1. INTRODUCTION TO NLP

What is NLP?

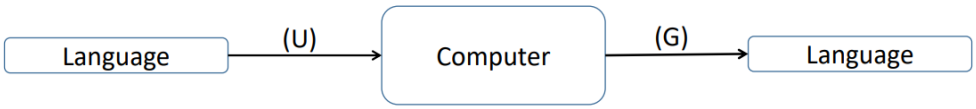
Analysis and design of computational agents that use natural languages to acquire information from other agents, human or machine.

Information given by following sentence:

"For sale: baby shoes, never worn"

- Advertisement
- Reselling an item
- Baby died during birth
- Shoes are not fit to the baby

Basic NLP pipeline



U – Understand, G – Generate

Why is language processing hard?

- It is difficult to understand the information passing with natural languages by computers
  Example: sarcastic way of passing information
- It is difficult to understand the context from collection of words
- Difficult to understand the rules of natural languages
- Ambiguity in language

Why human language ambiguous?

- Ambiguity can be held at many levels, phonological, morphological, syntactic, pragmatic
- Meaning of words differs with respect to the context
- New words are adding over time
- Difficult to catch sarcastic meaning
- Local slangs
- Inherently lot of information are inside a sentence
- If we explicit these information the language will be more complicated
- These information we should get from context and world knowledge
- Ambiguity in language makes actual communication much, much easier, because you don't need to enunciate every single bit of repeated or inferable information.
- Many words have multiple meaning

Advantages and disadvantages of ambiguity in language modelling?

Advantages:

- Less effort
- With small set of vocabulary we can do communication effectively.
- Lot of context available

Disadvantages:

- Difficult to understand and make decisions based on it.
- In computational perspective, representing the context is difficult

Reading Assignment-1: On the Nature of Economy in Language

In this paper author try to compare economy in language with economy in the physics. Nature always choose shortest or easy way to execute the things. For example light will choose a path to travel from one point to another which takes shortest time rather than shortest distance. Principle of least action implies that any action in the nature includes the travel of light and motion of bodies, which makes every action least. For making any action least, the difference between kinetic energy and potential energy associated with the any action should be least.

Economy of language can be explained as economy of representation and economy of derivation.

- Economy of representation means that for conveying an idea with a language one should use legitimate objects in the language to present the idea
- Economy of derivation implies that conveying any idea in a particular language, in order to save effort and time few or minimalistic words should be used.

Which means language always use minimal objects for conveying the ideas like principle of least action in physics. Language shares fundamental property of inorganic world which is reflect from economy of derivation of language. Economy of representation is an independent property of language which requires representations to be redundant which makes effort of understating and conveying idea least.

2. TEXT PROCESSING

Regular Expressions

- Formal language for specifying text strings
- For standardizing text so that they can use efficiently
- used for pre-processing, or as features in the classifiers
- Useful for capturing generalization

Ranges:

Pattern	Matches
[A-Z]	An upper case letter
[a-z]	A lowercase letters
[0-9]	A single digit
[A-za-z1-0]	An alpha numeric character
[^A-Z]	Not an upper case letter
^[A-Z]	Start with capital letter
^[^A-Za-z]	Not start with an alphabet

|	OR
*	Zero or more
+	One or more
.	Any character
?	Previous character optional
\.$	End with dot
.$	End with any character

Examples:

- Find instances of the word “the” in the text

  [^a-zA-Z][tT]he[^a-zA-Z]

- Match colour or color

  Colou?r

- Detecting email addresses

  ^[a-z0-9][a-z0-9\._]+@([a-z]+|.)+(com|in|edu)

Substitutions

s/regexp1/pattern/

Example: Substitute colour with color : s/colour/color/

Capture groups

- () to "capture" a pattern into a numbered register (1, 2, 3…)
- Use \1 to refer to the contents of the register
- Parentheses have a double function: grouping terms, and capturing

Examples:

- Put angles around all numbers : s/([0-9]+)/ <\1>/
- /the (.*)er they (.*), the \1er we \2/ matches the faster they ran, the faster we ran

Non-capturing groups

- Add a ?: after ()
- Example:
  /(?:some|a few) (people|cats) like some \1/
  Matches: some cats like some cats

Look ahead assertions

- (?= pattern) is true if pattern matches
- (?! pattern) true if a pattern does not match

Example: beginning of a line, any single word that doesn't start with "Volcano":

/^(?!Volcano)[A-Za-z]+/

ELIZA

Early NLP system that imitated a Rogerian psychotherapist which uses regular expressions

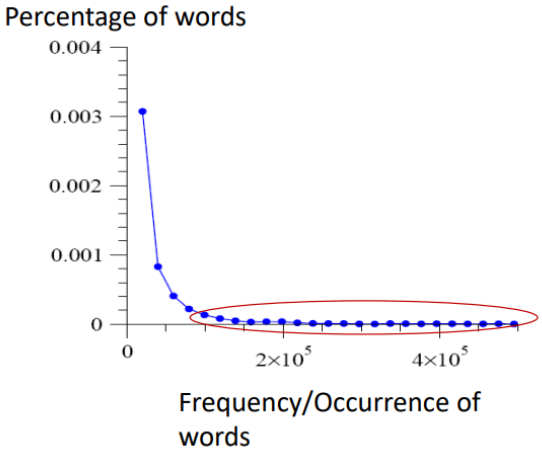
3. WORD DISTRIBUTIONS

Words are not distributed evenly they will follow the 80/20 rule like alphabet, city sizes, wealth, etc. 80% of the words in a corpus goes to 20% of the types and remaining 20% words goes to rest of the 80% types.

Stop words: more frequently occurred words in a corpus

Power-law Distribution

- Many words with a small frequency of occurrence
- A few words with a very large frequency
- High skew (asymmetry)



Mathematically power law distribution can be written as

$$p(x) = Cx^{-\alpha}$$

Where

- $p(x)$: probability of observing an item of size x
- C : normalization constant (probabilities over all x must sum to 1)
- α : scaling exponent, or power law exponent
-

Power law distribution is straight line in log-log scale

$$\ln\big(p(x)\big) = \ln(C) - \alpha\ln(x)$$

Zipf's Law

$$\text{Rank} \times \text{Frequency} \approx \text{Constant}$$

- Constant ≈ 0.1 × Length of collection (in words)
- Zipf's law can be used to identify stop words
- Using zipf's we can find important words (least frequent words)

Linguistic aspects of word distributions

- Most frequent words are more ambiguous as compared to least frequent words.
- The more frequent a word is, the shorter it tends to be
- Observed across almost all languages
- Very few words are used frequently, most of the words occurs a very few times.
- Economic vocabulary words used few times (consequence of least effort)
- In communication speaker use words by reducing his efforts (few words reducing effort) and hearer can correctly interpret meaning with less effort

Ambiguity, frequency of words, principle of least effort relation?

- Less frequent words are less ambiguous and convey meaning easily but effort is more
- Most frequent words are more ambiguous and less effort
- Speaker tends to use more ambiguous words to reduce his efforts

Computational aspects of word distributions

- Word frequencies in a language can be approximated as a distribution
- Some domain specific words treated as important
- More frequent words are ambiguous so word distribution helps to disambiguation.

Why the complex language production processes should conform to a mathematically concise equation?

Unlike programming language natural language don't have a specific usage cases for each words. Machines can understand the programming language from predefined usage case each words. Natural language in other hand, usage of words are uncertain or ambiguous. This uncertainty or ambiguity can represent with the help of probability concept in mathematics. Computing language with probability eliminates the disambiguation involved within the natural language. This is helpful in determining usage of words within a certain context.

Properties of Words

1. Word burstiness
   A word more likely to occur in a document when it is already appeared in the document. Burstiness and semantic contents are positively correlated. More informative words are more bursty. Computational aspects of word bustiness is caching i.e. things repeatedly accessing is likely to access again and again.

Heap's Law

The number of unique words in a text of n words is approximated by.

$$V(n) = Kn^{\beta}$$

In English, K is between 10 and 100, β is between 0.4 and 0.6

What are the linguistic and computational aspects of word burstiness and Heaps' law?

According to word burstiness a word is more likely to occur again in a document if it has already appeared in the document. Heaps law represents the vocabulary growth of a document with a mathematical equation. From the heap law it's clear that as new words visit number of new words added to the document decreases exponentially. Which clearly represents the word bustiness, i.e. as document progress the author will try to use same words again and again.

4. LANGUAGE MODELLING

Why probability for language modeling?

- Probability captures uncertainty
-
-

Language models compute either:

- Probability of a sentence or sequence of words : $p(w) = p(w_1, w_2, \ldots\ldots, w_n)$
- Probability of an upcoming word : $p(w_n | w_1, w_2, \ldots\ldots, w_{n-1})$

$$p(w_1, w_2, \ldots\ldots, w_n) = \prod_i p(w_i | w_1, w_2 \ldots\ldots w_{i-1})$$

Challenges in language modelling:

- No too many possible sentences
- Computationally expensive
- Cases where denominator is zero
- Need a large dataset

Markov's Assumptions:

Approximate calculation of $p(w_1, w_2, \ldots\ldots, w_n)$, that is current state depend only on previous k states

$$p(w_1, w_2, \ldots\ldots, w_n) \cong \prod_i p(w_i | w_{i-k}, w_2 \ldots\ldots w_{i-1})$$

N-gram Models

- Unigram models

  All words are independent

  Which will generate sentence without any meaning

$$p(w_1, w_2, \ldots\ldots, w_n) = \prod_i p(w_i)$$

- Bigram

  One word is conditioned on its previous word

$$p(w_1, w_2, \ldots\ldots, w_n) = \prod_i p(w_i | w_{i-1})$$

Why trigrams, 4-grams, 5-grams is an insufficient model of language?

- Long distance dependencies of language

Estimating N-gram probabilities

Maximum likelihood estimate for bi-gram

$$p(w_i|w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

Maximum log likelihood estimate

- Avoid underflow
- Addition of terms is computationally less expensive than multiplication

Evaluation of language models

Train the model on train data and evaluate the performance on an unseen data (test set) with some useful evaluation metrics.

Extrinsic (in-vivo) evaluation

In this method put each model in a task and get accuracies for each model and compare them for best model. But extrinsic evaluation is time consuming.

Intrinsic evaluation: perplexity

- It is helpful when test data looks more like training data
- The best language model is one that best predicts an unseen test set
- Perplexity is the inverse probability of the test set, normalized by the number of words

$$PP(W) = p(w_1, w_2, \ldots\ldots, w_n)^{-\frac{1}{N}}$$

For bi-gram model,

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

- Minimizing perplexity is the same as maximizing probability
- Lower perplexity = better model

Zero probability problem in N-gram models

- If a bi-gram have zero probability (the bi-gram don't ever occur in the training set but occur in the test set) the probability of entire test set will be zero. So cannot compute perplexity

Smoothing

- Overcomes zero probability problems in language models
- Avoids overfitting of language models – better generalization

1. Add-1 or Laplace smoothing

Pretend each word or N-gram occurred one more time in corpus. That is add 1 to all counts

For bigrams,

$$P_{MLE}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$$

$$P_{Add-1}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + 1}{C(w_{i-1}) + V}$$

Where V = vocabulary or number of types

Revised counts after laplace smoothing,

$$C^*(w_{i-1}, w_i) = \frac{(C(w_{i-1}, w_i) + 1) * C(w_{i-1})}{C(w_{i-1}) + V}$$

What is maximum likelihood estimates?

Estimating parameters of a model by maximizing the likelihood of the training set given the model

- Add one smoothing is not a good estimate and not used for N-gram models. Its generatlly used for text classification and language models where number of zeros is less.

2. Backoff

Use trigram if you have good evidence, otherwise bigram, and otherwise unigram

3. Interploation

- mix unigram, bigram, trigram
- it works better for most of cases

Simple interpolation

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

Lambdas conditional on context:

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

How we will choose λs?

- Found out using held-out or validation data
- Fix N-gram probabilities on training data then search for λs that give largest probability to held-out set

How to deal unknown vocabulary?

- Create a fixed lexicon L of size V
- At text normalization phase, any training word not in L changed to <UNK>
- Train its probabilities like a normal word
- At decoding time If input text have an unknown word, then use UNK probabilities for that.

4. Add-k smoothing and Unigram prior smoothing

$$P_{Add-k}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + k}{C(w_{i-1}) + kV}$$

$$P_{Add-k}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + m(\frac{1}{V})}{C(w_{i-1}) + m}$$

Where m is some weight

$$P_{UnigramPrior}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + m * P(w_i)}{C(w_{i-1}) + m}$$

5.Good turing smoothin

Use the count of things we have seen once to help the estimate the count of things we have never seen.

Good turing calculations:

$$P_{GT}(things\ with\ zero\ frequency) = \frac{N_1}{N}$$

Revised count for other types,

$$C^* = \frac{(c+1)N_{C+1}}{N_c}$$

Where:

N_1 = count of things seen once

N_c = count of things seen c once

For large c, N_c will be zero in most cases, so when count gets unreliable we can replace N_c by power law distribution.

Reading Assignment-2: 'On Chomsky and two cultures of statistical learning'

In this article authors explaining what are the views of Chomsky on statistical models and author contradicts some of his vies with fact and examples.

According to Chomsky:

- Words in a language occurs without any probability or statics. We can predict next word in a sentence by calculating the probabilities of previous word.
- Statistical models are incapable of learning and understating the language, it can't capture the reality.
- Statistical models are incomprehensible; they provide no insight.

Counter points of author:

- Chomsky neglects the factual benefits statistical models his views are totally theory oriented.
- One can gain insight by examing the properties of the model rather inspecting each parameters individually.
- Language is a stochastic phenomenon like phenomenon in science which can be model simply with probabilities.
- Similar to models like gravitational law or ideal gas law probabilistic models of language doesn't model reality completely but does provide good predictions and insights about the phenomenon.

Statistical Models:

Statistical model is a mathematical model trained by input data. Statistical models are often but not always probabilistic. A probabilistic model specifies a probability distribution over possible values of random variables, e.g., P(x, y), rather than a strict deterministic relationship.

Statistical language models proven the success in the following applications

- Search engines
- Speech recognition
- Machine translation
- Question answering

Some of the applications of language models in computational linguistics are as follow

- Word sense disambiguation
- POS tagging
- Coreference resolution
- Parsing

5. ETHICAL AND SOCIAL ASPECTS OF NLP

Example:

In google Books N-gram Viewer

- She is a nurse & He is a nurse (very higher probability for she is a nurse than he is a nurse)
- He is a programmer & She is a programmer (unable to catch she is a programmers)

This language model creating some stereotype.

- Language has to do with people and what they mean not with words and what they mean

- What tool and technologies we are making impact people and society

What is ethics?
- Its good things and right things

Are nuclear weapon good or bad?
- Depend on use.
- Who decide use?
- There are uses like controlled use of nuclear weapons for destructing mountains for construction of roads

What is good and right things?
- It's depends on the each individual thought. For some people it will be good, for some may be bad

Legal! = Ethical

Ethics will change over time with values and belief of people (example: same sex marriage)
Ethics is not a binary it depends on inner guiding and moral aspects of people and society

Some examples:

Chicken Classifier
- Gender classifier
- Roster : go to meat farm
- Hen: egg farm or poultry

Is ethical?
- By food perspective its ethical
- By animal conservation it's not ethical

IQ Classifier:

- Based on photo decide a person is intelligent or not
- It is an hypothetical case

Is ethical?

- IQ not only depends on fascial attributes but also it depends on other aspects like emotion, problem solving skills etc.
- Benefit people whose facial complexion and attributes represented by model. If model is trained with images from social media, the facial attributes of people who are not active in social media won't be represented by the model.
- Researcher, user, reviewer and society all are responsible for such a classifier
-

Identification of sexual orientation from facial features:

- Is it ok to do such research?
  - Academic perspective is ok to understand the effect of facial attributes in sexual orientation (exploration)
  - Chance of misusing since it is publishing to public that to consider

- Who will harm such a system?
  - In some countries it's illegal to be a homosexual and they get punished
  - Its affect employment of a person
  - Chance of discrimination among people
  - Personal attributes (example: sexual orientation, religion, social construct etc.) may change over time
- Bias issues
  - Only specific group of people considered
  - Public photos not implies it is publicized
  - Data collected in a specific time interval
  - Chance self -selection bias
  - Both classes are balanced, doesn’t represents true class distribution

Intent of this research is to expose threats to the privacy and safety of gay men and women

How model affect bias?

- Model is deep learning model with logistic regression classifier (non-interpretable model) for a sensitive problem, which amplifies the bias
- Even though model giving good accuracy we should consider how miss classification harmful to some people
- Accuracy is a one aspect of evaluation (example: annotation time, training time etc.)

For some application cost of miss-classification is huge

Dual use of AI technologies

- Need to aware of real-world impact of our research works
- We should aware of passible use of the research and its consequences
- Consider benefits and harms
- Cost of wrong actions
- How affect people lives

Ethical and social problems associated with AI products like Alexa, Siri, Barbie etc.

- All the conversations are recorded in the cloud which may include personal information of user.
- These data can be further processed and use for train some language models without any concern of users

6. PROBABILISTIC MODELS OF TEXTS

Bayes Theorem

Prior belief $p(y)$ → data (x) → Posterior belief $p(y/x)$

Intuition:

Initial belief + new data → improved belief

Prior + likelihood → Posterior

Equation:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Where:

- $p(y|x)$: Posterior probability
- $p(x|y)$: Likelihood
- $p(y)$: Prior
- $p(x)$: Evidence

Noisy channel model of communication

Enables to

- Make assumptions about generative process
- Represents domain knowledge
- Express the belief in the form of priors

Bayesian inference:

- Inference of parameters and hyper parameters of the model (parameters and hypothesis are analogous)
- In other words estimation of hidden variables that might have generated the data

Simple generative process:

$$p(< w1, \ldots., wn > |theta)$$

Theta = communication intent (example: movie review)

Assumptions:

Surface form

- Focus on what is literally expressed not on what is implied
- Computational challenge(lexical noise)
  Ambiguity in language:
  Polysemy (different meaning in different context)
  Synonym

  Context: Social, physical, temporal

- Ignore word order:
  Each words are related to only few words
  Some words are important to convey meaning

Conditional Independence

Exchangeability

Independence and Conditional Independence

Independent Events:

X and Y are independent if occurrence of X is independent of Y and vice versa

$$p(x1) = p(x1) * p(x2)$$

Example: X: throwing a die Y: throwing a coin

Example for not independent events: X: height of a person Y: weight of a person

Conditional Independent Events:

Events X1 and X2 are conditionally independent given θ if and only if $p(\theta) > 0$, and

$$p(x1, x2|\theta) = p(x1|\theta).p(x2|\theta)$$

Example: X: height Y: vocabulary Z: age of a person

Here X, Y are dependent without the condition Z. But for a particular age Z=z it both X and Y are independent.

Consequences of conditional independence in language modelling

- Simplifies model specification and computation
- Avoids over-fitting

Effect of conditional independence in generative process:

$$p(< w1, \ldots., wn > |\theta) = p(w1, \ldots., wn|\theta) \cong \prod_i p(w_i|\theta)$$

Exchangeability

Probability of occurring k events is same for any sequence of occurring these events.

Example: probability of choosing 1 white ball a one red ball from an urn of 10 red ball and 5 white ball with replacement and without replacement.

With replacement: events are independent and exchangeable
Without replacement: not independent but exchangeable

Exchangeability of words in language modelling:

- Count of words is important not their order

7. TEXT CLASSIFICATION

Examples:

Sentiment Analysis - detection of attitudes
Spam detection
Authorship identification
Language identification
Subject categorization - Flat categories, hierarchical categories, and multi labeled classification

Input: Document
Output: Predict the class of document from given classes $C = \{c_1 \ldots . . c_k\}$

Classification methods:

Rule based text classification:
Example: In spam detection blacklist of email addresses or dollar symbol etc.

- Accuracy will be high if rules are carefully chosen by an expert
- But building and maintaining these rules is expensive

ML based text classification:

Input:
- a document d
- a fixed set of classes $C = \{c_1 \ldots . . c_j\}$
- A training set of m hand-labelled documents $(d_1, y_1) \ldots \ldots (d_m, y_m)$

Output:
- a learned classifier f: d → c

Examples:
- Naïve Bayes
- Logistic regression
- Neural networks
- k-Nearest Neighbours

Naïve Bayes Classifier

Assumption:

- Documents represented as bag of words
  Order of words is not important, count is important. We can use multinomial distribution here.

- Conditional Independence

By Bayes theorem for a document d and class c

$$P(c|d) = \frac{P(d|c)P(c)}{p(d)}$$

Maximum a posteriori (MAP) estimate of class (most likely class)

$$C_{MAP} = argmax_{c \epsilon C}\, P(c|d)$$

By Bayes rule,

$$C_{MAP} = argmax_{c \epsilon C}\, \frac{P(d|c)P(c)}{p(d)}$$

Denominator common for all classes, so we can discard in argmax,

$$C_{MAP} = argmax_{c \epsilon C}\, P(d|c)P(c)$$

$$C_{MAP} = argmax_{c \epsilon C}\, P(x1, \ldots., xn|\mathrm{c})P(c)$$

Calculation of $P(x1, \ldots., xn|\mathrm{c})$

- Computationally complex
- Need more observations

Conditional independence and bag of words assumptions will make it easy using multinomial naïve Bayes classifier

Multinomial Naive Bayes Classifier

- Bag of word assumptions
- Conditional independence assumptions

$$C_{MAP} = argmax_{c \epsilon C}\, P(x1, \ldots., xn|\mathrm{c})P(c)$$
$$C_{NB} = argmax_{c_j \epsilon C}\, P(c_j) \prod_{x_i \epsilon d} P(x_i|c_j)$$

Log Maximum a posteriori

Log a posteriori for overcoming floating point underflow. So log a posteriori estimate of Naïve Bayes is as follow,

$$c_{\mathrm{NB}} = \underset{c_j \in C}{\operatorname{argmax}} \left[\log P(c_j) + \sum_{i \in \mathrm{positions}} \log P(x_i|c_j) \right]$$

- Taking log doesn't change the ranking of classes.
- The class with highest probability also has highest log probability
- It's a linear model: Just a max of a sum of weights So naive Bayes is a linear classifier

Learning the Multinomial Naive Bayes Model

- From training corpus, extract Vocabulary
- Calculate $P(c_j)$ for each c_j in C (prior)
- Calculate $P(x_i|c_j)$ terms
  Create mega-document for topic j by concatenating all docs in this topic

Where,

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Problem with maximum likelihood:

If a word in test document not in training set, probability will be zero no matter what are other probabilities. For overcoming zero probability problem we can use add-k smoothing with symmetric or asymmetric priors. Add one to all count means we are giving symmetric priors to all types.

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1 \right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c) \right) + |V|}$$

Dealing Unknown words: ignore them

Why don't we build an unknown word model?

Knowing which class has more unknown words is not generally helpful!

Stop words: Removing stop words doesn't usually help, so in practice most NB algorithms use all words.

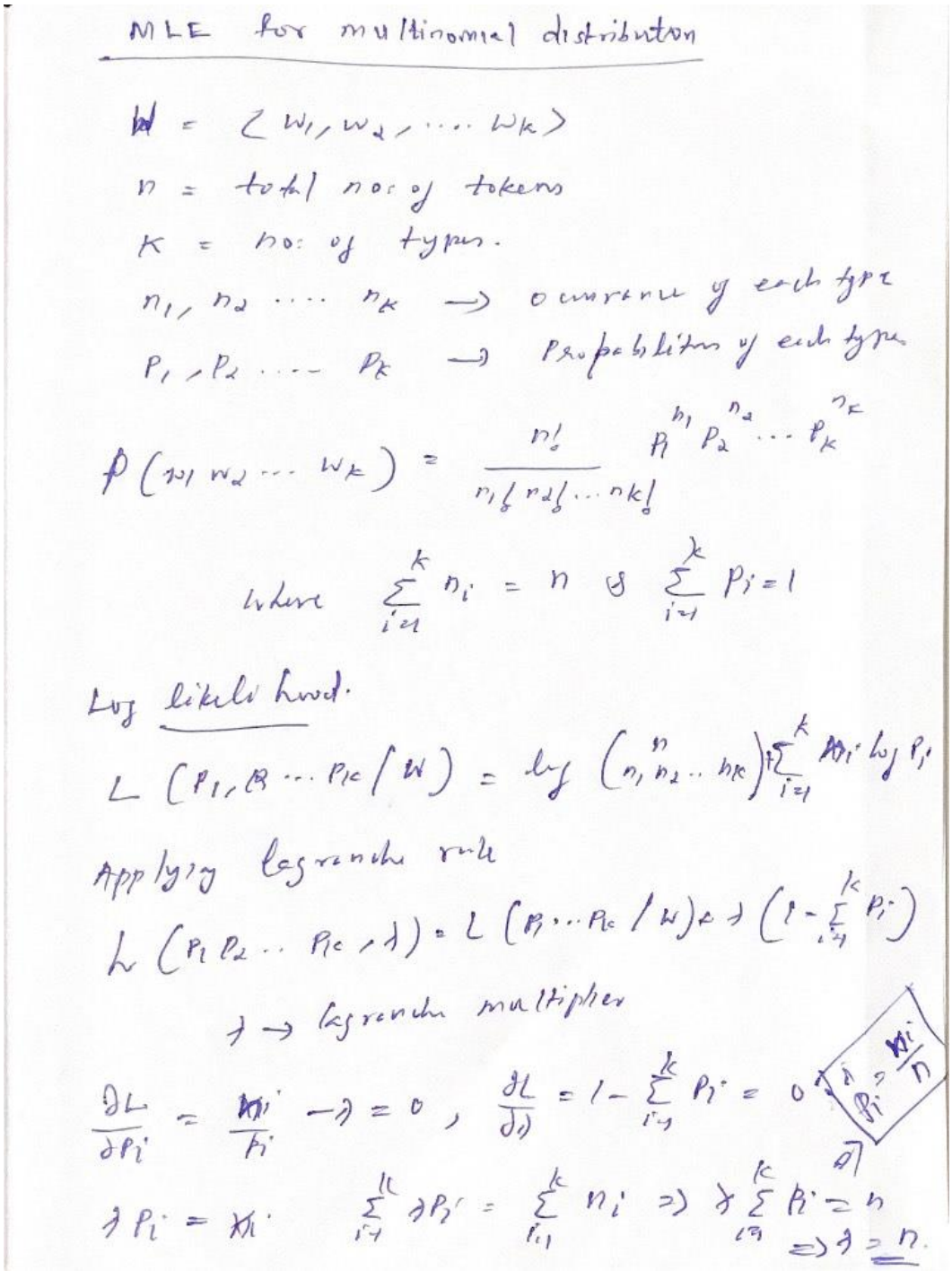
Binary multinomial NB for sentiment classification

- Occurrence of words is important than word frequency

Learning phase:

- First remove all duplicate words from d (not from all documents of a particular class)
- Calculate $P(c_j)$ for each c_j and $P(x_i|c_j)$ for each token as did in multinomial NB.
  NOTE: count of word may have more than 1 in entire concatenated document of a particular class

Maximum Likelihood estimate of multinomial distribution



Why Bayes theorem need to apply in Naïve Bayes classifier for calculating $P(c|d)$?

We need to calculate$P(c|d)$,

Using exchangeability assumption$d = < w1, \ldots., wk >$. So,

$P(c|d) = P(c| < w1, \ldots., wk >)$

But how we will calculate this? We can compute this using Bayes theorem and Conditional independence assumptions of Naïve Bayes classifier,

$$P(c|d)\ \alpha\ P(d|c) * P(c)$$

$$P(c| < w1, \ldots., wk >)\ \alpha\ P(< w1, \ldots., \mathrm{w}k > |c) * P(c)$$

$$P(c| < w1, \ldots., wk >) = P(w1|\mathrm{c}) \ldots. P(w\mathrm{k}|\mathrm{d}) * P(c)$$

Sentiment Classification: Dealing with Negation

- Negation can change positive to negative and negative to positive
- Simple baseline method for better classification add NOT_ to every word between negation and following punctuation

Sentiment Classification: Lexicons

- Lexicons is a pre-built word list with both positive and negative words
- It helps in classification when we don't have enough labelled training data
- We can add a feature that gets a count whenever a word from the lexicon occurs
- When training data is sparse or not representative of the test set, dense lexicon features can help

Advantages of Naïve Bayes Classifier

- Very Fast, low storage requirements
- Work well with very small amounts of training data
- Robust to Irrelevant Features
- Very good in domains with many equally important features
- Optimal if the independence assumptions hold:
- A good dependable baseline for text classification

Relationship of Naïve Bayes with language modelling

Naive Bayes has an important similarity to language modelling when we use features as all words from the document. So

Each class will be unigram language model

Assigning each sentence:

$$p(\mathrm{s} \mid \mathrm{c}) = \prod_{w \in s} p(\mathrm{w} \mid \mathrm{c})$$

So which class model maximize this probability, document will be in that class

Evaluation Metrics

- **Precision**
  Percentage of item detected as positive that are in fact positive

- **Recall**
  Percentage of item positive items which detected as positive

- **F-measure**
  A combined measure of precision and recall

$$F_\beta = (\beta^2 + 1)\frac{PR}{\beta^2 P + R}$$

When $\beta = 1$

$$F_1 = \frac{PR}{P + R}$$

Reading Assignment-3: Wikipedia-based Semantic Interpretation for Natural Language Processing

In this paper authors proposes Explicit Semantic Analysis (ESA) algorithm for fine-grained semantic interpretation. For this task they have used Wikipedia as their reference source of knowledge or getting contextual meaning of sentences. The model is capable of represent the meaning of any text in terms of Wikipedia-based concepts.

The model has the ability to address synonymy and polysemy which were considered as the two main problems in NLP and conventional models like Bag-of-words can't able to identify the common meaning among them. Since ESA follows concept-based representation, it will represent the problem of synonymy and polysemy to some extent.

For word relatedness, ESA model is considered as a measure of semantic relatedness instead of semantic similarity. On datasets used to benchmark relatedness of words, ESA outperforms other algorithms, including WordNet semantic similarity measures and skip-gram Neural Network Language Model (Word2vec).

SYNTAX OF LANGUAGE

- Language is more than bag-of words
- Adding and removing a word can change the meaning of a sentence completely

Example:

A sentence include

- Subject – noun phrase (sentence is about what or whom)
- Predicate – verb phrase (something about sentence)

Grammatical rules allow us to generalize from specific words to their categories.

When people learn a new word, they learn its syntactic usage

- What type it is
- Where to use it

Parts of Speech Tagging

Categorizing words in a text in correspondence with a particular part of speech, depending on the definition of the word and its context. Categorizing words makes analysis and generation of sentences makes easy

Some of the important part of speech tags are given below

POS tag	Symbol
Noun	NN
Verb	VB
Adjective	JJ
Adverb	RB
Preposition	IN
Conjunction	CC
Determiner	DT
Pronoun	PRP

Open class vs closed class

Open class – new words are added

Nouns, verbs, adjectives, adverbs are examples

Closed class – new words are very rarely added to this class

- Determiners: a, an, the
- pronouns: he, she, him
- Prepositions: on, under, over, near, by etc.

Challenges in POS tagging:

Words often have more than one POS, with respect to the context of the sentences.

For example, the word “back”

- The back door → adjective
- On my back → noun
- Win the voters back → adverb
- Promised to back the bill → verb

About 11% of the word types in Brown Corpus are ambiguous with regards to parts of speech. Also they are tend to be more common words.

For example: the

- I know that he is honest → IN
- Yes, that play was nice → DT
- You can’t go that far → RB

40% word tokens are ambiguous

Information inferred from neighbouring POS tags

- Noun can be preceded by “the”
- Verbs can be preceded by “can’t”
- Adjectives can come between “the” and a noun
- Determiners → a, the, many, no, five
- Prepositions→ for, to, in, without, before

Method to improve accuracy of POS tagging tasks?

- Introducing Feature based tagger
- Can tag some words like ‘the’, ‘a’ as determinant just by seeing them
- Upper case and Lower case of the same words convey different meaning.
- Certain words as Prefix or Suffix can determine the POS of the words. For example, ‘un’ as prefix and ‘ly’ as Suffix.
- Use probabilistic model to tagging the tokens.

Noun as verb example: google

Constituents

Constituents are group of continuous words and are non-crossing. That is if two constituents share one word, then one of them must completely contain the other. Each word in the sentence will be a constituent.

Example:

I have seen blue elephants

- Blue elephants → valid constituent
- Seen blue → not a valid constituent
- Seen blue elephants → valid constituent

Generation of Sentences

1. Tree structure

Generate the tree structure first, then fill the leaf nodes with terminals

Grammatical Rules:

- Sentence generate noun followed by verb (S → N V)
  Example: Birds fly
- Sentence generate noun phrase followed by verb phrases
  S → NP VP
  Noun phrase → proper noun or determiner followed by common noun (NP → PN | DT CN)
  Verb phrase → verb followed by noun phrase (VP → V NP)

  NOTE:
  Verbs
  - Intransitive - no direct object
  - Transitive verbs – direct object present

  S → NP VP
  NP → DT CN
  NP → PN
  VP → V NP
  DT → the | a
  CN → child | cat | dog
  PN → Samantha | Jorge | Min
  V → took | saw | liked | scared | chased

- Optional categories
  NP → N | DT N | JJ N | DT JJ N (JJ – adjective)
  It can also be written as
  NP → (DT) (JJ) N

  VP → V (NP) (P) (NP) (P – preposition at, in, on etc.)

Grammar changes to

S → NP VP
NP → DT CN
NP → PN
VP → V (NP) (P) (NP)
DT → the | a
CN → child | cat | dog
PN → Samantha | Jorge | Min
P → to | for | from | in
V → took | saw | liked | scared | chased | gave

- Incorporating preposition
  Prepositions gives the details about position, time, special relation etc.
  Preposition phrases (PP) alters both NP and VP

  S → NP VP
  NP → (DT) (JJ) N (PP)
  VP → V (NP) (PP)
  PP → P (NP)

  PP Ambiguity
  The boy saw the woman with the telescope

  This can be interpret as two ways
  1. Boy saw with telescope -- VP → V NP PP
  2. Women with telescope -- NP → DT N PP
- Repetition (*)

  (JJ*) = a sequence of zero or more JJ
  In this case adjective ordering is a selection preference
  Adjective ordering
  det < number < size < colour < purpose < noun
  Strength < material < noun
  Origin < noun

- Nested Sentences

  VP → V (NP) (NP) (C S) (PP*)

  (C S) – Conjunction followed by new sentence

Recursion

Why recursion is important in language perspective?

Recursion provides a brief abstraction of a particular phenomenon. So in language sentences can be break down into parts which is similar. By recursion we can join NPs with conjunction, VPs with conjunction, and PPs with conjunction.

Fractals?

Repeating a simple process over and over again

Mete rule for conjunction

X and X, X can be NP, VP, PP or an entire sentence

Auxiliary verbs

VP → Aux VP

Can use multiple auxiliaries inside a sentence

Is recursion unlimited?

In day to day life we are not using an infinite length sentence even it possible grammatically

Final Grammar

S → NP VP | CP VP

NP → (DT) (JJ*) N (CP) (PP*)

VP → V (NP) (NP) (PP*) | V (NP) (CP) (PP*)

PP → P NP

CP → C S

Exercise:

What rules are needed to generate these three sentences?

1. The small dog of the neighbours brought me an old tennis ball.

2. That wugs have three eyes is unproven by scientists.

3. I saw the gift that the old man gave me at the meeting.

"Colorless green ideas sleep furiously"

This is an example of a sentence that is grammatically well-formed, but semantically nonsensical

Relation between syntax and semantics are important and complementary

Information extraction

To convert unstructured textual information to some structured form

Application example:

Resume filtering

- With key word search for Microsoft excel skill it will retrieve both Microsoft as employer and as skill
- But with information extraction we case specify which information we want to filter. In this case skill is information

Entities:

Basic unit of information. It an object or set of objects in real world.

General entity types: PERSON, ORGANIZATION, LOCATION

Entities can be domain specific

Example:

Resumes: EMPLOYER, DEGREE, EDUCATIONAL_INSTITUTE, DESIGNATION

Entity Mentions:

Entities are referenced in text through entity mentions

Entity mentions are generally of 3 types

1. Named mentions: Names of people, organizations; often expressed through proper nouns

   Examples: Sachin Tendulkar, Reliance Industries Ltd., New Delhi

2. Nominal mentions: Entity mentions expressed through common nouns

   Examples: batsman, company, city

3. Pronoun mentions: Entity mentions expressed through pronouns

   Examples: he, they, her, it

Named Entity Recognition (NER)

Extracting named mentions from the text.

Examples:

Russia – geopolitical entity

Sachin, he – person

Reliance industries – organization

Task: named entity extraction

Techniques:

1. Rule-based techniques – hand crafted lexical, syntactic or semantic rules
2. Unsupervised or semi-supervised - learning of gazetteers
3. Supervised – features-based, deep learning

Supervised technique for named entity extraction

Sequence labelling

Each word in a sentence is labelled as is it a part of any entity mention and if yes its type

Labelling strategies:

- IO : Inside / Outside
  Example: Air/I-ORG India/I-ORG women/O
- BIO : Begin / Inside / Outside
  Example: Air/B-ORG India/I-ORG
  Advantage: we can differentiate two entity mentions of same type if the come together
- BILOU : Begin / Inside / Last / Outside / Unit
  Example: Tata/B-ORG Consultancy/I-ORG Services/L-ORG , /O India/U-GPE

Models using for named entity extraction:

1. Conditional Random Fields

   Sequence labelling extension of multinomial logistic regression

   Linear chain CRFs are used for named entity extraction

2. Deep learning based models
   LSTM, encoder – decoder

Evaluation

Calculate precision, recall and F1 measure for each entity type and calculate macro or micro precision, recall and F1 measure from them.

Language parsing

How human language is different from computer language?

- No types for words
- No brackets around phrases
- Some words and phrases human languages are ambiguous
- Most of the sentences contains implied information

What is parsing?

Parsing means associating tree structures to a sentence, given a grammar. For a given sentence it can be one, many or none. Grammar won't specify how the parse tree will be constructed, rather it will gives the abstraction of how sentences are generated

Applications of parsing

- Grammar checking
- Question answering (text to SQL)
- Machine translation E.g., word order – SVO vs. SOV
- Information extraction
- Speech generation
- Speech understanding

Context-free grammars

A context-free grammar is a 4-tuple (N,Σ, R,S)

N: non-terminal symbols

Σ: Terminal symbols (disjoint from N)

R: rules ($A \rightarrow \beta$), where β is a string from $(\Sigma \cup N)^*$

S: start symbol from N

Example: ["the", "child", "ate", "the", "cake", "with", "the", "fork"]

S -> NP VP

NP -> DT N | NP PP

PP -> PRP NP

VP -> V NP | VP PP

DT -> 'a' | 'the'

N -> 'child' | 'cake' | 'fork'

PRP -> 'with' | 'to'

V -> 'saw' | 'ate'

Phrase structure grammar

S -> NP VP | Aux NP VP | VP

NP -> PRON | Det Nom

Nom -> N | Nom N | Nom PP

PP -> PRP NP

VP -> V | V NP | VP PP

Det -> 'the' | 'a' | 'this'

PRON -> 'he' | 'she'

N -> 'book' | 'boys' | 'girl'

PRP -> 'with' | 'in'

V -> 'takes' | 'take'

Parsing Approaches

1. Top-down
2. Bottom-up

Bottom up

- explores options that won't lead to a full parse
- Example: shift-reduce (srparser in nltk)
- Example: CKY (Cocke-Kasami-Younger)

Top down

- explores options that don't match the full sentence
- Example: recursive descent (rdparser in nltk)
- Example: Early parser

Shift reduce parsing

Dynamic Programming

In parsing same process are repeating, so caching intermediate results improves the complexity

CKY (Cocke-Kasami-Younger) Parse

- Uses dynamic programming
- It is a bottom up search
- Requires normalized (binarized) grammar
- Space complexity – O(n*n)
- Time complexity for single parsing – O(n*n*n)

Chomsky Normal Form

Non-binary productions to binary productions

X $\rightarrow$ Y Z : Binary form

S $\rightarrow$ Aux NP VP

Becomes

S $\rightarrow$ R1 VP

R1 $\rightarrow$ Aux NP

Limitation of CKY parsing

- Same language different structure
- If the grammar had to be converted to CNF, then the final parse tree doesn't match the original grammar. However, it can be converted back using a specific procedure
- No way to perform syntactic disambiguation

Probabilistic Context Free Grammars

Why probabilistic parsing required?

To overcome uncertainty and ambiguity in the deterministic parsing. Deterministic parsing check whether this sentence generated by this grammar or not.

Probabilistic parsing

Given a grammar G and a sentence s, let T(s) be all parse trees that correspond to s

Task 1:

Find which tree t among T(s) maximizes the probability $p(t)$

$$p(t) = \prod_{i=1}^{n} p(\alpha i \rightarrow \beta i)$$

Most likely parse is

$$\underset{t \in T}{argmax} \quad p(t)$$

Task 2:

Find the probability of the sentence p(s) as the sum of all possible tree probabilities $p(t)$

Probabilities can be learned from a labelled corpus with MLE estimates

$$P(\alpha \rightarrow \beta) = count(\alpha \rightarrow \beta)/\text{count}(\alpha)$$

$$\text{Example: } P(S \rightarrow NP\ VP) = \text{Count}\ (S \rightarrow NP\ VP)/\text{Count}(S)$$

Evaluation of parsing

- For evaluation of syntactic parsing Tree Bank corpus is used
- Precision, recall and F1 measure used as metrics
- Crossing bracket is another metrics which checks percentage of groupings crossed

Inter judge agreement and cohen's Kappa

Agreement vs expected agreement between annotators for a given task

Cohen's kappa,

$$k = \frac{P(A) - p(E)}{1 - p(E)}$$

$$p(A) = agreement\ between\ annotators$$

$$P(E) = expected\ agreement$$

If K > 0.7 agreement is considered as high

Judge agreement on a binary classification task is 60%, is this high?

P (A) = 0.6

P (E) = 0.5 (agreement by chance)

K = 0.2, not high

Can we use the value of agreement to decide the task is difficult or not??

If task is easy the agreement value will be close to 1. If it is difficult, inter judgement agreement is less.

Dependency Grammars

Assumption: words are dependent on each other.

Characteristics

- They define lexical/syntactic dependencies between words
- The top-level predicate of a sentence is the root
- Simpler to parse than context-free grammars
- Particularly useful for free word order languages

SEMANTICS

Word meaning and similarity

What is meaning of a word?

It's depends on how you use it

Lemma and word form

In morphology and lexicography, a lemma is the canonical form, dictionary form, or citation form of a set of words.

Word from is the "inflected" word as it appears in text

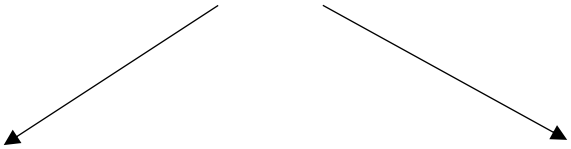
Example: lemma → Bank

Word form → Banks

Lemma may have different senses or meaning

Homonymy

(Words that share a form but have unrelated, distinct meanings)



	Homographs	Homophones
Example:	bank/bank, bat/bat	Write / right. Piece / peace

Polysemy

Words having related meaning

Bank →A financial institution

→The building belonging to a financial institution

Systematic polysemy (metonymy)

Systematic relationship between senses

School, university, hospital (building and organization)

Synonyms

Word that have the same meaning in some or all contexts. It is relation between senses rather than words.

Example: couch / sofa, big / large

Antonym

Senses that are opposites with respect to one feature of meaning

Examples: hot/cold, up/down

Word Similarity

Term document matrix

Matrix with each cell represents the count of term *t* in a document *d.*

Each document will be a count vector.

Using this vector (across column and row) we can calculate similarity. If vectors corresponding to two documents are similar, then documents are similar.

If documents corresponding to two words are similar, then two words are similar

Term Context Matrix

Instead of using entire matrix use use smaller contexts

Instead of counts

- tf-idf → term-document matrix
- Positive Pointwise Mutual Information (PPMI) → term context matrix

Reading assignment -4: The Naive Bayes Model, Maximum-Likelihood Estimation, and the EM Algorithm

Naive bayes algorithm estimate probability from the data with conditional independance and bag of word assumption

by naive bayes algorith:

class label of a document x

$$\hat{y} = \underset{y \in Y}{\arg\max}\ P(Y/x)$$

by bayes theorem

$$\hat{y} = \underset{y \in Y}{\arg\max}\ P(x/y) \cdot P(y)$$

by bag of word assumption and conditional independance.

$$\hat{y} = \underset{y \in Y}{\arg\max} \prod_{i=1}^{d} P(x_i/y) \cdot P(y).$$

this $P(x_i/y)$ and $P(y)$ can be calculated from the given data point if data is completely labelled. (data is fully observed)

The Expectation maximization for Naive Bayes

If data are not fully observed how to calculate $P(y/x)$.

EM helps get this probability

If label for any example $\underline{x}$ is missing the probability of that model under NB model can be calculated by marginalizing out the labels

$$P(\underline{x}) = \sum_{y=1}^{k} P(x, y) = \sum_{y=1}^{k} \left(q(y) \prod_{j=1}^{d} q_j(x_j/y) \right)$$

$$q_j(x_j/y) = P(X_j = x_j / Y = y)$$

log likelihood of all examples

$$L(\theta) = \sum_{i=1}^{n} \log P(x_i)$$

$$= \sum_{i=1}^{n} \log \sum_{y=1}^{k} q(y) \prod_{j=1}^{d} q_j(x_j^{(i)}/y)$$

($q \rightarrow$ Probability measure)

Reading Assignment -5: Information Retrieval and the Philosophy of Language

In this article author discuss the importance of language understating in the task of information retrieval. Language understanding is important for document and query representation which facilitate affective information extraction. Philosophy of Language has been primarily concerned with the propositional content of language.

Illocutionary acts:

Illocutionary acts refers to the use of a sentence to express an attitude with a certain function. There are five types of illocutionary acts

- Assertives: in which we tell others (truly or falsely) how things are.
- Directives: in which we attempt to get others to do things.
- Commissives: in which we commit ourselves to doing specific things.
- Declarations: in which we bring about changes in our world by our utterance.
- Expressives: in which we express our personal feelings and attitudes.

Illocutionary Acts and Information Retrieval:

Illocutionary Acts are plays an important role in understanding electronic messages. And message transmissions are special case of information retrieval. One such application is COORDINATOR which structure electronic messages according to the Illocutionary Act under which it falls. So messages can compared to a document which contains lots of useful information. So the taxonomy of Illocutionary Acts becomes a classification scheme for the representation of performative messages, and and the links that were made with other messages that were used to perform the same act become useful ways of clustering the messages when they are stored as documents.

MACHINE TRANSLATION

Translating one language to another

Language differences

- Word order in phrases and sentences are differ for different languages
- One word in a language represent with more number of words in another language
- Colour names differ in different language. In some languages more distinct colours may be there

Translation: Art or Science?

Translation is an art, an exact word by word translation won't produce good sentence in target language. Rather a translator need to capture the context of source language sentence and then translate it in to target language. A translator need many years of experience in using both language in order to capture the information content in a sentence.

Evaluation

Adequacy and fluency are the two important matrix for evaluating a machine translational model

Adequacy: Does the output convey the same meaning as the input sentence? Is part of the message lost, added, or distorted?

Fluency: Is the output good fluent English? This involves both grammatical correctness and idiomatic word choices.

BLUE

- N-gram overlap between machine translation output and reference translation
- Compute precision for n-grams of size 1 to 4
- Add brevity penalty (for too short translations)
- Typically computed over the entire corpus, not single sentence

$$\text{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$