Name : Dharamraj Bhatt
Reg.No : 19477216

Natural Language Processing (MCA541E)

Mid term exam

Ans 1 a) Ambiguity is a natural characterstic of human conversations and one that is particulary challenging NLU (Natural language understanding) scenarios. By ambiguity. Here ambiguity means a sentence that has multiple alternative interpretation.

it is the one of the are of cognitive science that doesn't have a wfll defined solution. From a technical point of view any sentence in a language with a large enough grammer can have alternative interpretation however most native speaker only recognize the primary interpretation when hearing a phrase. while alternative representation may be more obvious to non-native speaker. If we human are facing difficulty to deal with ambiguity then just think of NLU system.

types of ambiguity :- These are the some of the ambiguity

i) Lexical Ambiguity
ii) syntactic ambiguity
iii) Semantic ambiguity
iv) metonymy

Reg. No: 19472IC

we can handle ambiguity by using a ~~those~~ built in library such as expert.ai. ~~left unclear~~. For tagging and semantic tagging can be handled by expert.ai.

Part of speech tagging :- "language is a ambigious" not only a sentence could written in a different ways and still convay the same ~~thing~~ meaning, but even lemmas/a concept that supposed to be less ambiguous can carry different meaning.

Semantic ~~tagging~~ :- One word can also have the same grammatical label and have different meaning. this phenomenon is called polysemy. being able to infer the correct meaning for each word is to perform semantic tagging. word that are more common tend to have more meaning that have been added to them in time.

Ans2b) ~~To~~ To keep a language model from assigning a zero probability to the unseen event, we will have to shave off a bit of a probability mass from some more frequent events and give it to the event we have never seen. this modification we call ~~smoth~~ smoothing. the simplest way to do ~~something~~ smoothing is to add one ~~word~~ to all the bigram count, before we normalize them into probabilities. all the count that used to be zero will now have a count of 1, the

count of 1 will be 2 and so on. this algorithm is also known as laplace smoothing.

Any two methods -

1) N-gram : let's compute (P(w/H) - probability of word 'w', given some history 'H'. Suppose H is "its water is so transparent that" and we want to know the probability of next word 'the': P(the/its water is so transparent that). one way to estimate this probability is relative frequency count.

$$P(\text{the}/\text{its water is transparent that}) = \frac{C(\text{its water is so transparent that the})}{C(\text{its water is so transparent that})}$$

The intuition of the n-gram model is that instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

2) Markov model : markov models are the class of probabilistic model that assume we can predict the probability of some future unit without looking too far into the past. we can generalize the bigram (which looks one word into the past) to the trigram (which looks 2 words into the past) and thus to the n-gram (which looks n-1 grams into the past)

Reg.No: 1947216

we can compute the probability of a complete word sequence by substituting -

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

Ans 3c) HMM is a probabilistic ~~model~~ machine learning model, which is used in speech recognition. HMM provides the solution for three problem: evaluation, decoding and learning to find most likelihood classification

before understanding HMM ~~properly~~ properly we have to understand markov chain.

markov chain and HMM (Hidden Markov model) both are extensions of finite automata which is based on the input observation. a weighted finite automata is a simple augmentation of finite automaton in ~~the~~ which each are associated with probability. the markov chain is specified by the following components

$Q = q_1 q_2 \cdots q_n$ - a set of states.

$A = [a_{ij}]_{N \times N}$ - a transition probability matrix

q_0, q_{end} - start and end state

· A markov chain embodies ~~in~~ important assumptions about these probabilities ~~in~~ a first order markov chain.

markov assumption : $P(q_i | q_1 \cdots q_{i-1}) = P(q_i | q_{i-1})$

Reg. No : 1947216

$\pi = \pi_1, \pi_2 \ldots \pi_n$ - an initial probability distribution

each π_i expresses the probability $p(q_i)$ start, all the π probabilities must sum to 1.

HMM: HMM is specified by the following components

$Q = q_1 q_2 \ldots q_n$ - set of states

$A = [A_{ij}]_{N \times N}$ - transition probability matrix

$O = O_1, O_2 \ldots O_N$ - set of observations.

$B = b_i(O_t)$ - set of observations likelihood

q_0, q_{end} - start and end state

$\pi = \pi_1, \pi_2, \pi_3 \ldots \pi_n$ - Initial probability distribution

$Q_A = \{ q_x, q_y \ldots \}$ - set of $Q_A \subset Q$ of legal accepting states.

~~For computing we use the forward algorithm~~

Evaluation: it is meant what is the probability that a particular sequence of symbol is produced by a perticalar model. For evaluation there algorithm mostly used the forward algorithm or the backword algorithm.

Reg. No: 1947216

Decoding :- decoding means determining the most likely sequence of states that produced the same sequence. For this problem we use Viterbi algorithm.

End

Name : Dharamraj Bhatt
Reg. No: 1947216