun

Q5 Explain Anaphoric Resolution, What is Anaphora?

→. Anaphora resolution (AR) which most commonly appears as pronoun resolution is the problem of resolving references to earlier or later items in the discourse. the items are usually noun phrases representing objects in real word called references but can also be verb phrases, whole sentences or paragraphs.

Reference to an entity that has been previously introduced in the discourse is called anaphora, and the referring expression used is said to be anaphoric.

there are three types of Anaphora.

a. Pronomial :- this is the most common type where a referent is referred by a pronoun. Example. "John found the love of his life" where 'his' refers to 'John'.

b. Definide noun phrase:- the antecedent is referred by a phrase of the form "<the><noun phrase>". Example: "this relationship did not last long", where 'this relationship' refers to 'the love' in the preceding sentence.

c. Quantifier / Ordinal: the anaphor is a quantifier such as 'one' or an ordinal such as 'first'. Example: "He started a new one", whe 'one' refers to 'the relationship'.