Computational Linguistics - 2

CL is the application of CSc in Linguistics. Includes a lot of theoretical questions - NLP has a more applied focus

Three eras of CL

1) Rule based approaches
2) Statistical methods
3) Neural models (Deep Learning)

Ambiguity:- Can be at word, phrase or sentence levels

Words are challenging

-> Segmenting into words

-> Domain specific meanings etc

Origin of He/Him : She/Her -> Bantu langs

Manning and Schulze

Zipf's Law:-

Tokens:- Individual occurences of words

Types:- No. of unique instances

a rose is a rose is a rose

Types = a, rose, is

Tokens = 8

Type-token ratio is the measure of lexical diversity.

Hapax Legomena :- Words that appear only once.

Common 100 words usually account for > 50% of tokens.

Zipf's Law = $\underset{\text{freq}}{\nu} \propto \frac{1}{\text{rank}}$

Speaker prefers a smaller vocab of common words for easier comms.
Hearer prefers a larger vocab of rarer words for lucid comms.

Thus the two arrive at a maximally economical compromise

ZL -> $f \propto \frac{1}{r}$ => $f \cdot r = k$ (const)

Mandelbrot in 1956 argued that this was a bad fit for both the low and the high ranks. He instead suggested

$f = P(r+\rho)^{-\beta}$ where P, ρ and β are text based params.

If $\beta = 1$ & $\rho = 0$, Mandelbrot's law becomes Zipf's law.

There is a lot of variation b/w the different types of text, hence parsers trained on one type would usually not work for other types

Supposing we randomly generate text, it will exhibit Zipf's Law.

The probability of word length n = $\underbrace{\left(\frac{26}{27}\right)^n}_{\text{non blank character}} \times \underbrace{\frac{1}{27}}_{\text{followed by a blank}}$

Text Classification:-

Document Classification:- Sort documents into user defined classes

Sentiment Analysis:- Assigning ~~a letter~~ a sentiment to a text. Initially a ternary system:- +ve, -ve & neutral, but now has 9 types.

Authorship Attribution:- Author identification / Plagiarism

Spam Filtering:- Spam vs Ham

Language Identification:- closed-world domain

Assumption:- Single source, monolingual documents of certain length where we know every language n-grams:- continuous n-sized sequences of words or characters

Store a frequency of distribution of trigrams for every given language. Apply the freq dist to a new text and use it to judge the source language

But cross domain performance is much (much!) poorer than ~~intra-~~ in-domain performance

Our goal is to identify n-grams with high ~~and low~~ language association and low domain association

Trigrams are the tradeoff b/w paucity and reliability

Higher n-grams are rare but reliable and vice-versa

Information Gain → or called Entropy (degree of uncertainity)

$$\text{Info}(D) = -\sum_{i=1}^{m} p_i \log_2 (p_i)$$

→ Total no. of classes

Avg information to identify the class label of a tuple d

Dataset → Tuple with class label

→ Non zero probability that any tuple in D belongs to class C_i

Training Samples

Attribute or feature A having v distinct values $\{a_1, \dots, a_n\}$

Unit of entropy is bits

To encode n different sequences is $\lceil \log_2 n \rceil$

But info. is related to probability. $p \propto \frac{1}{\text{info}}$

Entropy in a sense a measure of impurity of data (mixing of classes, imbalance in classes)

High Entropy is better for training data

Supervised Learning :- Training & test data have been labelled with the correct answers

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$

→ no of types times | cardinality?

Attribute $A = \{a_i\} \ \forall \ i \in [1, v]$

↳ values

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

Information Gain tells us how important a given attribute of the feature vector is.