Module 3: Syn

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 3: Syntax Analysis

Syntax Analysis: Part of speech (POS) - Open and closed words. Tag set for English (Penn Tree Bank), Rule based POS Tagging, Transformation based Tagging, Stochastic POS Tagging and Issues - Multiple Tags and words, unknown words. Introduction to cfg, hidden markov model (HMM), maximum Entropy and conditional random field (CRF).

- Part of speech Tagging: Part of speech tagging is a process of assigning a part of speech (such as noun, verb, pronoun, preposition, adverb and adjective) to each word in a sentence. The input to the tagging algorithm is the sequence of words of a natural language sentence and specified tag sets. The output is a single best tag for each word. For example, the English word 'book' can be a noun as in 'I am reading a good book' or a verb as in 'The police booked the snatcher'. The same is true for other languages. For example the Hindi word 'sona' may be/mean 'gold' (noun) or 'sleep' (verb). However only one of the possible meanings is used at a time. No tagger is efficient enough to identify the correct lexical category of each word in a sentence in every case. The tag assigned by a tagger is the most likely for a particular use of word in a sentence. The collection of tags used by a particular tagger is called as tag set. Most part of speech tag sets make use of some basic categories ie, noun/verb/adjective and prepositions. Most tag sets capture morpho-syntactic information such as the singular/plural, number, gender, tense etc. Consider the following sentences: for eg
  Ziha eats an apple daily.
  Aman ate an apple yesterday.
  They have eaten all apples in basket.

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

The word 'eat' has distinct grammatical form in each of the four sentences. The tag assigned by the tagger is the most likely for a particular use of word in a sentence.

The collection of tags used by a particular tagger is called as tag set. Most of the part of speech tags make use of the basic categories ie noun, verb, adjective and prepositions.

Open and closed words :→

→ Words are classified into categories called as part of speech, these are sometimes called as word classes or lexical categories These lexical categories are defined by their syntatic and morphological behaviours. Word classes are further categorized by 'open and closed' class.

Open word class: Open class is one that commonly accepts the addition of new words. It includes nouns, verbs and adjectives. Open word/vocabulary reveals more specific and concrete patterns across a broad range of content domains, for better address of ambiguous word senses. These are less prone to misinterpretation

- Open class distinction is related the distinction between lexical and functional categories.
- Open classes are generally lexical categories in strict senses.

Closed word class:- A closed class is the one in which new items are rarely added. This includes pronouns, conjunctions. closed classes are much smaller. closed classes are normally functional categories, consisting of words that perform grammatical functions.

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

Rule Based POS Tagging:- ① Rule based taggers use hand-coded rules to assign tags to words. These rules use lexicon to obtain a list of candidate tags and then use rules to discard incorrect tags. If a word has more than one possible tag then rule-based taggers use hand-written rules to identify the correct tag.

* Rule based tagging can handle any disambiguity by analyzing the linguistic features of a word. And that is done by its preceding as well as following words.

— for example if the preceding word of a word is article or adjective then the word must be a noun.

— These rules may be either:-

1] Context pattern rules

2] Regular expression compiled into finite automata.

POS tagging (Rule-based) can be visualised by its two stage architecture:-

•] First stage: Here dictionary is used to assign each word a list of potential part of speech.

•] Second stage: Here the method uses large list of hand-written disambiguation rules to sort down the list to a single part of speech for each word.

Properties of Rule-based POS Tagging:-

- These taggers are knowledge-driven taggers.
- The rules in Rule-based-POS tagging are done manually.
- There are around 1000 number of rules.
- Smoothing and language modelling are defined in rule based taggers and it is done explicitly.

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

Stochastic Tagger:

1. The standard stochastic tagger algorithm is the HMM tagger
2. A markov model applies a simplifying assumption that the probability of chain of symbols can be approximated in terms of its parts or n-grams.
3. The simple n-gram is the unigram model which assigns the most likely tag to each token.
4. Stochastic taggers have data driven approaches in which frequency based information is automatically derived from the corpus and it used to tag words.
5. Stochastic taggers disambiguate words based on a probability that a word occurs with a particular tag.
6. The simplest scheme is to assign the most frequent tag to each word.
7. An early example of stochastic tagger was CLAWS (constituent likelihood automatic word-tagging system). Hidden markov model (HMM) is the standard stochastic tagger.

→ The unigram model needs to be trained using a tagged training corpus before it can be used to tag data.

For example: Consider the following sentence –

She had a fast

Muslims fast during Ramadan

Those who were injured in the accident need to be helped fast.

In the first sentence fast is used as a noun, in the second it used as a verb and in the third, an adverb.

→ A bigram tagger uses the current word and tag of the previous word in the tagging process.

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

It is also called as HMM tagger because it uses two layers of states a visible layer corresponding to input words and hidden layer learnt by a system corresponding to input words and a hidden layer learnt by a system corresponding to the tags.

→ The HMM makes use of lexical and bigram probabilities estimated over a tagged training corpus inorder to compute the most likely tag sequence, for each sequence.

→ One way to store the statistical information to built a probability matrix. The probability matrix contains both the probability that an individual word belongs to a word class as well as ngram analysis.

— This matrix is used to drive HMM tagger while tagging an unknown text.

— Let W be the sequence of words

$W = W_1 W_2 \ldots W_n$

The task is to find the tag sequence

$T = t_1 t_2 \ldots t_n$

which maximises $P(T|W)$ ie

$T' = \text{argmax}_T P(T|W)$

Applying Bayes rule:

$$P(T|W) = P(W|T) * P(T) / P(W).$$

$$P(t_i | t_{i-2} t_{i-1}) = \frac{c(t_{i-2}, t_{i-1}, t_i)}{c(t_{i-2}, t_{i-1})}.$$

for example: Consider the sentence: The bird can fly and the tag sequence

DT NNP MD VB

Using bigram approximation, the probability

$P\left(\begin{matrix}DT & NNP & MD & VB \\ | & | & | & | \\ The & bird & can & fly\end{matrix}\right)$ can be computed as

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

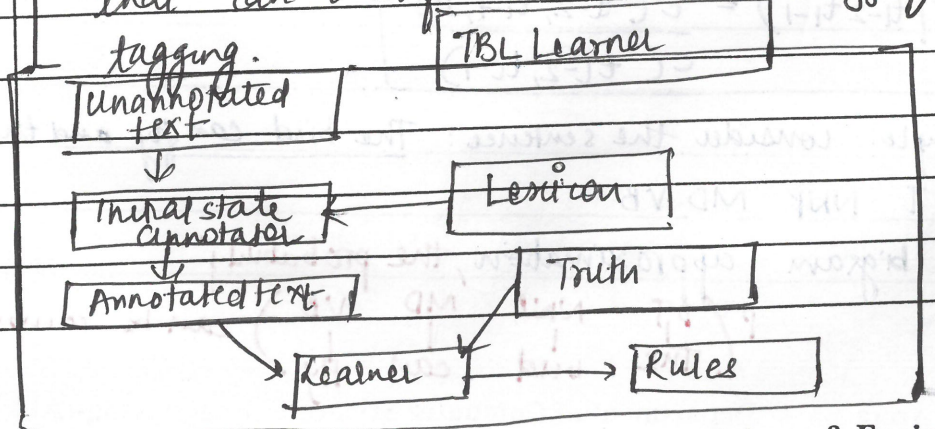
$= P(DT) \times P(NNP|DT) * P(MD|NNP) * P(VB|MD) * P(the|DT) * P(bird|NNP) \times P(can|MD) \times P(fly|VB)$.

→ **Hybrid Taggers:**

- Hybrid approaches to tagging combines the features of both rule based and stochastic approaches.
- They use rules to assign tags to words.
- Like stochastic taggers this is a machine learning technique and rules are automatically induced from data.
- Transformation-based learning (TBL) of tags is also called as Brill Tagging is an example of Hybrid approach.
- Like most HMM taggers, TBL is also a supervised learning technique

Steps involved in TBL

The input to Brill's TBL is a tagged corpus and a lexicon
The initial state annotator uses lexicon to assign the most likely tag to each word as the start state.
An ordered set of transformation rules are applied sequentially
The rule that results in the most improved tagging is selected. A manually tagged corpus is used as the reference for truth. The process is iterated until some stopping criteria is reached. The output of the algorithm is a ranked list of learned Transformation that can transform the initial tagging to correct tagging.



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)

Issues with Tagging: (POS Tagging)

* **Multiple Tags and multiple words**

→ Two issues that arise in tagging are tag indeterminacy and multipart words

→ Tag indeterminacy arises when a word is ambiguous between multiple tags and it is impossible and very difficult to disambiguate. In this case some taggers allows the use of multiple tags.

→ The second issue concerns multipart words. The C5 and C7 tagsets for example allow the prepositions 'in terms of' to be treated as a single word by adding numbers to each tag.

* **Unknown words:**

Unknown words are the words that do not appear in a dictionary or training corpus. They create a problem during tagging.
There are several potential solutions to this problem, one is to assign the most frequent tag to the unknown word.

— Another solution is to assume that the unknown words can be of any part of speech and initialize them by assigning open class tags. Then proceed to disambiguate them using probabilities of those tags. We can also use morphological information such as affixes to guess the possible tag of an unknown word.

— In this approach unknown word is assigned a tag based on the probability of the words belonging to a specific part of speech in training corpus having the same suffix or prefix.

Out of vocabulary (OOV) words:—

→ Out of vocabulary (OOV) are terms that are not the part of the normal lexicon found in the natural language processing environment

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

Introduction to cfg:—

Cfg is a set of rules that define a set of all well formed sentences.

- Context free grammar (cfg) was first defined for natural language by Chomsky (1957) and was used for Algol programming language by Backus and Naur
- A cfg (also called as phrase structure grammar) consist of four components:—
  1] A set of non terminals.
  2] A set of terminals.
  3] A designated start symbol (S)
  4] A set of production rules (P) eg $A \rightarrow \alpha$.

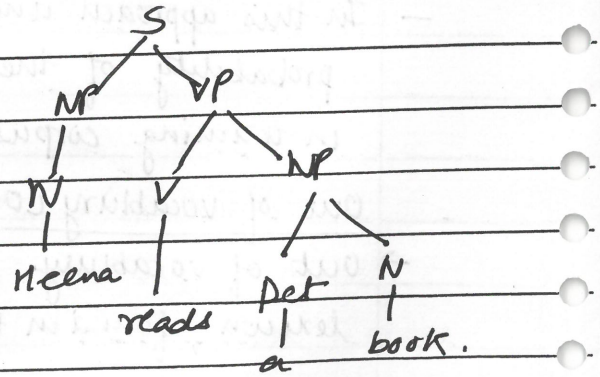
The rule $A \rightarrow \alpha$ says that the constituent A can be written as α, this is also called as phrase structure rule.

for example: $S \rightarrow NPVP$ states that S consist of NP followed by VP ie a sentence consist of noun phrase followed by a verb phrase.

- A language is usually defined by the concept of derivation. The basic operation is of rewriting a symbol appearing on the left hand side of production by its right hand side.

For example consider a sentence:— Heena reads a book.

$S \rightarrow NPVP$
$NP \rightarrow N$
$NP \rightarrow det\ N$
$VP \rightarrow V\ NP$
$VP \rightarrow V$
$N \rightarrow$ Heena | She
$V \rightarrow$ reads | sings | sleeps



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

Hidden Markov Model (HMM) [Imp (MU, 2023), (MU 2024 (May)).]

- Hidden markov models (HMM's) are a type of statistical modelling which are applied in different fields such as medicine, computer science and data science.
- For example let us consider the case of (weather prediction), weather prediction is all about trying to guess what weather will be like tomorrow based on the history of observations of weather. Lets assume a simplified model of weather prediction which will collect statistics on what the weather was like today based on what the weather was like yesterday, the day before and so forth. The computation of probabilities will be as follows: $P(w_n | w_{n-1}, w_{n-2} \cdots w_1)$
- The simplifying markov assumption is as follows:—

$$P(w_n | w_{n-1}, w_{n-2} \cdots w_1) \approx P(w_n | w_{n-1})$$

- first order markov assumption assumes that the probability of an observation at time 'n' depends on the observation at time n-1
- A second order markov assumption would have an observation at time 'n' depend on n-1 and n-2 . . .

Joint probability using Markov assumption:—

$$P(w_1 \cdots w_n) \approx \prod_{i=1}^{n} P(w_i | w_{i-1})$$

- The hidden markov model (HMM) is a type of markov model where there are two states hidden.
- It is a hidden variable model which can give the observation of another hidden state using markov assumption.
- Markov assumption is the assumption that a hidden variable is dependent only on the previous hidden state.
- The markov model is made up of two components, the state transition and hidden random variables that are conditioned on each other.

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 hidden markov model consist of five important components:—

a) Initial probability distribution
b) one or more hidden states
c) Transition probability distribution. (A transition matrix is used to show the hidden state to hidden state transition probabilities)
d) A sequence of observations
e) Emission probability: A sequence of observations likelihood also called as emission probabilities. Each observation expresses the probability of an observation generated from the state.

Advantages of HMM:—

1. HMM has a strong statistical foundation with efficient learning algorithms where learning can take directly from the raw sequence data.
2. It allows consistent treatment of insertion and deletion penalties
3. They are the most flexible generalization of sequence profiles. It can also perform a wide variety of operations including multiple alignment, data mining.
4. It is also easy to combine into libraries.

Applications in NLP:

1. Part of speech tagging:

. Goal: 1) Assign a part of speech to each word in a sentence. HMM's are extensively used for part of speech (POS) tagging in Natural language processing (NLP). The task of POS tagging involves assigning each word in a sentence its corresponding part of speech such as noun, verb, adjective 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

Components of HMM in POS Tagging:—

States: Represent the POS tags (eg NN for noun, VB for verb, JJ for adjective).

Observations: The words in a sentence.

Transition probabilities: The probability of transitioning from one POS tag to another $P(T_i/T_{i-1})$ where T_i is the POS tag at position i.

Emission probabilities: The probability of word being generated by a particular POS tag $P(W_i/T_i)$ where W_i is the word at position i.

Initial probabilities: The probability of a POS tag being the start of sentence. $P(T_i)$.

- Issues in HMM model:— (Disadvantages)
- Hidden markov model (Tagger) it is dependent on every state and its corresponding observed object.
- The sequence labelling in addition to having relationship with individual words, also relates to such aspects as observed sequence length, word context and others.

Optimising HMM with viterbi algorithm:—

The viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states. It is called as Viterbi path. It results in a sequence of observed events.

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

Need of viterbi decoding algorithm:—
It is based on dynamic programming.
Dynamic programming breaks down the problem into sub-problems
It saves the result for future purposes therefore no need to compute the result again.
HMM gives us the probabilities but what we need is the actual sequence of tags. we need an algorithm that can give us tag sequence with highest probabilities. Viterbi needs two tables
The first table is used to keep track of maximum sequence probability and second table is used to keep track of the actual path that led to the probability.

Steps of viterbi algorithm:—

1. Initialization ① Initialize the probabilities for first observation.
② For each state calculate the probability of starting in that state and emitting in that state.

$$V_1(s) = \pi(s) \cdot P(o_1|s)$$

2. Recursion ③ For each subsequent observation calculate the probability of each state being the current state by considering all the previous states. Select the state with the highest probability.

$$V_t(s) = \max [V_{t-1}(s') \cdot P(s|s')] \cdot P(o_t|s)]$$

3. Termination: Identify the final state with the highest probability

$$P = \max V_T(s).$$

4. Backtracking: . Back track through the states to find the most probable sequence of states.
. Store the best previous state for each state at each time during recursion step.