Word level Analysis (Module 2).

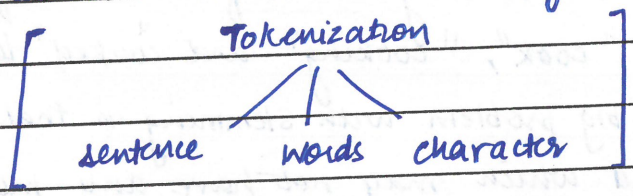
Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject: Natural Language Processing

Module 2: Word level Analysis

Syllabus: Tokenization, Stemming, Segmentation, Lemmatization, Edit distance, collocations, Finite Automata, Finite state transducers, Porta Stemmer, Morphological analysis, Derivational and Inflectional morphology, Regular expressions and its types, N-Grams (Unigrams / Bigrams), Language modelling, Corpora, Computing the probability of word sequence, Training and testing.

Tokenization: Tokenization is a common task of natural language processing Tokenization is a way of separating a piece of text into smaller units called as Tokens. Here tokens can be either sentence, word or character. Tokenization is broadly classified into



Tokens are the building blocks of natural language. for example Consider the sentence "Never give up" The most common way of forming tokens is based on space, assuming space as a delimiter the tokenisation of above sentence results in 3 tokens - Never-Give-up

Similarly tokens can be either characters or sub words for eg>

Character tokens: s-m-a-r-t-e-r

Subword tokens: Smart-er.

→ Q) Why is tokenization required?

·] Every sentence gets its meaning by the words present in it. So by analyzing the words present in the text we can easily interpret the meaning of the text.

·] Once we have the list of words we can also use the statistical tools and methods to get the insights into the text.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject: Natural Language Processing

Stemming: Stemming usually refers to the crude heuristic process that chops off the end of words in hope of achieving the goal to reduce inflectional forms and sometimes derivationally related forms of word to a common base form.

With stemming words are reduced to their word stems. Stemming algorithms are typically rule based, one can view them as a heuristic process that sort of lops off the ends of words.

For example: we may have a suffix rule that based on a list of known suffixes cuts them off, In English language we have suffixes like "ed" and "ing" which may be useful to cut off in order to map the words "cook", "cooking" and "cooked" to the same stem of cook. The big problem with stemming is that it produces the root word which may not have any meaning.

Challenges in stemming

Over stemming comes from when too much of the word is cut off This can result in non-sensical stems where all the meaning of the word is lost / or muddled. Or it can result in words being resolved to the same stems, even though they probably should not be. Understemming is the opposite issue it comes from when we have several words that actually are the forms of one another.

Lemmatization ① It is the process of grouping together the different inflected forms of a word so that they can be analysed as a single item.

② Lemmatization usually refers to doing things normally with the use of vocabulary and morphological analysis of words aiming to remove inflectional endings and return the base or dictionary form of the word, also known as lemma.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Lemmatization is a more calculated process, it involves resolving words to their dictionary form. Infact lemma of the word is its dictionary or canonical form. For lemmatization it requires an extra computational linguistic power such as part of speech. It is seen with a token 'saw' stemming might just return 's' however lemmatization would return see or saw depending on whether the use of token as a verb or a noun.

	Stemming	Lemmatization
1]	Stemming is faster because it chops words without knowing the context of word in a given sentence.	1] Lemmatization is slower as compared to stemming because it knows the context of word in a given sentence.
2]	It is a rule based approach	2] It is a dictionary based approach
3]	When we convert any word to its root form then stemming may create non existence meaning of word.	3] Lemmatization is more a calculated process, it involves resolving words to their dictionary form.
4]	Accuracy is less	4] Accuracy is more
5]	Stemming is preferred when the meaning of word is not important example: spam detection.	5] Lemmatization is preferred when the meaning of the word is important for analysis example: Question Answering system.
6]	for example : "Studies" ⇒ "Studi"	6] for example : 'studies' ⇒ "Study".

Segmentation: ① segmentation refers to dividing text into larger, coherent units such as sentences, paragraphs or even topics.

Purpose: ① To structure text into meaningful sections that preserve the flow and context of original content.
② To facilitate higher level text processing tasks.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

granularity → Segmentation typically operates at sentence level, paragraph level or document level.

Examples of Sentence segmentation:

Input : " Mr Smith bought a car. Its new and Shiny."

output: ["Mr Smith bought a car", "Its new and Shiny".]

Paragraph segmentation

Input: " This is the first paragraph. It talks about introduction. This is the second paragraph. It talks about details."

output: [" This is the first paragraph. It talks about introduction", "This is a second paragraph. It talks about details"].

Challenges of segmentation :

1] correctly identifying sentence boundaries, especially when dealing with abbreviations, punctuations and quotes.

2] Determining paragraph boundaries in documents without explicit markers.

3] Segmenting text into topics in a way that each segment is coherant and contextually relevant.

Tools:

NLTK

Spacy

Stanford NLP.

Differences

Tokenisation	Segmentation
1] Tokenisation focuses on breaking text into meaningful units.	1] Segmentation focuses on breaking text into larger coherent units.
2] Tokenisation focuses on finer granularity.	2] Segmentation deals with coarser granularity.
3] Tokenisation is foundation step for nlp tasks such as parsing & tagging.	3] Segmentation is essential for understanding structure and flow of large text bodies.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Edit Distance [MU 2023 Dec)] – (Important) –

Edit distance also known as Levenshtein distance is the measure of similarity between the two strings. It quantifies how many character edits (insertions, deletions or substitutions are required to transform one string to another.

Key concepts:

1. Insertions: Adding a character to the string.
2. Deletions: Removing a character from the string
3. Substitutions: Replacing one character with another.

Example:

Consider the transformation of word kitten to sitting.

kitten → sitten (substitute 'k' with 's')

sitten → sittin (substitute 'e' with 'e')

sittin → sitting ('insertion of 'g').

The edit distance is '3' [two substitutions and one insertions]

Applications of Edit distance:–

1. Spell checking: Find the closest correct word to a misspelled word.
2. Approximate String matching: Searching for a pattern in a text where the pattern may have errors.
3. Plagiarism Detection: comparing documents to find similarity.
4. DNA sequence analysis: Comparing genetic sequences for similarity and differences.
5. Machine Translation Evaluation: Comparing translated sentences to reference translations to measure translation quality.

Collocation: Collocations in NLP refer to the combinations of words that frequently occur together and exhibit a linguistic pattern or a relationship. These words pairs or groups often convey a particular meaning or context when used together which may not be evident when words are used separately.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Examples of collocations:

1. Adjective + noun: "strong tea", "heavy rain"
2. Verb + noun: "make a decision", "catch a bus".
3. Noun + noun: "data analysis", "coffee machine"
4. Adverb + adjective: "deeply concerned", "highly effective"
5. Verb + adverb: "run quickly", "speak softly".

Importance of collocations in NLP:

1. Natural language understanding: Recognizing collocations helps in understanding the context
2. Language modelling: collocations help to improve the performance of language models.
3. Machine Translations: Translating collocations is essential for preserving meaning across languages.
4. Text Generation: Generating natural sounding text requires an understanding of common word pairing.

Methods to identify collocations:—

1. Frequency based methods: ① simply counting the frequency of word pairs occuring together in the corpus.
   Advantages: 1. Easy to implement
   Disadvantages 2. High frequency alone does not account for significance common words may co-occur frequently by chance.
2. Statistical methods: Methods such as chi-square test, point wise mutual information, likelihood ratio test are used.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

3. Machine learning approaches: Using supervised and unsupervised learning to identify collocations. Word embeddings and neural networks are extensively used.

Regular Expressions: R.E is a language for specifying text search strings or search patterns. The regular expression languages are used for searching texts in UNIX. Usually search patterns are used by searching algorithms for find or find/replace operations on strings or for input validation. A regular expression is a formula in a special language that is used for specifying simple class of strings. Formally, a regular expression is an algebraic notation for characterizing the set of strings. Basically a regular expression is a pattern describing a certain amount of text.

→ Sums on Regular expressions

$0^* = \{\varepsilon, 0, 00, 000, 0000\}$ → zero or more occurences

$0^+ = \{0, 00, 000, 0000\}$ → one or more occurences of 0.

.] Write a R.E for a language accepting all combinations of $a \Rightarrow a^* \Rightarrow \{aa, aaa\}$

.] Design a R.E for the language accepting all strings having any number of a's and b's ⇒ $\{a+b\}^*$

.] Construct an R.E for the language accepting all strings which are ending with 00 over input 0,1 $(0|1)^*00$.

.] Input {a,b,c} in which any number of a's followed by any number of b's and c's $- a^* b^* c^*$.

.] Construct R.E for language which consist of atleast 2 b's over the string a,b. zero b → $(a)^*+\varepsilon$, one b → $(a)^*+b$, two b' $(a)^*+bb$.

$[(a^*+\varepsilon)+(a^*+b)+(a^*+bb)$.

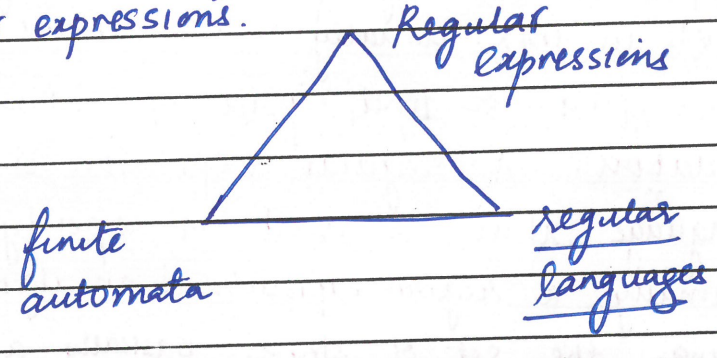
$a^*[\varepsilon+b+bb]$

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Finite Automata:—

The regular expression is more than just a convenient metalanguage for text searching. First a regular expression is one way of describing a finite state automaton (FSA). Both regular expressions and finite state automata can be used to describe regular expressions.



Finite automata (FA) also known as FSM are computational models used to represent and manipulate a set of states and transitions. In context of natural language processing, finite automata are employed to model and process sequences of characters, words or other linguistic units.

Key concepts of finite automata:—

1. States: Finite automata consist of finite number of states, one of these states are initial state and one or more can be designated as final state.
2. Alphabets: The set of symbols that the automaton can read In NLP, this could be characters, word or tokens.
3. Transitions: Rules that describe how to move from one state to the other.
4. DFA [Deterministic finite automation]: In DFA for each state for each input symbol there is exactly one transition.
5. NFA [Non deterministic finite automata]: In NFA for each state for each input symbol there can be zero, one or more multiple transitions.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Components of finite automaton:—

A finite automaton is defined by a 5 tuple $(Q, \Sigma, \delta, q_0, F)$

Q: A finite set of states

Σ: A finite set of alphabets

δ: A transition function $\delta: Q \times \Sigma \to Q$ for DFA and $\delta: Q \times \Sigma \to 2^Q$ for NFA.

q_0: initial state

F: The set of final (accepting) states.

FST [finite state transducers] extend the concept of finite automata by associating input symbols with output symbols, making them powerful tools for various tasks in natural language processing. FSTs are particularly useful for modelling and implementing transformations in text processing such as morphological analysis phonological rules and text to speech systems.

A FST is defined as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$

Q: finite set of states

Σ: A finite set of i/p alphabets

Γ: A finite set of o/p symbols.

δ: A transition function $Q \times \Sigma \to Q \times \Gamma$

q_0: initial state

F: The set of final (accepting) states.

FST as a recognizer; FST as a generator, FST as a translator, FST as a set relator, these are roles of finite state transducers.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Morphological parsing with finite state transducers:

- The objective of morphological parsing is to produce output lexicons for a single input lexicon
- The term morphological parsing is related to parsing of morphemes.
- We define morphological parsing as the problem of recognizing that a word breaks down into smaller meaningful units called as morphemes producing some sort of linguistic structure for it.
- For example we can break the word foxes into two fox and es

In other sense morphology is the study of:

- The formation of words
- The origin of words
- grammatical forms of words.

Types of morphemes:—

- Stems
- Word order

Stems: It is core meaningful unit of word. We can also say it is root of the word. for example in the word foxes the stem is fox.

- Affixes: As the name suggest they add some additional meaning and grammatical functions to the words.

Affixes can be divided into four types:—

- Prefixes — Prefixes precede the stem, for example in the word unbuckle (un → prefix)

Suffixes — Suffixes follow the stem for example in the word cats, -s is suffix.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

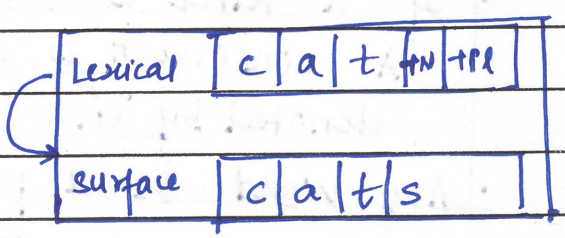
Subject: Natural Language Processing

Morphological parsing with FST:→

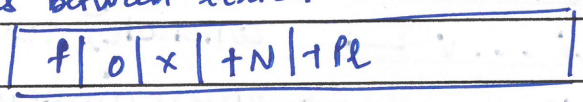
The FST is a multi function device and can be viewed in following ways:-

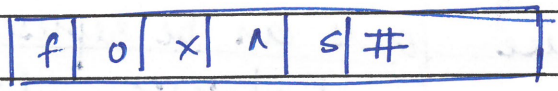
- **Translator**: It reads one string on one tape and outputs another string
- **Recognizer**: It takes a pair of strings as two tapes, and accepts/rejects based on matching.
- **Generator**: It outputs a pair of strings on two tapes along with yes/no result.
- **Relator**: It compares the relations between two strings

Input	Parsed output
cat	cat +N+SG
cats	cat+N+PL
geese	goose+N+PL
reading	read+N+present participle



FST-1 translates between lexical and intermediate level.





Orthographic rules for FST. — Language English often requires spelling changes at morpheme boundaries by introducing spelling rules (or orthographic rules)

Name	Description of rule	Example
consonant doubling	1 letter consonant doubled before -ing	beg/begging
E deletion	silent 'e' dropped before -ing	make/making
E insertion	e added after -s, -z, -x, -ch	watch/watches.
y replacement	y changes to ie before s	try/tries
K insertion	verbs ending with vowel + c add k	panic/panicked.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject: Natural Language Processing

Porter's stemming Algorithm [Imp, MU '23 (May, Dec), MU '22 (May)]

- The porter's stemming Algorithm is used to remove the suffixes from an English word and obtain its stem which is very useful in field of NLP
- This process reduces the number of terms kept by an NLP system and will be very advantageous.
- A consonent is a letter other than vowel and other than 'y' preceded by a consonant.
- If a letter is not a consonant it is a vowel
- A consonant is denoted by letter 'c' and a vowel is denoted by 'v'.
- Any word or a part of word therfore has one of the four forms given below :-
  - cvcv ... c → management, collection
  - cvcv ... v → conclude, revise
  - vcvc ... c → entertainment, illumination
- All of these forms can be represented using a single form as [c] vcvc ... [v]

Here square brakets denote the arbitrary presence of consonants or vowels.

$(vc)^m$ → denotes vc repeated m times so we can have expression written as

$[c]\ (vc)^m [v]$

- The value 'm' found in the above expression is called as the measure of any word or word part when represented in the form $[c](vc)^m v$

Some examples for different values of m are:

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

TREE
↓ ↓ ↓ ↓
C C V V
[C][CV]^{m}[V]
m=0

T R O U B L E
↓ ↓ ↓ ↓ ↓ ↓ ↓
C C V V C C V
[C][VC]^{m}[V]
m=1

The rules for replacing a suffix will be given in the following form:—

(condition) S1 → S2

This means that if a word ends with S1 (suffix) and stem before S1 satisfies the given condition S1 is replaced by S2.
This condition is usually given in terms of 'm' in regards to S1 (m>1) ement → S1 is ement and S2 is null
This would map REPLACEMENT to REPLAC, since REPLAC is a word part for which m=2.

Conditions:

1. m → measure of the stem, m=k or m>k where k is an integer
2. *X ← stem ends with a given letter X.
3. *V* ← The stem contains a vowel.
4. *d ← The stem ends in a double consonant
5. *O ← The stem ends in a consonant-vowel-consonant.

• Rules are divided into sets and in each successive step one set of rules are applied.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Porter steps:

- Each step corresponds to the set of rules. The rules in the step are examined in a sequence and only one rule from a step can apply.

```
{ step 1 a (word);
  step 1 b (stem);
  if (step second or third rule of step (1b) was used)
        step 1b1 (stem);
  step 1c (stem);
  step 2 (stem);
  step 3 (stem);
  step 4 (stem);
  step 5a (stem);
  step 5b (stem);
}
```

Porter rules:

Step 1a (rules)

Conditions	Suffix	Replacement	Examples
NULL	sses	ss	caresses → caress
NULL	ies	i	ponies → poni
NULL	ss	ss	caress → caress
NULL	s	Null	cats → cat

Step 1b (rules)

Conditions	Suffix	Replacement	Examples
(m>0)	eed	ee	feed → feed
(*v*)	ed	Null	plastered → plaster
(*v*)	ing	Null	motoring → motor

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Step2 (rules).

conditions	Suffix	Replacement	Examples
(m>0)	ational	ate	relational→relate
(m>0)	enci	ence	valenci→valence
(m>0)	tional	tion	conditional→condition
(m>0)	izer	ize	digitizer→digitize
(m>0)	alism	al	feudalism→feudal
(m>0)	iviti	ive	sensitiviti→sensitive
			(Refer ppt for more rules of step2)

Step3, Step4, (Refer ppt for more rules of step2, step3, step4)...

Conditions	Suffix	Replacement	Examples
(m>0)	icate	ic	triplicate→triplic
(m>0)	ative	Null	formative→form
(m>0)	alize	al	formalize→formal

Step4 (Refer ppt for rules 4)..

conditions	Suffix	Replacement	Examples
(m>1)	al	null	revival→reviv
(m>1)	ance	null	allowance→allow

Step5 (Refer ppt for more rules 5)..

Conditions	Suffix	Replacement	Examples
(m>1)	e	null	probate→probat
(m>1)	e	null	cease→ceas

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Example:

computers —step1a→ computer —step4→ comput

singing —step1b→ sing.

Porter mishaps:—

gas (noun) → ga

gases (plural) → gase

gasses (verb, present tense) → gass

gaseous (adjective) → gaseou

* This is not good — all these words should ideally reduce to the same stem.

Tradeoff: More rules (accurate but slow) vs Less rules (efficient but sometimes wrong).

* Google does give different results for gas and gases, so may be they use these porter rules...

Morphology: (Imp).

- Morphology is the study of the way words are build up from smaller meaning bearing units, morphemes
- A morpheme is defined as minimial meaning bearing unit in a language.
- For example the word 'for' consist of a single morpheme while the word 'cats' consist of two morphemes morpheme 'cat' and morpheme 's'.
- Morphemes are divided into two types

Bound morphemes

Free morphemes

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject: Natural Language Processing

Bound morphemes: These are ones that cannot appear as a word by itself. eg) -s (dogs), (quickly)-ly, -ed (walked).
Free morphemes: These can appear as words by themselves... and can often combine with other morphemes.
eg) house (house-s), walk (walk-ed), of, or...

- Two broad classes of morphemes are → Stem, Affixes

Stem → main morpheme of the word.
Affixes → add additional meanings of various kinds

Affixes are further classified into → prefixes, suffixies, circumfixes, infixes.

prefixes → precede the stem.
suffixes → follow the stem
circumfixes → precede as well as follow the stem
infixes → inserted inside the stem.

Types of morphology (V.V.Imp) –

→ Morphology can be classified into two types

Inflectional morphology Derivational morphology.

Inflectional morphology: It is a combination of a word stem with a grammatical morpheme which results in a word of same class as original stem. In English inflection is simple only nouns, verbs, adjectives can be inflected. Eg cat → cats, mous → mice, walk → walking etc.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Derivational morphology: It is a combination of a word stem with a grammatical morpheme which results in a word of different class. In English it is very hard to predict the meaning of stem from its derived structure.
Eg appoint → appointee clue → clueless, kill → killer etc

	Inflectional Morphology	Derivational Morphology
1.	Inflection may or may not be affected by affixation, re-duplication, modification of bases and other morphological process	1. Derivation may be affected by formal means like affixation reduplication or modification of bases and other morphological process.
2.	Inflection prototypically serves to modify the lexemes to fit a different grammatical context.	2. ~~Defin~~ Derivation serves to create new lexemes.
3.	Inflection typically adds grammatical information	3. Derivation changes the category (eg takes a verb like employee and make it a noun employment).
4	Inflection prefix is to change the category but does not add a new meaning eg ~~un~~happiness	4. Derivation adds a substantial new meaning. eg unhappy.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject: Natural Language Processing

N gram model. (V.V.Imp). — (10M) [MU'23 (May, Dec), MU'22 (Dec)].

- In natural language processing n gram is a contagious sequence of n items generated from a given sample of text
- A statistical language model is a probability ~~sequence~~ distribution P(s) over all possible word sequence. The dominant approach in a statistical language modelling is the Ngram model.
- The goal of a statistical language model is to estimate the probability of a sentence.
- This is achieved by decomposing sentence probability into a product of conditional probabilities using chain rules as follows -

$$P(s) = P(w_1, w_2, w_3 \dots w_n)$$

$$P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_2 w_1) \cdot P(w_4|w_3 \cdot w_2 w_1)$$

$$= \prod_{i=1}^{n} P(w_i|h_i)$$

where h_i ⟶ history of word w_i

- Inorder to calculate sentence probability we need to calculate the probability of word, given sequence of words preceding it.
- An n gram model simplifies the task by approximately the probability of a word given all the previous words by conditional probability given previous (n-1) words only.

$$P(w_i|h_i) = P(w_i|w_{i-n+1} \cdot w_{i-1})$$

n gram model calculates $P(w_i|h_i)$ by modelling language as Markov model of order n-1 ie by looking at previous n-1 words only.

A model that limits the history to previous one word is termed as bigram model and previous two words is termed as trigram model.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Using bigram and trigram estimate the probability of a sentence :—

$$P(s) = \prod_{i=1}^{n} P(w_i | w_{i-1})$$

$$P(s) = \prod_{i=1}^{n} P(w_i | w_{i-2}\, w_{i-1})$$

- A special <s> tag is used to mark the beginning of the sentence in a bigram estimation.
- The probability of first word in a sentence is a condition on <s>. Similarly in trigram estimation we introduce two pseudo words <s1> <s2>

$$P(w_i | w_{i-n+1} \ldots w_{i-1}) = \frac{C(w_{i-n+1} \ldots w_{i-1}, w_i)}{\sum_{w} C(w_{i-n+1} \ldots w_{i-1}, w)}$$

- The sum of all n grams that share (n-1) words is equal to the count of common prefix, w_{i-n+1} hence the previous expression can be rewritten as follows:—

$$P(w_i | w_{i-n+1} \ldots w_{i-1}) = \frac{C(w_{i-n+1}, \ldots w_{i-1}, w_i)}{\sum C(w_{i-n+1}, \ldots w_{i-1})}$$

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Numericals on N-grams (Imp) ** [MU22'(May, Dec), MU23'(May, Dec)]

(1) Find the probability of : The Arabian knights are fairy tales of east.

Training set:

<S> The Arabian knights
<S> These are the fairy tales of the east
<S> The stories of the Arabian knights are translated in many languages.

P (The|<S>) = 2/3 = 0.67
P(Arabian|the) = 2/5 = 0.4
P(knights|Arabian) = 2/2 = 1.0
P(are|these) = 1/1 = 1
P(the|are) = 1/2 = 0.5
P(fairy|the) = 1/5 = 0.2
P(tales|fairy) = 1/1 = 1
P(the|of) = 2/2 = 1
P(east|the) = 1/5 = 0.2
P(stones|the) = 1/5 = 0.2
P(of|stones) = 1/1 = 1

P(are|knights) = 1/2 = 0.5
P(translated| are) = 1/2 = 0.5
P(in|translated) = 1/1 = 1
P(many|in) = 1/1 = 1
P(languages|many) = 1/1 = 1

Test sentence: The Arabian knights are the fairy tales of the east

P(the|<S>) × P(Arabian|the) × P(knights|Arabian) × P(are|knights) × P(the|are) × P(fairy|the) × P(tales|fairy) × P(of|tales) × P(the|of) × P(east|the).

= 0.67 × 0.5 × 1 × 1 × 0.5 × 0.2 × 1 × 1 × 1 × 2 = 0.0067

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Example on Ngram model.

<s> I am Sam </s>
<s> Sam I am </s>
<s> Sam I like </s>
<s> Sam I do like </s>
<s> do I like Sam </s>

(do|<s>) = 1/5 = 0.2
(I|do) = 1/2 = 0.5
(like|I) = 2/5 = 0.4
(Sam|like) = 1/3 = 0.33
(</s>|Sam) = 2/5 = 0.4

(I|<s>) = 1/5 = 0.2
(am|I) = 2/5 = 0.4
(Sam|am) = 1/2 = 0.5
(</s>|Sam) = 2/5 = 0.4

<Sam|<s>) = 3/5 = 0.6
(I|Sam) = 3/5 = 0.6
(am|I) = 2/5 = 0.4
(<s>|am) = 1/2 = 0.5

(Sam|<s>) = 3/5 = 0.6
(I|Sam) = 3/5 = 0.6
(like|I) = 2/5 = 0.4
(</s>|like) = 2/3 = 0.66

(Sam|<s>) = 3/5 = 0.6
(I|Sam) = 3/5 = 0.6
(do|I) = 1/5 = 0.2
(like|do) = 1/2 = 0.5
(</s>|like) = 2/3 = 0.66

Prediction of the next word

1] <s> Sam ___

(Sam|<s>) has the probability of 0.6 the next highest probability of bigram having Sam on right hand side is (I|Sam) = 0.6, ∴ The next predicted word after Sam will be I.

2] <s> Sam I do

(do|I) have the probability 0.2, the next highest probability of bigram having do on right hand side is (I|do) = 0.5 and (like|do) = 0.5 So next predicted word is like or I.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

Numerical ③ on NGram (MU 2023, Dec) Given the following corpus find the probability of occurence of test data <s> Students are from Vellore </s>

Corpus:

<s> I am from Vellore </s>

<s> I am a teacher </s>

<s> Students are good and are from various cities </s>

<s> Students from Vellore do engineering </s>.

P(Student|<s>)* P(are|students)* P(from|are)* P(Vellore|from)* P(</s>|Vellore)

$$P(\text{Students}|<s>) = \frac{\text{count}(<s>\ \text{students})}{\text{count}(<s>)} = \frac{2}{4} = \frac{1}{2}$$

$$P(\text{are}|\text{students}) = \frac{\text{count}(\text{students are})}{\text{count}(\text{students})} = \frac{1}{2}$$

$$P(\text{from}|\text{are}) = \frac{\text{count}(\text{are from})}{\text{count}(\text{are})} = \frac{1}{2}$$

$$P(\text{Vellore}|\text{from}) = \frac{\text{count}(\text{from Vellore})}{\text{count}(\text{from})} = \frac{2}{3}$$

$$P(</s>|\text{Vellore}) = \frac{\text{count}(\text{Vellore }</s>)}{\text{count}(\text{Vellore})} = \frac{1}{2}$$

$$P(\text{Test data}) = \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{2}{3} * \frac{1}{2} = 0.0416.$$

Numerical ④ on NGram: Corpus: The girl bought a chocolate
The boy ate a chocolate
The girl bought a toy
The girl played with the toy.

(Choose from the options that fills)

Input: The girl _____

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Subject:Natural Language Processing

a) The gul bought
b) The gul played
out of these (2) which has the highest probability.

$$P(\text{bought} \mid \overset{\text{the}}{\text{gul}}) = \frac{\text{count (The gul bought)}}{\text{count (The gul)}} = \frac{2}{3} = 0.67$$

$$P(\text{played} \mid \text{the gul}) = \frac{\text{count (The gul played)}}{\text{count (the gul)}} = \frac{1}{3} = 0.33$$

probability of P(bought | the gul) > probability of P(played | the gul)
0.67 > 0.33

∴ option a) The gul bought fills the blank.

List of previous years university Questions —

Q1. Explain in brief Inflectional and Derivational morphology with suitable examples?
Q2 Explain Porter's Stemming Algorithm ?
Q3. Explain Edit distance algorithm with an example? Show the working of minimum number of operations required to transform "kitten" to sitting ?
Q4. Explain difference between Stemming and Lemmatization?
Q5 Consider the following corpus:
<s> She asks you to wait patently </s>
<s> He wants me to help him </s>
<s> They expect us to arrive early </s>
List all possible bigrams. Compute the Conditional probabilities and predict the next word to.