

Q (1) Word Sense Disambiguation is an important method of NLP. Meaning of a word can be determined by this method. NLP systems are complex, so they face difficulties in identifying the words properly and determining the specific usage of a word in a particular sentence.

Word Sense Disambiguation solves the ambiguity that we face in determining the meaning of the same word used in different scenarios.

The Lesk algorithm is based on the idea that words in a given section of the text will have a similar meaning. The correct meaning of each word context is found by getting the sense which overlaps or matches the most among the given context.

For example, consider the sentence:
"Go straight and take a right turn".

Here we are trying to find the sense of the word "right".

According to Lesk Algo, we take all of its context words and find the intersection of target word meaning with meaning of each context word.

(f) 

Then the max value among these is chosen to be the actual sense of the target word.

We take the whole sentence as its context.

Let the context words are c_1, c_2, \dots, c_n .
 right can have 2 senses — $\begin{matrix} \text{right} \\ (S_1) \text{ direction} \end{matrix} / \begin{matrix} \text{right} \\ (S_2) \text{ correct} \end{matrix}$

then we find intersection of S_1 with the sense of c_i and add to each c_i and repeat the same for S_2 .

$$o_1 = \sum_{i=1}^n [S_1 \cap S(c_i)] \quad o_2 = \sum_{i=1}^n [S_2 \cap S(c_i)]$$

We observe the words like straight, turn.

So, the conclusion that can be drawn is

$$[o_1 > o_2]$$

So, the final sense of right in the sentence is direction.

Sentiment analysis may be done
Senti

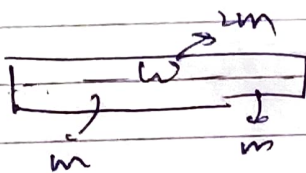
Sentiment analysis using SentimentNet.

Anchor words are words that are definitely $+ve$ or $-ve$.

Ex! - good-bad, joyful-cursed.

$w \rightarrow$ arbitrary word.

- there are some neutral words that are condition dependent. we could use it in $+ve$ or in $-ve$ sentence both.



Mutual information.

$$I(P, w) = \frac{\text{Prob}(P, w)}{\text{Prob}(P) \text{Prob}(w)}$$

1 - $+ve$ anchor word.

↓
It will imply independence

Across a very large no of documents, the prob that P & w occur together divided by their individual probability

$$\frac{\text{Prob}(P, w)}{\text{Prob}(P) \text{Prob}(w)} \rightarrow \text{using both probabilities } P \& w,$$

we will get a sense of whether this word ' w ' co-occure with $+ve$ or $-ve$ words or neutral.

this is how, we can estimate sentiment of diff words.

Q 2)

i). First of all, we have to check the polarity of each sentence. It will ~~sh~~ conclude that whether the given feedback is true or true in sense.

A single review may has more than one sentiment as one & aspect.

ii) After the polarity, we train our classifier for expanded sets using sentimentnet.

(iii) we will decide a particular k value based on k -mean classification.

iv) we train different classifiers of different types and different k -values for the given data. we

✓) we will calculate the association score based on eq $assoc(s_i, s_j) = \begin{cases} f(x) \cdot c & \text{if } (j-i) \leq T \\ 0 & \text{otherwise} \end{cases}$

vi) we will build ~~an~~ undirected graph with vertices (v_1, v_2, \dots, s, t)

Add edge (s, v_i) each with weight $\text{ind}(x_i)$
 $u \quad u \quad (t, v_i) \quad u \quad u \quad \text{ind}(x_i)$
 $u \quad u \quad (v_i, v_f) \quad u \quad u \quad \text{abs.c}(v_i, v_f)$

then we use the k-means clustering algo on polarity. these words can be linked using isopolarity