Natural Language Processing Exam Previous Year Question Solve (Fattah Vai) - KUET CSE

Natural Language Processing (Khulna University of Engineering and Technology)

SECTION B

(Answer ANY THREE questions from this section in Script B)

5. a) Define Natural Language Processing (NLP). What are the major areas of research and development of NLP? (10)

b) What does *n*-gram mean? Drive the equation of calculating the probability for *n*-grams model. (10)

c) Consider the following corpus. (08)

<s> I am Sam </s>
<s> Sam I am </s>
<s> I am Sam </s>
<s> I do not like green eggs and Sam </s>

Using a Bigram Language model with add-one smoothing, what is $P(Sam \mid am)$? Include <s> and </s> in your counts just like any other token.

d) What is absolute discounting? What is its advantages? (07)

6. a) What is closed class and open class of Part-of-Speech (POS)? Explain with example. (08)

b) Discuss about Rule-Based POS tagging. Write the ADVERBIAL-THAT RULE. (12)

c) For Hidden Markov Model (HMM) POS Tagging, using the following formula, find the equation of calculating tag transition probabilities. (08)

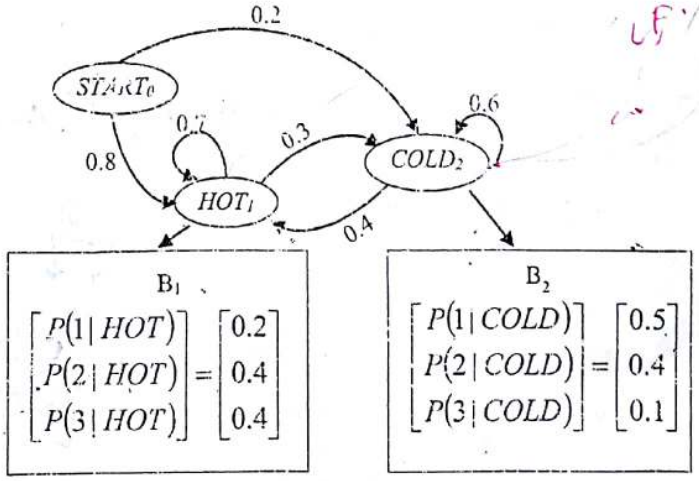
$$\hat{t}_1^n = \arg\max_{t_1^n} P\left(t_1^n \mid w_1^n\right)$$

d) Consider the sentence: "Secretariat/NNP is/BEZ expected/VBN to/TO race/? Tomorrow/NR". The word "race" is often used as VB or NN. Given the probabilities below, find the right POS tag for the word "race". (07)

$P(NN \mid TO) = 0.00047, P(VB \mid TO) = 0.83, P(race \mid NN) = 0.00057, P(race \mid VB) = 0.00012,$
$P(NN \mid VB) = 0.0027, P(NR \mid NN) = 0.0012.$

7. a) HMM characterized by three fundamental problems. Name and discuss about the problems. (09)

b) Given a sequence of ice-cream observations 313 and an HMM $\lambda = (A, B)$ in the following figure, find the best hidden weather sequence $Q(like\ H\ H\ H)$. (12)



c) Define the term odds for logistic regression. Show that the observation should be labeled true if $\sum_{i=0}^{N} w_i f_i > 0$. (09)

d) Write the three-steps of Forward Algorithm. (05)

8. a) Name and discuss about the types of TTS. (06)

b) Speech Synthesis perform text to waveform mapping in two-steps. Name and discuss about the steps. Using Hourglass Metaphor. (12)

c) What is Homograph disambiguation? What are the problems of CMU? How does UNISYN overcome the problems of CMU? (10)

d) Define text normalization. Why does text normalization important for Speech Synthesis? (07)

N grams

1. Why would you want to predict upcoming words, or assign probabilities to sentences?
2. Define LM, N-gram, bigram, MLE, normalize, log probabilities, extrinsic and intrinsic evaluation, training set, test set, held out and development set, ...
3. Drive the N gram probability calculation equation with example of Sam.
4. Given some sentences and count of words and size of V, draw the figures (4.1 and 4.2 in pdf page 40) and example of "I want English food".
5. What does it mean to "fit the test set"?
6. Define Perplexity, drive equation, example
7. Define zeros situation and effect of it.
8. Define closed vocabulary and OOV, Describe the ways to train the probabilities of unknown word model.
9. Define smoothing, drive the equation for laplace smoothing with example and equation of add-k smoothing.

10. Define backoff, interpolation with equation, katz backoff with eqation, absolute discounting and equation, stupid backoff and equation.
11. Define entropy and equation, example of horse, drive the equation of Perplexity's Relation to Entropy (pdf page 56 and 57)
12. Exercises of chapter: 4.1, 4.2, 4.4, 4.7

Important for CT only: 1, 2, 3, 4, 9, 12

1. Reason for predicting upcoming words, or assigning probabilities to sentences:- Probabilities are essential in any of the following tasks

- To identify words in noisy, ambiguous input. eg. Handwriting recognition, speech recognition.
- To perform auto correction, spell correction. eg their ~~are~~ is a blackboard. Here There is mistyped as their.
- for machine translation, eg. translating a bengali sentence to english with appropriate words. eg.

আমি ভাত খেয়েছিলাম

I had eaten rice

- for augmented communication. eg. eye movement to an instruction,

- Identify words
- Auto Correction
- Machine translation
- Augmented Communication

CT-2
NLP

N grams

1 2 3 4 5 6 7 8 9 10 11 12

2/ LM:- Models that assign probabilities to sequences of words are called language models or LMs.

MLE:- An intutive way to estimate probabilities is called maximum likelihood estimation or MLE. We get the MLE estimate for the parameters of an n-gram model by getting counts from a corpus, and normalizing the counts so that they lie between 0 and 1.

Normalizing:- for probabilistic models, normalizing means dividing by some total count so that the resulting probabilities fall legally between 0 and 1.

Log probability:- We always represent and compute language model probabilities in log format that is known as log probabilities.

Extrinsic Evaluation:- The best way to evaluate the performance of a language model is to embed it in an application and measure how much the application improves. Such end-to-end evaluation is called extrinsic evaluation.

Intrinsic Evaluation:- An intrinsic evaluation metric is one that measures the quality of a model independent of any application.

Training Set:- The probabilities of an n-gram model come from the corpus it is trained on is known as training set or training corpus.

Test Set:- The quality of an n-gram model is measured by its performance on some unseen data, which is called test set or test corpus.

Held out set:- Test sets and other datasets that are not in our training sets are called held out corpora because we hold them out from the training data.

Development set:- A fresh test set that is truly unseen is sometimes needed and in such cases, we call the initial test set the development test set or devset.

3. N-Gram probability calculation equation:-

let "Sam is a good" is an incomplete sentence. We want to know the probability of occuring boy after this sentence part

$$P(\text{boy} \mid \text{Sam is a good}) = \frac{C(\text{Sam is a good boy})}{C(\text{Sam is a good})}$$

Now for the full sequence of word "Sam is a good boy" we can use the chain rule -

$$P(x_1, x_2 \ldots x_n) = P(x_1)\, P(x_2|x_1)\, P(x_3|x_1 x_2) \cdots P(x_n|x_1^{n-1})$$

for words $(w_1, w_2, w_3 \cdots w_n)$

$$P(w_1, w_2 \cdots, w_n) = P(w_1)\, P(w_2|w_1)\, P(w_3|w_1^2) \cdots P(w_n|w_1^{n-1})$$

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

$$\Rightarrow P(w_1^n) = \prod_{i=1}^{n} P(w_i|w_1^{i-1}) \quad \text{---} \quad (1)$$

But calculating probability sequences are ~~tiresome~~, rather we can consider only previous one or two words. eg. $P(\text{boy}|\text{good})$

That implies,

$$P(w_n | w_1^{n-1}) \approx P(w_n | w_{n-1}) \text{ for bigram} \quad \ldots (2)$$

This is called markov assumption

for N-gram model,

$$P(w_n | w_1^{n-1}) \approx P(w_n | w_{n-N+1}^{n-1})$$

Substituting (2) in (1),

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

Q.

I	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

$V = 1446$

$\frac{5}{2533}$ ...

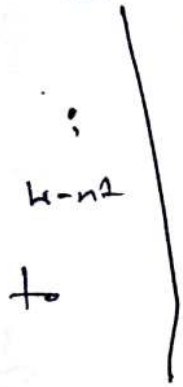
	I	want	to
I	5	827	0
want	2	0	608
to	2	0	4

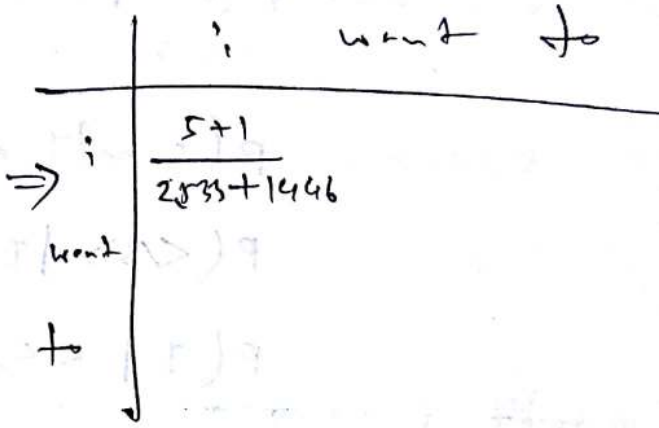
$\Rightarrow$

	I	want	to
I	0.002	0.33	0
want	.0022	0	.66
to	.00083	0	.0017

$P(i \mid <s>) = 0.25$ $\quad P(</s> \mid \text{food}) =$

Smoothing $\quad P^{*}_{\text{Laplace}}(w_n \mid w_{n-1}) = \dfrac{C(w_{n-1} w_n) + 1}{C(w_{n-1}) + V}$





5. What does it mean to "fit the test set"?

Ans:- "fit the test set" means :- whichever model assigns a higher probability to the test set - meaning it more accurately predicts the test set - is a better model. Given two probabilistic models, the better model is the one that has a tighter fit to the test data or that better predicts the details of the test data, and hence will assign a higher probability to the test data.

6. Define perplexity, derive equation, example.

Ans:- Perplexity :- ~~Inv pro~~ The perplexity (PP) of a language model on a test set is the inverse probability of the test set, normalized by the number of words.

for a test set $W = w_1 w_2 \cdots w_N$:

$$PP(W) = P(w_1 w_2 \cdots w_N)^{-1/N}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \cdots w_N)}}$$

We can use the chain rule to expand the probability of W:

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1, \cdots, w_{i-1})}}$$

Thus, if we are computing the perplexity of W with a bigram language model, we get :

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_{i-1})}}$$

The higher the 'gram' the ~~lesser~~ lesser the perplexity ~~that~~ which is a better. ~~approach~~

for example, considering the task of recognizing the digits (zero, one, ... nine) in english each having probability $P = \frac{1}{10}$, imagine a string of digits of length N.

$$PP(W) = P(W_1 W_2 \cdots W_N)^{-\frac{1}{N}}$$

$$= \left(\left(\frac{1}{10}\right)^N\right)^{-\frac{1}{N}}$$

$$= 10$$

7. Define zeros situation and effects of it.

Ans:- Zeros are situations where things that don't even occur in the training set but do occur in the test set.

Effects of zeros situation are the following:-

- First, their presence means we are underestimating the probability of all sorts of words that might occur, which will hurt the performance of any application we want to run on this data.
- Second, if the probability of any word in the test set is zero, the entire probability of the test set is 0. By definition, perplexity is based on the inverse probability of the test set. Thus if some words have zero probability, we can't compute perplexity at all, since we can't divide by 0!.

8. Define closed vocabulary and OOV. Describe the ways to train the probabilities of unknown word model.

Ans:- Closed vocabulary :- A closed vocabulary system is where the test set can only contain words from a lexicon, and there will be no unknown words.

OOV:- In some cases we have to deal with words we haven't seen before, which we'll call unknown words or out of vocabulary (OOV) words.

There are two common ways to train the probabilities of the unknown word model <UNK>

- The first one is to turn the problem back into a closed vocabulary one by choosing a fixed vocabulary in advance:
  1. Choose a vocabulary that is fixed in advance.
  2. Convert in the training set any word that is not in this set to the unknown word token <UNK> in a text normalization step.
  3. Estimate the probabilities for <UNK> from its counts just like any other regular word in the training set.

- The second alternative, in situations where we don't have a prior vocabulary in advance is to create such a vocabulary implicitly, replacing words in the training data by <UNK> based on their frequency.

9. Smoothing :- To keep a language model from assigning zero probability to unseen events, we'll have to shave off a bit of probability mass from some more frequent events and give it to the events we've never seen. This modification is called smoothing or discounting.

Derivation of Laplace Smoothing equation :-

$$P(w_i) = \frac{c_i}{N}$$

Here, N = total No. of tokens

c_i = frequency of the word

Add one to each count for laplace smoothing. For adjustment we consider extra V observations in denominator.

$$P_{Laplace}(w_i) = \frac{c_i + 1}{N + V}$$

To make it easier, we define an adjusted count c^* normalizing N, where normalization factor is $\frac{N}{N+V}$

$$c_i^* = (c_i + 1)\frac{N}{N+V}$$

Now for n-gram

$$P(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

$$P^*_{laplace}(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + 1}{\sum_w (C(w_{n-1} w) + 1)}$$

$$= \frac{C(w_{n-1} w_n) + 1}{C(w_{n-1}) + V}$$

Count can be replaced by ~~$C^*(w_{n-1} w_n)$~~

$$C^*(w_{n-1} w_n) = \frac{[C(w_{n-1} w_n) + 1] \times C(w_{n-1})}{C(w_{n-1}) + V}$$

Example for ~~using~~ laplace smoothing count

	I	want	to	eat
I	5	827	0	9
want	2	0	608	1
to	2	0	4	686
eat	0	0	2	0

	I	want	to	eat
I	6	828	1	10
want	3	1	609	2
to	3	1	5	687
eat	1	1	3	1

Add k smoothing

$$P^{*}_{\text{Add-k}}(w_n \mid w_{n-1}) = \frac{c(w_{n-1} w_n) + k}{c(w_{n-1}) + kV}$$

Exercise

3.1

General Case.

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{c(w_{n-N+1}^{n-1}\, w_n)}{c(w_{n-N+1}^{n-1})}$$

For trigram,

$$P(am \mid \langle s\rangle I) = \frac{c(\langle s\rangle I\ am)}{c(\langle s\rangle I)} = \frac{1}{2}$$

$$P(sam \mid I\ am) = \frac{1}{2}$$

$$P(\langle /s\rangle \mid I\ am) = \frac{1}{2}$$

$$P(I \mid \langle s\rangle\ sam) = \frac{1}{1} = 1$$

3.2

$$P(I\ want\ chinese\ food)$$

$$= P(want \mid I)\ P(chinese \mid want)\ P(food \mid chinese)$$

$$=$$

3.4 $P(\text{sam}|\text{am}) = \frac{c(\text{am sam})}{c(\text{am})}$

$c(\text{am sam}) = 2$

$c(\text{am}) = 3$

$V = 11$

Applying add 1 smoothing

$$P(\text{sam}|\text{am}) = \frac{c(\text{am sam}) + 1}{c(\text{am}) + V}$$

$$= \frac{3}{14} = 0.21428$$

3.7

$$P(\text{Sam}|\text{am}) = \lambda_1 P(\text{sam}|\text{am}) + \lambda_2 P(\text{sam})$$

$$= \lambda_1 * \frac{c(\text{am sam})}{c(\text{am})} + \lambda_2 * \frac{c(\text{Sam})}{N}$$

$$= \frac{1}{2} \times \frac{2}{3} + \frac{1}{2} \times \frac{4}{25}$$

$$= 0.4133$$

(d) Absolute discounting: when we shave off a portion (called d) from non-zero occurances and give that to zero or lower occurrance, is called ~

$$P_{Ab}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{\sum C(w_{i-1}\, v)} + \lambda(w_{i-1})\, P(w_i)$$

Advantages

- Increases the smaller counts
- Makes more accurate result
- More reliable result.

6(a)

Clossed class: A class where it has a fixed membership. For example: Preposition, Conjunction they have fixed number of members. There is very little chance to add new words in them.

Open class: A class where there is a chance of inserting new member. For example: In English, there are 4 classess where new words can be included.

i) Noun: Any naming thing....

ii) Adjective: Talks about quality

iii) Verb: Action

iv) Adverb: modifies

POS tagging

2018
6.(a)

- **Closed class of Part of Speech :** Closed class of POS are those having relatively fixed membership.

Example –

- Prepositions, because new prepositions are rarely coined.
- function words : Grammatical words like of, and or you, which tend to be very short, occur frequently, and play an important role in grammar.

- **Open class of Part of speech:** Open class of POS are those not having fixed membership and continually coined or borrowed from other languages.

Example –

- four major open classes occuring in the languages of the world : nouns, verbs, adjectives and adverbs.

Q. What is POS tagging? Write short notes on POS importance/significance.

Ans. POS tagging:- Pos tagging is the process of marking up a word in a corpus to a corresponding part of a speech tag, based on its context and definition.

Importance of POS tagging:-

- The significance of the POS for language processing is that it gives a significant amount of information about the words and its neighbors.
- POS can be used in stemming for Information Retrival (IR), since
  - Knowing a word's POS can help tell us which morphological affixes it can take
  - They can help an IR application by helping select out nouns or other important words from a document.

Importance of POS tagging:-

- Speech reeognition
- NL Parsing
- Information Retrieval.

2018
6(b)

Rule-based POS tagging:-

- Involve a large database of hand-written disambiguation rule specifying, for example, that an ambigious word is a noun rather than a verb if it follows a determiner.
- In Rule based pos tagging the earliest algorithms for automatically assigning POS were based on a two stage architecture
  - first, use a dictionary to assign each word a list of potential POS.
  - Second, use large lists of hand written disambiguation rules to winnow down this list to a single POS for each word.

The ADVERBSIAL-THAT Rule:-

Given input: "that"

```
if
    (+1  A/ADV/QUANT);  // if next word is adj, adverb, or quantifier
    (+2  SENT-LIM);     /* and following which is a sentence boundary */
    (NOT-1 SVOC/A);     /* and the previous word is not a verb like
                           'consider' which allows adj as object complement */
then eliminate non-ADV tags
else eliminate ADV tags
```

Example:-

I can't wait that long. - "that" used as adverb.

I consider that odd. - "that" used as determiner.

* Stochastic Taggers:-

Resolve tagging ambiguities by using a training corpus to count the probability of a given word having a given tag in a given context.

Hidden Markov Model

Markov Chain: A weighted automaton in which the input sequence uniquely determines which states the automaton will go through.

A markov chain is specified by the following components:

- $Q = q_1 q_2 \cdots q_N$ — a set of N states
- $A = a_{01} a_{02} \cdots a_{n1} \cdots a_{nm}$ — a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, such that $\sum_{j=1}^{n} a_{ij} = 1 \ \forall i$
- q_0, q_F — a special start state and end state which are not associated with observations.

HMM POS Tagging

NLP CT # 3 (Fatah Vai)

1. for HMM POS Tagging, using the following formula, find the equation of calculating tag transition probabilities. (8)

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

2. Consider the sentence: "Secretariat/NNP is/BEZ expected/VBN to/TO race/? Tomorrow/NR? The word "race" is often used as VB or NN. Given the probabilities below, find the right POS tag for the word "race". (7)

$P(NN|TO) = 0.00047$, $P(VB|TO) = .83$

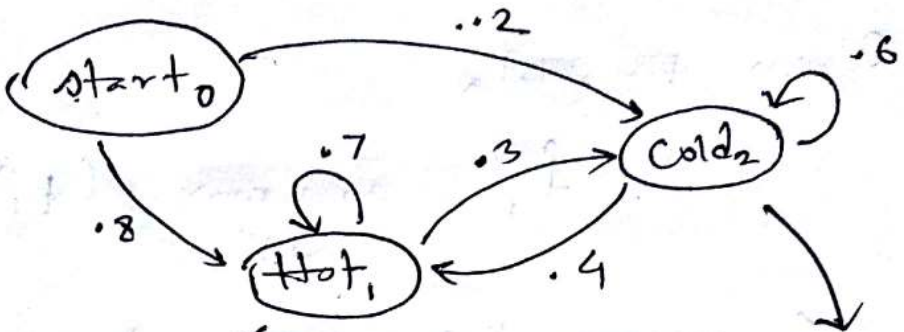
$P(race|NN) = 0.00057$, $P(race|VB) = 0.00012$

$P(NN|VB) = 0.0027$, $P(NR|NN) = 0.0012$

3. HMM characterized by three fundamental problems. Name and Discuss about the problems. (2)

4. Given a sequence of ice-cream observations 3 1 3 and an HMM $\lambda = (A, B)$ in the following figure. find the best hidden weather sequence (12)
Q (like H H H)

↓
Viterbi



B_1

$$\begin{bmatrix} P(1|Hot) \\ P(2|Hot) \\ P(3|Hot) \end{bmatrix} = \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix}$$

$$\begin{bmatrix} P(1|Cold) \\ P(2|Cold) \\ P(3|Cold) \end{bmatrix} = \begin{bmatrix} .5 \\ .4 \\ .1 \end{bmatrix}$$

✓5. Define the term odds for logistic regression. Show that the observation should be labeled true if $\sum_{i=0}^{N} w_i f_i > 0$. (5)

6. Write the three steps of forward algorithm. (5)

1. Ans.

Given Formula

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

^ means "our estimate of the correct tag sequence"

We know,

$$P(x|y) = \frac{P(y|x)\,P(x)}{P(y)}$$

$$\therefore \hat{t}_1^n = \arg\max_{t_1^n} \frac{P(w_1^n|t_1^n)\,P(t_1^n)}{P(w_1^n)}$$

But $P(w_1^n)$ doesn't change for each tag sequence.

$$\therefore \hat{t}_1^n = \arg\max_{t_1^n} \underbrace{P(w_1^n|t_1^n)}_{\text{likelihood}}\,\underbrace{P(t_1^n)}_{\text{prior probability}}$$

Assumption 1: The probability of a word appearing is dependent only on its own part of speech tag.

$$P(w_1^n|t_1^n) \simeq \prod_{i=1}^{n} P(w_i|t_i)$$

Assumption 2:- The probability of a tag appearing is dependent only on the previous tag:

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$$

$$\hat{t}_1^n = \underset{t_1^n}{\arg\max}\ P(t_1^n | w_1^n) \approx \underset{t_1^n}{\arg\max}\ P(w_i|t_i) P(t_i|t_{i-1})$$

The tag transition probabilities:

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

The word likelyhood probabilities:

$$P(w_i|t_i) = \frac{c(t_i, w_i)}{c(t_i)}$$

2. Ans.

to - verb noun
to - noun noun

for verb, $P(VB|TO)\,P(NR|VB)\,P(race|VB) = .00000027$

for noun, $P(NN|TO)\,P(NR|NN)\,P(race|NN) = .00000000032$

$0.00000027 > .00000000032$

∴ The right POS tag for the word "race" is "VB".

5. Ans:

Odds:- In logistic regression, odds are defined as the ratio of the probability of success and the probability of failure.

$$\frac{P(y=true|x)}{1-P(y=true|x)} = w.f \quad (\text{dot product of weight vector and feature vector})$$

According to logit function.

$$\ln\left(\frac{P(y=true|x)}{1-(y=true|x)}\right) = w.f$$

$$\Rightarrow \frac{P(y=true|x)}{1-P(y=true|x)} = e^{w.f}$$

$$\Rightarrow P(y=true|x) = \left(1-P(y=true|x)\right) e^{w.f}$$

$$\Rightarrow P(y=true|x) = e^{w.f} - P(y=true|x)\, e^{w.f}$$

$$\Rightarrow P(y=true|x)\left(1+e^{w.f}\right) = e^{w.f}$$

$$\therefore P(y=true|x) = \frac{e^{w.f}}{1+e^{w.f}}$$

$$\therefore p(y=false|x) = 1 - \frac{e^{w \cdot f}}{1+e^{w \cdot f}}$$

$$= \frac{1+e^{wf} - e^{wf}}{1+e^{wf}}$$

$$= \frac{1}{1+e^{wf}}$$

Our observation should be labeled 'true' if

$$p(y=true|x) > p(y=false|x)$$

$$\Rightarrow \frac{p(y=true|x)}{p(y=false|x)} > 1$$

$$\Rightarrow \frac{e^{wf}/1+e^{wf}}{1/1+e^{wf}} > 1$$

$$\Rightarrow e^{wf} > 1$$

$$\therefore wf > 0$$

$$\Rightarrow \sum_{i=0}^{N} w_i f_i > 0$$

[Proved]

3. Ans:

HMM characterized by three fundamental problems, that are

- Problem 1 - Computing Likelihood
- Problem 2 - Decoding
- Problem 3 - Learning

Computing Likelihood :- Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $\phi(O|\lambda)$.

The most common algorithm for computing Likelihood in HMM is "The Forward Algorithm".

Decoding :- Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.

The most common decoding algorithm for HMMs is "the viterbi Algorithm."

Learning :- Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

The most common learning algorithm for HMM is "The forward-Backward Algorithm".

6. Ans:

Three steps of Forward Algorithm are:-

1. Initialization

$$\alpha_1(j) = a_{0j} b_j(o_1) \qquad 1 \le j \le N$$

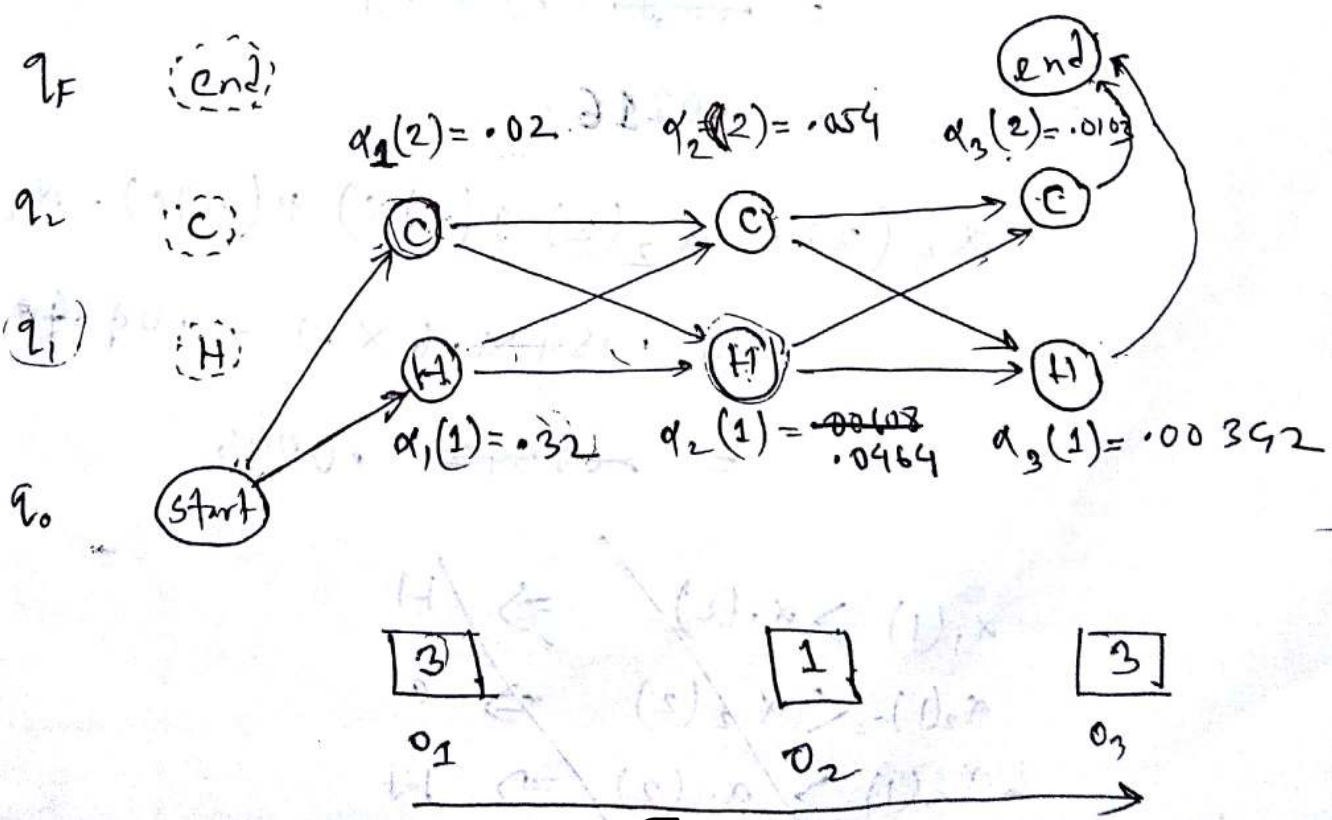
2. Recursion (since state 0 and F are not emitting)

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

3. Termination

$$P(O|\lambda) = \alpha_T(q_F) = \sum_{i=1}^{N} \alpha_T(i)\, a_{iF}$$

4. Ans: Using forward Algorithm:-



$\alpha_1(1) = P(H|start) * P(3|H) = .8 \times .4 = .32$

$\alpha_1(2) = P(C|start) * P(3|C) = .2 \times .1 = .02$

$\alpha_2(1) = \alpha_1(1) * P(H|H) * P(1|H) + \alpha_1(2) * P(H|C) * P(1|H)$

$= .32 \times .7 \times .2 + .02 \times .4 \times .2$

$= .0464$

$\alpha_2(2) = \alpha_1(2) P(C|C) P(1|C) + \alpha_1(1) P(C|H) P(1|C)$

$= .02 \times .6 \times .5 + .32 \times .3 \times .5$

$= .054$

$\alpha_3(1) = \alpha_2(1) P(H|H) P(3|H) + \alpha_2(2) P(H|C) P(3|H)$

$= .0464 \times .7 \times .4 + .054 \times .4 \times .4$

$= .0216$

$\alpha_3(2) = \alpha_2(2) P(C|C) P(3|C) + \alpha_2(1) P(C|H) P(3|C)$

$= .054 \times .6 \times .1 + .0464 \times .3 \times .1$

$= .0046$

$\alpha_1(1) > \alpha_1(2) \Rightarrow H$

$\alpha_2(1) < \alpha_2(2) \Rightarrow C$

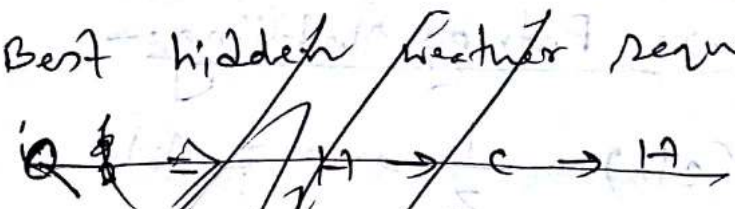
$\alpha_3(1) > \alpha_3(2) \Rightarrow H$

...proper parse tree.

...question answering? Explain.

For maximum likelihood forward

∴ Best hidden weather sequence is



Q = H C H

~~P(313 H C H)~~ $P(O|H) = \alpha_3(1) \times P(\text{end}|H) + \alpha_3(2) \times P(\text{end}|H)$

=

Q2 Viterbi

for sequence

$V_1(1) = .8 \times .4 = .32$

$V_1(2) = .2 \times .1 = .02$

$V_2(1) = \max(.32 \times .7 \times .2, .02 \times .4 \times .2)$

$= .0448$

$V_2(2) = \max(.32 \times .3 \times .5, .02 \times .6 \times .5)$

$= .048$

$V_3(1) = \max(.0448 \times .7 \times .4, .048 \times .4 \times .4)$

$= .0125$

$V_3(2) = \max(.0448 \times .3 \times .1, .048 \times .6 \times .1)$

$= .003$

$V_1(1) > V_1(2) \Rightarrow H$

$V_1(1) < V_2(2) \Rightarrow C$

$V_3(1) > V_3(3) \Rightarrow H$

∴ Best hidden weather sequence Q = H C H.

* Maximum Entropy Modelling :-

$$P(c|x) = \frac{1}{Z} \exp \sum_i w_i f_i$$

Z = normalizing factor

$$P(c|x) = \frac{\exp\left(\sum_{i=0}^{N} w_i f_i\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)}$$

Ex

Given, six features $f_1, f_2, f_3, f_4, f_5, f_6$

		f_1	f_2	f_3	f_4	f_5	f_6
VB	f	0	1	0	1	1	0
VB	w		.8		.01	.1	
NN	f	1	0	0	0	0	1
NN	w	.8					-1.3

$$P(NN|x) = \frac{e^{.8} e^{-1.3}}{e^{.8} e^{-1.3} + e^{.8} e^{.01} e^{.1}} = .2$$

$$P(VB|x) = \frac{e^{.8} e^{.01} e^{.1}}{e^{.8} e^{-1.3} + e^{.8} e^{.01} e^{.1}} = .8$$

$\therefore P(VB|x) > P(NN|x)$, so the right POS tag for the word "race" is "VB".

Linear Regression

- regression when the output is real-valued.
- classification when the output is one of a discrete set of classes.

$$\# \quad y = mx + c$$

$$m = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$= \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x\right)^2}$$

$$= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$c = \frac{\sum y_i - m\sum x}{n} = \bar{y} - m\bar{x}$$

Example page-46

$n = 6$

x	y	xy	x^2
4	0	0	16
3	1000	3000	9
2	1500	3000	4
2	6000	12000	4
1	14000	14000	1
0	18000	0	0
$\sum x = 12$	$\sum y = 40500$	$\sum xy = 32000$	$\sum x^2 = 34$

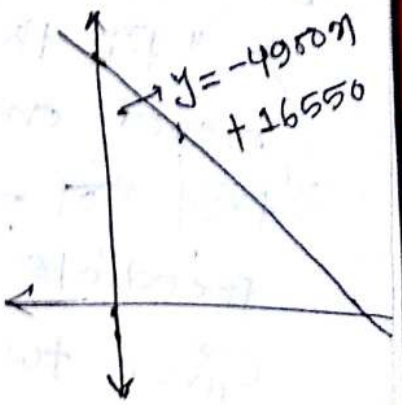
$$m = \frac{6 \times 32000 - 12 \times 40500}{6 \times 34 - (12)^2}$$

$$= -4900$$

$$c = \frac{40500 - (-4900) \times 12}{6}$$

$$= 16550$$

$$\therefore \boxed{y = -4900x + 16550}$$



Speech Synthesis

2018
7.(a)

Types of TTS :-

There are three types of TTS (Text to Speech)

- Formant
- Articulatory
- Concatenative

- Articulatory Synthesis : Model movements of articulators and acoustics of vocal tract.
- Formant Synthesis : start with acoustics, create rules/filters to create each formant.
- Concatenative synthesis : Use databases of stored speech to assemble new utterances.

A full system needs to go all the way from random text to sound.

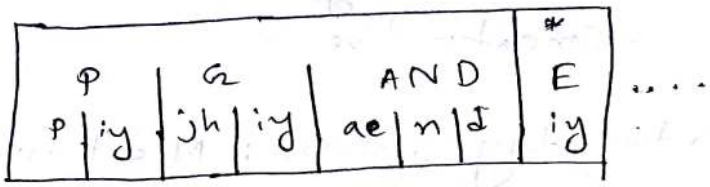
* Concatenative Synthesis :-

- Concatenative synthesis is a technique for synthesizing sounds by concatenating short samples of recorded sound (called units).
- The duration of the units is not strictly defined and may vary according to the implementation, roughly in the range of 10 milliseconds up to 1 second.
- All current commercial systems are concatenative synthesis.

2018
8.(b)

Speech synthesis perform text to wave form mapping in two following steps :-

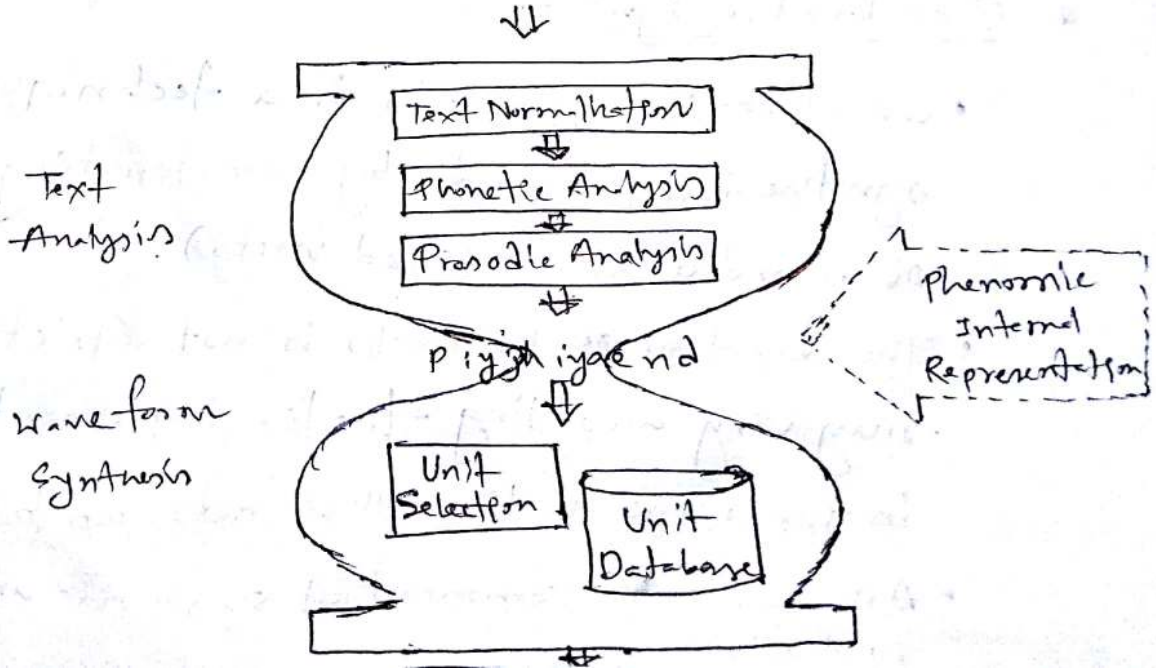
- Step 1 (text analysis) : convert the input text into a phonemic internal representation.



- Step 2 (wave form synthesis) : convert this internal representation into a waveform

According to the hourglass metaphor

PG&E will file schedules on April 20



2018
8 (c)

Homograph Disambiguation :- Homographs are words that are spelled the same but have different meanings. The process of making it dhambiguous is known as homograph dhambiguation. eg.

Do you live near a zoo with live animals.
/l ih v/ /l ay v/

It is not a huge problem, but still important to make it dhambiguous.

Problems of CMU :-

- Har errors
- Only American pronounciation
- No syllable boundaries
- Doesn't tell us which pronounciation to use for which homophones
  - no POS tags
- Doesn't distinguish case
  - The word US has 2 pronounciation
    [AH1 S] and [Y UW1 EH1 S]

UNISYN solve problems of CMU in the following way:-

- Has syllabification, stress, some morphological boundaries.
- Pronunciation can be read off in
  - General American
  - RP British
  - Australia
  - Etc

2018
8.(d)

Text normalization :- Text normalization is the process of transforming text into a single canonical form that it might not have had before

paragraph → sentence → token.

Importance of text normalization for Speech Synthesis :-

- Identify tokens in text
- Chunked tokens into reasonably sized sections
- Map tokens to words
- Identify types of words.

2017
5.(a)

NLP:- NLP is a field of computer Science, artificial intelligence, and lingvistics concerned with the interactions between computers and human languages.

Major areas of R&D in NLP:-

- Text procesing
  - Procesing of raw text
- Morphological Analysis - Study of word formation
  - Washing = Wash + ing
  - Running = Run + ing
- POS tagging

```
>> text = word_tokenize("today is a beautiful day")
>> nltk.pos_tag(text)
```

Output: [('today', 'NN'), ('is', 'PRP'), ('a', 'DT'), ('beautiful', 'JJ'), ('day', 'NN')]

- Machine translation
  - Translating content in one natural language to another natural language.

Ex: আমি ভাত খাই - I eat rice.

- Parsing
  - Identifying sense structure
    S → NP + VP
- Text to speech
  - Converting electronic text to digital speech
- Automatic speech Recognition
  - Automatic transcription of spoken content to electronic text.
- Speech to speech translation
  - Translating spoken content from one language to another in realtime or offline.

Industrial Application of NLP:-

- Search engines
- fraud detection
- Information extraction
- Collaborative filtering
- Computational advertising
- Advanced text Editors
- Sentiment analysis
- Opinion mining

Backoff: We use the trigram if the evidence is sufficient. otherwise we use the bigram, otherwise the unigram. so if the higher order n-gram have zero-evidence then we will "back-off" to the lower-order n-gram

Interpolation: Interpolation is the method where we mix the probability estimates from all the n-gram estimators, weighing and combining the trigram, bigram. and unigram counts.

A simple linear interpolation

$$\hat{P}(w_n | w_{n-2} w_{n-1}) = \lambda_1 P(w_n | w_{n-2} w_{n-1}) + \lambda_2 P(w_n | w_{n-1}) + \lambda_3 P(w_n)$$

Such that the λ's sum to 1:

$$\sum_i \lambda_i = 1.$$

Katz backoff: In Katz backoff we only use a discount probability p^* if we've seen this n-gram before. otherwise, we recursively back off to the Katz probability for the shorter-history (N-1)-gram. the prob. for a backoff n-gram P_{BO} is

$$P_{BO}(w_n \mid w_{n-N+1}^{n-1}) = \begin{cases} p^*(w_n \mid w_{n-N+1}^{n-1}) & \text{if } C(w_{n-N+1}^{n}) > 0 \\ \alpha(w_{n-N+1}^{n-1}) P_{BO}(w_n \mid w_{n-N+2}^{n-1}) & \text{otherwise,} \end{cases}$$

absolute discounting: Absolute discounting formalizes ~~this~~ by subtracting a fixed discount d from each count. The intuition is that since we have good estimate already for the very high counts, a small discount d won't affect them much.

for bigram,

$$P_{AD}(w_i \mid w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}v)} + \lambda(w_{i-1}) P(w_i)$$

Katz backoff: In Katz backoff we rely on a discount probability p^* if we've seen this n-gram before, otherwise, we recursively back off to the Katz probability for the shorter-history (N-1)-gram, the prob. for a backoff n-gram P_{BO} is

$$P_{BO}(w_n \mid w_{n-N+1}^{n-1}) = \begin{cases} p^*(w_n \mid w_{n-N+1}^{n-1}) & \text{if } C(w_{n-N+1}^{n}) > 0 \\ \alpha(w_{n-N+1}^{n-1}) P_{BO}(w_n \mid w_{n-N+2}^{n-1}) & \text{otherwise,} \end{cases}$$

absolute discounting: Absolute discounting formalizes ~~this~~ by subtracting a fixed discount d from each count. The intuition is that since we have good estimates already for the very high counts, a small discount d won't affect them much.

for bigram,

$$P_{AD}(w_i \mid w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}v)} + \lambda(w_{i-1}) P(w_i)$$

Stupid backoff: In this algorithm there is no discounting of the higher-order prob. if a higher-order n-gram has a zero count, we simply backoff to a lower order n-gram.

$$S(w_i \mid w_{i-k+1}^{i-1}) = \begin{cases} \frac{count(w_{i-k+1}^{i})}{count(w_{i-k+1}^{i-1})} & \text{if } count(w_{i-k+1}^{i}) > 0 \\ \lambda S(w_i \mid w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

11 Entropy: Entropy is a measure of information. Given a random variable x ~~xxx~~ ranging over whatever we are predicting and with a particular probability function, call it p(x), the entropy of the rand. var. x is,

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Example of horse: suppose we want to bet on a horse race but it is too far away. so we'd like to send encoded messages using binary representation. so, the horse 1 001, horse 2 010, horse 3 011, and so one with horse 8 encoded as 000, is we spent all day to bet and each horse is coded with 3 bits on avg. we will send 3 bit. suppose the probability of winning of the horses

horse 1	1/2	horse 5	1/64
horse 2	1/4	" 6	"
horse 3	1/8	" 7	"
horse 4	1/16	" 8	"

so. ~~now we will try~~ to

The entropy of the random variable X that ranges over horses gives us a

lower bound on the number of bits and its

$$H(x) = -\sum_{i=1}^{i=8} p(i) \log p(i)$$

$$= -\frac{1}{2}\log 1/2 - \frac{1}{4}\log 1/4 - \frac{1}{8}\log 1/8$$

$$- 4(1/16 \log 1/16) - 4(1/64 \log 1/64)$$

$$= 2 \text{ bits.}$$

so we can create a coding system that averages 2 bits per race,

so we can encode the most likely horse with 0, remaining horses as 10, 110, 1110, 111100, 111101, 111110, 111111

if all winning prob. is same and let $1/8$

then;

$$H(x) = -\sum_{i=1}^{i=8} \frac{1}{8}\log\frac{1}{8}$$

$$= -\log\frac{1}{8} = 3 \text{ bits.}$$

Rel^n with perplexity: The perplexity of a model p on a sequence of words w is now formally defined as the exp of this entropy:

~~perplexity (w) = $2^{H(w)}$~~

~~= p(~~

$$H(w) = -\frac{1}{N} \log P(w_1 w_2 \cdots w_N)$$

$$\therefore \text{perplexity}(w) = 2^{H(w)}$$

$$= P(w_1 w_2 \cdots w_N)^{-1/N}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \cdots w_N)}}$$

$$= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \cdots w_{i-1})}}$$