Multidisciplinary Natural Language Processing

Prof. Mukta Sunil Taklikar

Composed by - Anisha Gunjal
Kinjal Sanghvi
Sneha Gathani

Unit 1 - Multidisciplinary Natural Language Processing

Table of Contents

Introduction

Natural language processing (NLP) is the ability of a computer program to understand human speech as it is spoken. It is a component of artificial intelligence (AI) – actually another big trend these years. In other words, Natural language processing is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human languages. It is a computer activity in which computers analyze, understand and generate natural language. This includes the automation of any or all linguistic forms, activities, or methods of communication, such as conversation, correspondence, reading, written composition, publishing, translation, lip reading, and so on. In fact, natural language processing is one aspect of machine learning, big data, and artificial intelligence that has the potential to truly change everything. NLP is concerned with questions involving three dimensions: language, algorithm and problem.

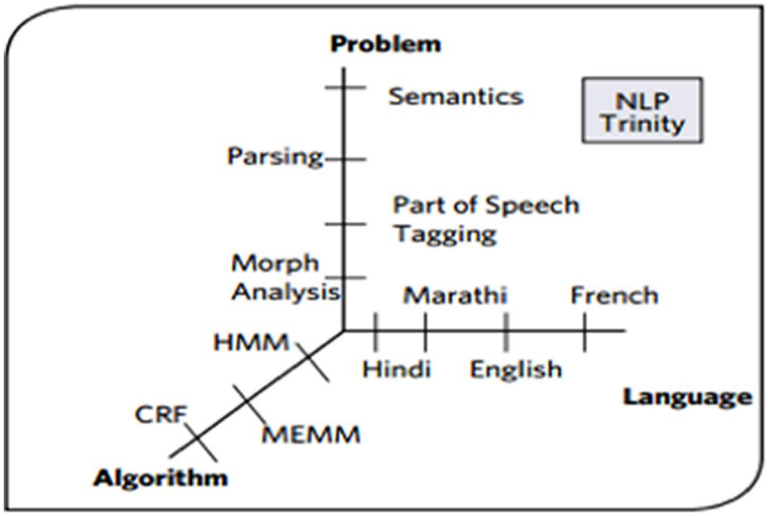


Figure 1: Three Dimensions of NLP

On the language axis are different natural languages and linguistics. The problem axis mentions different NLP tasks like morphology, part of speech tagging etc.

The algorithm axis depicts mechanisms like HMM, MEMM, CRF etc. for solving problems. The goal of natural language analysis is to produce knowledge representation structures like predicate calculus expressions, semantic graphs or frames.

This processing makes use of foundational tasks on the problem axis like morphology analysis, Part Of Speech Tagging, Named Entity Recognition, both shallow and Deep Parsing, Semantics Extraction, Pragmatics and Discourse Processing.

How is natural language processing used nowadays

1. Information Extraction

Many important decisions in financial markets are increasingly moving away from human oversight and control. Algorithmic trading is becoming more popular, a form of financial investing that is entirely controlled by technology. But many of these financial decisions are impacted by news, by journalism which is still presented predominantly in English. A major task, then, of NLP has become taking these text announcements, and extracting the info in a format that can be put into algorithmic trading decisions. For example, news of a merger between companies can have a big impact on trading decisions, and the speed at which the the merger, players, prices, who acquires who, can be incorporated into a trading algorithm can have profit implications of millions of dollars.

2. Machine Translation

You may also have used Natural language processing for yourself if you have ever used the “translate” link inside Facebook to translate a foreign language into your own. Google is a company at the forefront of machine translation, using a proprietary statistical engine for its Google translate service. The challenge with machine translation technologies is not in translating words, but in preserving the meaning of sentences, a complex technological issue that is at the heart of Natural language processing.

3. Fighting Spam

Another use of NLP is text classification. Google and other email providers use it to determine if an email is spam or not. Spam filters have become important as the first line of defense against the unwanted email.

4. Summarization

Other NLP programs are being developed and used. Those can automatically summarize long documents or extract relevant keywords for searching. The legal system is using these types of applications, for example, to help lawyers sort through thousands of pages of documents in any given legal case to find relevant information. Information overload is a real phenomenon in our digital age, and already our access to knowledge and information far exceeds our capacity to understand it. This is a trend that shows no sign of slowing down, and so an ability to summarise the meaning of documents and information is becoming increasingly important. It really helps to absorb the pertinent information from vast amounts of data.

5. Emotional meaning

Marketers are using NLP for sentiment analysis, combining the millions of tweets and other social media messages to determine how users feel about a particular product or service. It has the potential to turn all of Twitter or Facebook into one giant focus group.

6. Question Answering

Search engines put a lot of information at our fingertips, but are still generally quite primitive when it comes to actually answering specific questions asked by humans. It is getting better and better year by year. And companies are predicting that chatbots; another growing trend will be able to take over some customer-service functions in as little as five years, providing automated, real-time responses to simple customer-service problems and questions

Stages of NLP

Traditionally, NLP - of both spoken and written language has been regarded as consisting of the following stages:

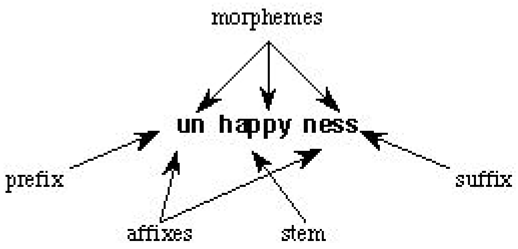
1. Phonology and Phonetics (processing of sound)

Phonetics is about the acoustic and articulatory properties of the sounds which can be produced by the human vocal tract, particularly those which are utilised in the sound systems of languages. Examples are bun, pun, put, putt, butt, etc.

Phonology includes syntactic understanding of homophones or similar sounding words.Examples are bank(Banks of rivers) and bank(Money in the bank).

As humans, we do not falter in this understanding because context and world knowledge helps in understanding the syntax but it is not easy for a machine to understand this.

2. Morphology (processing of word forms)
Morphology is the study of the structure and formation of words. Its most important unit is the *morpheme*, which is defined as the "minimal unit of meaning".
Consider a word like: "unhappiness". This has three parts:



There are three morphemes, each carrying a certain amount of meaning. *un* means "not", while *ness* means "being in a state or condition". *Happy* is a *free morpheme* because it can appear on its own as a root word. *Bound morphemes* like “un”, “a”, “ness”, etc. have to be attached to a free morpheme, and so cannot be words alone. This can be explained as you cannot have sentences like "Jason feels very un ness today". Other examples of atomic morphemes are “sends”, “resend”, “sending”, etc.

Languages differ in their morphological richness. Languages like Dravidian, Turkish, Hungarian and Slavic are examples of morphologically rich languages while languages like Chinese and English are examples having relatively simpler morphology.

3. Lexicon (Storage of words and associated knowledge)
The lexicon contains information about particular idiosyncratic properties of words; eg. what sound or orthography goes with what meaning like sent(not sended), assigning parts-of-speech(storm can be noun or verb), semi-productive meaning extensions and relations(animal could mean herbivorous, carnivorous or omnivorous).

Example of the word “dog” stored in the lexicon would look like:
c.i. POS (Noun)
c.ii. Semantic Tag (Animal, 4-legged)
c.iii. Morphology (takes ‘s’ in plural)
Words typically have multiple meanings even in the same part of speech. Dog, for example, means an animal and a very detestable person.

4. Parsing (Processing of structure)
Parsing resolves a sentence into its component parts to describe their syntactic roles. Syntax concerns the way in which words can be combined together to form correct grammatical sentences; eg. “revolutionary new ideas appear infrequently” is grammatically correct in English, “colourless green ideas sleep furiously” is grammatical but nonsensical, while “ideas green furiously colourless sleep” is ungrammatical. Various parsers exist like top-down, bottom-up, LR(0), LR(1), LALR parsers, etc.

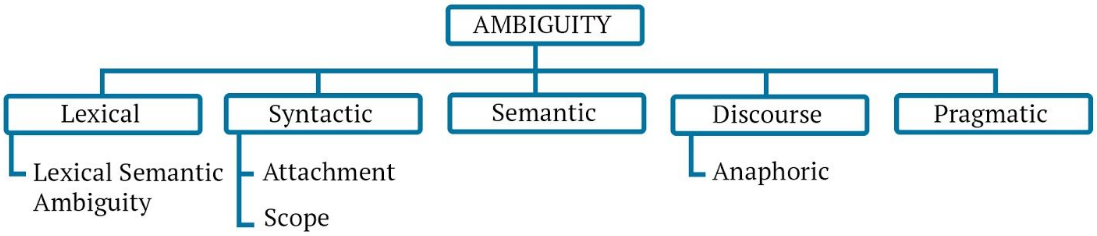
5. Semantics (Processing of meaning)
Semantics is about the manner in which lexical meaning is combined morphologically and syntactically to form the meaning of a sentence. Mostly, it is regular, productive and rule-governed. Because the meaning of a sentence is a combination of the meaning of its words, syntactic information is important for interpretation – it helps us work out what correlates with what – but other information, such as punctuation or intonation, pronoun reference, etc, can also play a crucial part.

6. Pragmatics (Processing of user intention, modeling etc.)

This is one of the hardest problems of NLP and has seen very little progress. The problem involves processing user intention, sentiment, belief world, modals etc. It is a complex task as every individual would have the understanding of each of these in their own, individual ways.

7. Discourse (Processing of connected text)
This is the task of processing connected sentences. When many sentences are spoken together in the form of a speech or written in the form of an essay, the reference of what refers to what, mostly the pronouns and the connectivity of the objects to their subjects needs to be understood to make semantic sense of the sentences and this is performed using discourse.

Ambiguity and Resolutions[1]



There are ambiguities in each of the stages of NLP as discussed above and they broadly categorised in 5 forms are explained below:

1. Lexical Ambiguity
It is the ambiguity of a single word. A word can be ambiguous with respect to its syntactic class.Eg: book, study.
For eg: The word silver can be used as a noun, an adjective, or a verb. This can be seen in the following sentences - “She bagged two silver medals”, “She made a silver speech”, “His worries had silvered his hair”.
Resolution - Category disambiguation i.e, parts-of-speech tagging.
As many words may belong to more than one lexical category, part-of-speech tagging is the process of assigning a part-of-speech or lexical category such as a noun, verb, pronoun, preposition, adverb, adjective etc. to each word in a sentence.

Type of Lexical ambiguity :
1.1 Lexical Semantic Ambiguity
Also called the phonetics and phonology ambiguity. It occurs when a single word is associated with multiple senses. Eg: bank, pen, fast, bat, cricket etc.
Sentences having two varied meanings of the word “tank” - “The tank was full of water”, “I saw a military tank”. The occurrence of “tank” in both sentences corresponds to the syntactic category noun, but their meanings are different.
Resolution - Word Sense Disambiguation (WSD) techniques
It aims at automatically assigning the meaning of the word in the context in a computational manner.

2. Syntactic Ambiguity
The ambiguities occuring in structures are syntactic ambiguities.
Structural ambiguity is of two types:
2.1 Scope Ambiguity
Scope ambiguity involves operators and quantifiers.
Example: Old men and women were taken to safe locations.
The scope of the adjective (i.e., the amount of text it qualifies) is ambiguous. That is,whether the structure (old men and women) or ((old men) and women)

[1] Ambiguities in Natural Language Processing, Anjali M K1, Babu Anto P2

The scope of quantifiers is often not clear and creates ambiguity.

Another example - Every man loves a woman.
The interpretations can be, For every man there is a woman and also it can be there is one particular woman who is loved by every man.

2.2 Attachment Ambiguity
A sentence has attachment ambiguity if a constituent fits more than one position in a parse tree. Attachment ambiguity arises from uncertainty of attaching a phrase or clause to a part of a sentence.
Example: The man saw the girl with the telescope.
It is ambiguous whether the man saw a girl carrying a telescope, or he saw her through his telescope.
The meaning is dependent on whether the preposition 'with' is attached to the girl or the man.
Another example: Buy books for children.
Preposition Phrase 'for children' can be either adverbial and attach to the verb buy or adjectival and attach to the object noun books.

3. Semantic Ambiguity
This occurs when the meaning of the words themselves can be misinterpreted.Even after the syntax and the meanings of the individual words have been resolved, there are two ways of reading the sentence.
Example - Seema loves her mother and Sriya does too.
The interpretations can be Sriya loves Seema's mother or Sriya likes her own mother.
Semantic ambiguities born from the fact that generally a computer is not in a position to distinguishing what is logical from what is not.

Like another example - The car hit the pole while it was moving.
The interpretations can be, The car, while moving, hit the pole and The car hit the pole while the pole was moving. The first interpretation is preferred to the second one because we have a model of the world that helps us to distinguish what is logical (or possible) from what is not. To supply to a computer a model of the world is not so easy.
Example - We saw his duck.
Duck can refer to the person's bird or to a motion he made.
Semantic ambiguity happens when a sentence contains an ambiguous word or phrase.

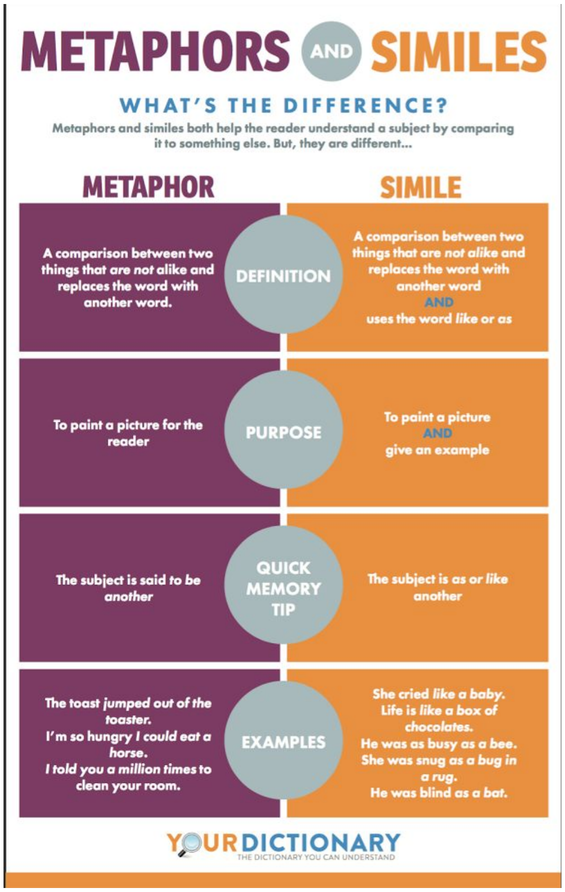
4 Discourse
Discourse level processing needs a shared world or shared knowledge and the interpretation is carried out using this context.
Type of discourse ambiguity is Anaphoric ambiguity.
4.1 Anaphoric Ambiguity
Anaphoras are the entities that have been previously introduced into the discourse.
Example - The horse ran up the hill. It was very steep. It soon got tired.
The anaphoric reference of 'it' in the two situations cause ambiguity. Steep applies to surface hence 'it' can be hill. Tired applies to animate object hence 'it' can be horse.

5. Pragmatic Ambiguity
Pragmatic ambiguity refers to a situation where the context of a phrase gives it multiple interpretation. One of the hardest tasks in NLP as the problem involves processing user intention, sentiment, belief world, modals etc.- all of which are highly complex tasks.
Example - Tourist (checking out of the hotel): Waiter, go upstairs to my room and see if my sandals are there; do not be late; I have to catch the train in 15 minutes.
Waiter (running upstairs and coming back panting): Yes sir,they are there.
Clearly, the waiter is falling short of the expectation of the tourist, since he does not understand the pragmatics of the situation.
I love you too.

Pragmatic ambiguity arises when the statement is not specific, and the context does not provide the information needed to clarify the statement. Information is missing, and must be inferred.
Example - This can be interpreted as
I love you (just like you love me)
I love you (just like someone else does) I love you (and I love someone else)
I love you (as well as liking you)

Metaphors

Metaphor is a rhetorical figure of speech that compares two subjects without the use of “like” or “as.” Metaphor is often confused with simile, which compares two subjects by connecting them with “like” or “as” (for example: “She’s fit as a fiddle”). While a simile states that one thing is like another, a metaphor asserts that one thing is the other, or is a substitute for the other thing.
A metaphor asserts a correlation or resemblance between two things that are otherwise unrelated. The English word “metaphor” originates from the Greek metaphorá, which means “to transfer” or “to carry over.” Indeed, a metaphor transfers meaning from one subject on to another so that the target subject can be understood in a new way.
Rhetoricians have further elaborated on the definition of metaphor by separating and naming the two key elements. There are a few different sets of names for these two parts: they can be called the “tenor” and the “vehicle”, the “ground” and the “figure”, or the “target” and the “source”. Consider this famous example of a metaphor from Shakespeare’s “As You Like It”:
All the world’s a stage,
And all the men and women merely players.
In this example, the world is the primary subject, and it gains attributes from the stage (ie, from theater). Thus, in the binary pairs, the world is the “tenor,” the “ground,” and the “target,” while the stage is the “vehicle,” the “figure,” and the “source.”



METAPHORS AND SIMILES

WHAT’S THE DIFFERENCE?

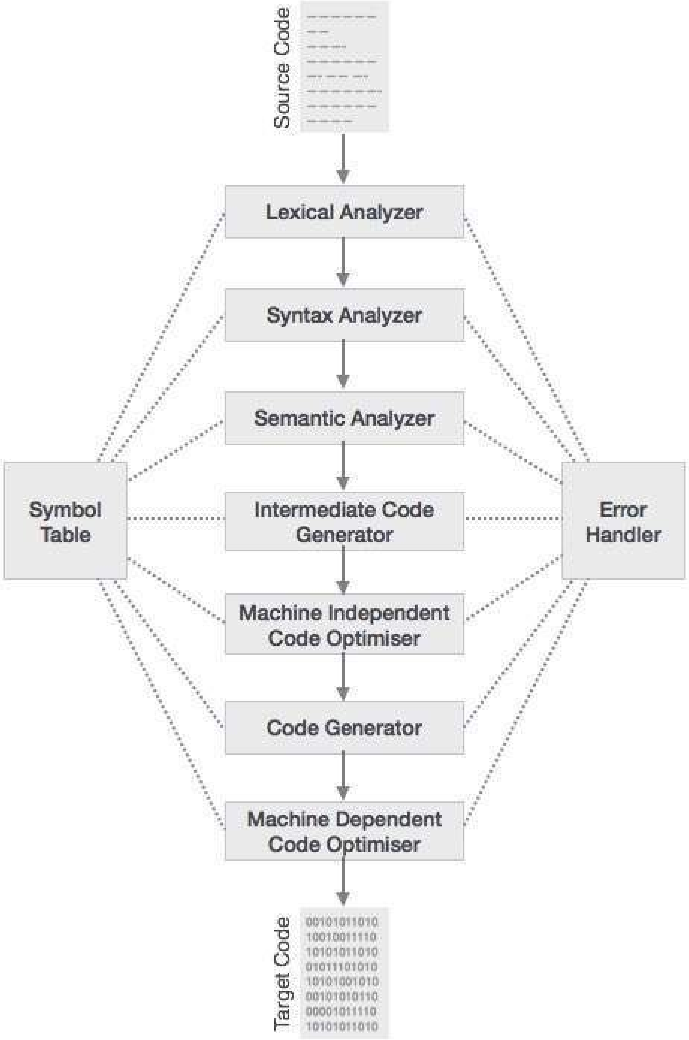
Metaphors and similes both help the reader understand a subject by comparing it to something else. But, they are different...

METAPHOR		SIMILE
A comparison between two things that *are not* alike and replaces the word with another word.	DEFINITION	A comparison between two things that are *not alike* and replaces the word with another word AND uses the word *like* or *as*
To paint a picture for the reader	PURPOSE	To paint a picture AND give an example
The subject is said *to be another*	QUICK MEMORY TIP	The subject is *as* or *like* another
The toast *jumped out of the toaster.* I’m so hungry *I could eat a horse.* *I told you a million times* to clean your room.	EXAMPLES	She cried *like a baby.* Life is *like a box of chocolates.* He was as busy *as a bee.* She was snug *as a bug in a rug.* He was blind as *a bat.*

YOURDICTIONARY
THE DICTIONARY YOU CAN UNDERSTAND

Theories of Parsing

Parsing is commonly known as Syntactic Analysis. It is the process of analyzing strings of symbols either in natural language or in computer languages by considering the rules of formal grammar.
Basically parsing used to identify structure in data, in simple words we say it determines whether the input data has some predetermined structure and respond accordingly.
Parsing is an important phase in the process of compiler and comes from Latin word "pars" which means part of speech. Parser or Syntax Analyzer takes the tokens produced from lexical analyzer.



Stages of Parsing

Ruled Based Parsing
Parsing consists of two phases. The analysis phase and the synthesis phase. The analysis phase is made of the lexical analyser, syntax analyser and the semantic analyser. All the other stages come under the synthesis phase.

Lexical Analysis : The first phase of scanner works as a text scanner. This phase scans the source code as a stream of characters and converts it into meaningful lexemes. Lexical analyzer represents these lexemes in the form of tokens as: <token-name, attribute-value>
Syntax Analysis : The next phase is called the syntax analysis or parsing. It takes the token produced by lexical analysis as input and generates a parse tree (or syntax tree). In this phase, token arrangements are checked against the source code grammar, i.e. the parser checks if the expression made by the tokens is syntactically correct.
Semantic Analysis : Semantic analysis checks whether the parse tree constructed follows the rules of language. For example, assignment of values is between compatible data types, and adding string to an

integer. Also, the semantic analyzer keeps track of identifiers, their types and expressions; whether identifiers are declared before use or not etc. The semantic analyzer produces an annotated syntax tree as an output.
Intermediate Code Generation : After semantic analysis the compiler generates an intermediate code of the source code for the target machine. It represents a program for some abstract machine. It is in between the high-level language and the machine language. This intermediate code should be generated in such a way that it makes it easier to be translated into the target machine code.
Code Optimization : The next phase does code optimization of the intermediate code. Optimization can be assumed as something that removes unnecessary code lines, and arranges the sequence of statements in order to speed up the program execution without wasting resources (CPU, memory).
Code Generation : In this phase, the code generator takes the optimized representation of the intermediate code and maps it to the target machine language. The code generator translates the intermediate code into a sequence of (generally) re-locatable machine code. Sequence of instructions of machine code performs the task as the intermediate code would do.

Classification of parser based on the way the production rules are implemented.
Types Of Parsers:

1. Top down Parser:
The Parser starts constructing the parse tree from the start symbol and then tries to transform the start symbol to the input.
Subtypes:
1.1 Recursive Descent Parsing
1.2 Backtracking

1.1 Recursive Descent Parsing :
Recursive descent is a top-down parsing technique that constructs the parse tree from the top and the input is read from left to right. It uses procedures for every terminal and nonterminal entity. This parsing technique recursively parses the input to make a parse tree, which may or may not require backtracking. But the grammar associated with it (if not left factored) cannot avoid back-tracking. A form of recursive-descent parsing that does not require any back-tracking is known as predictive Parsing. This parsing technique is regarded recursive as it uses context-free grammar which is recursive in nature.
1.2 Backtracking :
Top- down parsers start from the root node (start symbol) and match the input string against the production rules to replace them (if matched). To understand this, take the following example of CFG:
S → rXd | rZd
X → oa | ea
Z → ai
For an input string: read, a top-down parser, will behave like this:
It will start with S from the production rules and will match its yield to the left-most letter of the input, i.e. ‘r’. The very production of S (S → rXd) matches with it. So the top-down parser advances to the next input letter (i.e. ‘e’). The parser tries to expand non-terminal ‘X’ and checks its production from the left (X → oa). It does not match with the next input symbol. So the top-down parser backtracks to obtain the next production rule of X, (X → ea).

Bottom Up Parser :
Bottom-up parsing starts from the leaf nodes of a tree and works in upward direction till it reaches the root node. Here, we start from a sentence and then apply production rules in reverse manner in order to reach the start symbol. The image given below depicts the bottom-up parsers available.
For Example, input string : a + b * c

Probabilistic Parsing

Probabilistic parsers are used to solve the problem of disambiguation - as sentences tend to be syntactically ambiguous due to coordination and attachment ambiguity. A probabilistic parser offers a solution to the problem of efficiently representing ambiguities by: **computing the probability of each interpretation and choosing the most probable interpretation.**

Probabilistic parsers commonly use probabilistic context free grammar (PCFG) - each rule in PCFG is associated with a probability. A PCFG is used to estimate a number of useful probabilities for a sentence and its parse trees which is useful for **sentence disambiguation** and l**anguage modelling.**

A PFG is defined as : (**N,**Σ**,R,S**), where

N is a set of non-terminal symbols

Σ is a set of terminal symbols

R is a set of rules or productions, each of the form **A** $\rightarrow \beta[p]$ where A is a non terminal, β is a string of terminals and nonterminals and p is a probability.

S is the start symbol

Note:

1) All possible expansions of a non-terminal should have a summation equal to 1.

$$\sum_{\beta} P(A \rightarrow \beta) = 1$$

2) A PCFG is **consistent** if the sum of all probabilities of all sentences in a language equals to 1.

Probabilistic CKY Parsing of PCFGs

Goal: To produce the most likely parse tree T' for a given sentence S.
Assumption: PCFG is in Chomsky Normal Form.

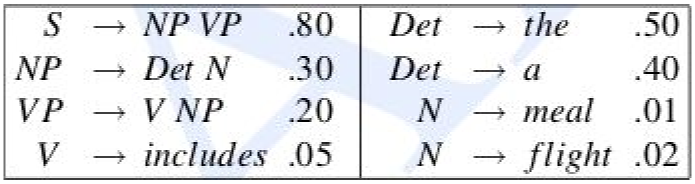
For CKY Algorithm each sentence is represented with indices between words.
⊕ Book ⊙ th ⊙ e flight ⊙ through ⊙ Houston ⊙

Now, for a sentence of length n and grammar containing V non-terminals, we use the upper triangle of a (n+1) × (n+1) × V matrix. For every non terminal A that belongs to V, the cell [i,j,A] in the matrix is a probability that A spans positions i through j of the input sentence.

Example: For a sentence,

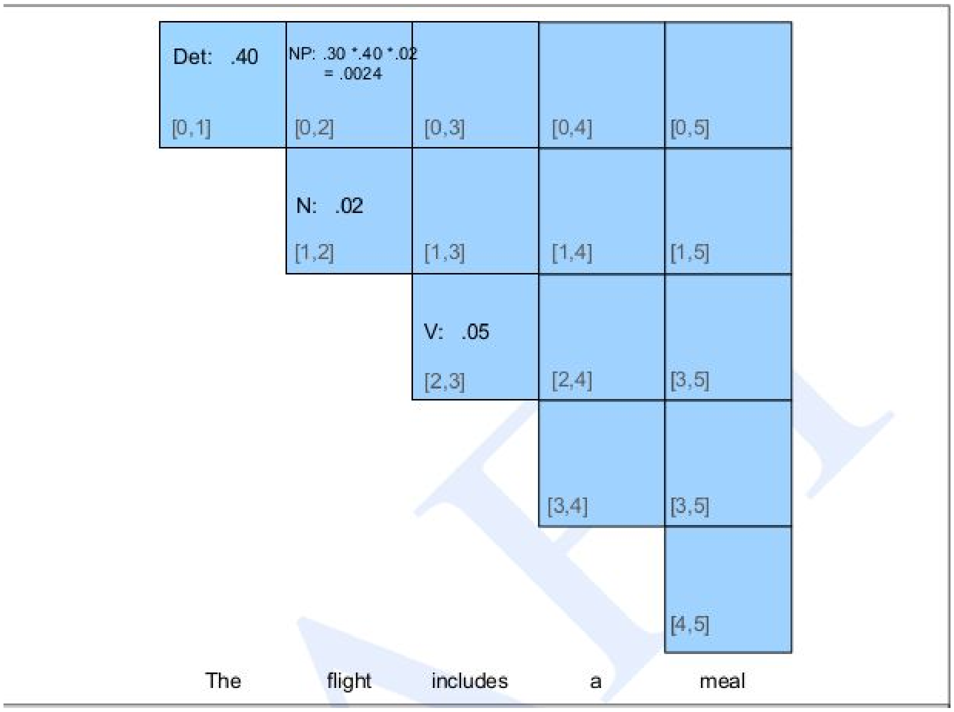
This flight includes a meal

the grammar is given as:



S	$\rightarrow$	$NP\ VP$	.80	Det	$\rightarrow$	the	.50
NP	$\rightarrow$	$Det\ N$	.30	Det	$\rightarrow$	a	.40
VP	$\rightarrow$	$V\ NP$	.20	N	$\rightarrow$	$meal$	.01
V	$\rightarrow$	$includes$	.05	N	$\rightarrow$	$flight$	.02

Then the matrix can be constructed as:



Det: .40 [0,1]	NP: .30 *.40 *.02 = .0024 [0,2]	[0,3]	[0,4]	[0,5]
	N: .02 [1,2]	[1,3]	[1,4]	[1,5]
		V: .05 [2,3]	[2,4]	[3,5]
			[3,4]	[3,5]
				[4,5]
The	flight	includes	a	meal

matrix covers only step 1

Application of Parsing : Noisy Text

Noisy Text:
The noise can be seen as all the differences between the surface form of a coded representation of the text and the intended, correct, or original text. It can be due to e.g. typographic errors or colloquialisms always present in natural language and usually lowers the data quality in a way that makes the text less accessible to automated processing by computers such as natural language processing. The noise can also get introduced through an extraction process (i.e. transcription, OCR) from media other than original electronic texts.

Language usage over computer mediated discourses, like chats, emails and SMS texts, significantly differs from the standard form of the language. An urge towards shorter message length facilitating faster typing and the need for semantic clarity, shape the structure of this text used in such discourses.

Robust and Scalable Parsing:
Robustness:
The ability of continuing the work even if something wrong with the productions or input string
i.e., fault tolerance behavior.
Robust Parsing:

- It overcomes the drawback of Top-Down Parsing algorithm
- The ability to find partial parses in an ungrammatical input
- One approach to the problem is to build robustness into the grammar itself.
- In the simplest case one could add top-level productions
- Alternative method for making parser robust is to modify the parser itself so as to accept arbitrary input and find all or a chosen subset of possible substring parses.

Scalable:

- Change the way of from parsing from bottom level to top level
- Instead to perform parsing on smaller dataset move out to the larger one, simply scale the initial phase of parsing.
- It can be applicable in top-down bottom-up charting algorithm method and works effectively.

Lexical Knowledge Networks

They are networks that represent semantic relations between concepts. This is often used as form of the knowledge representation and is similar to a graph (directed or undirected). A semantic network is used when one has knowledge that is best understood as a set of concepts that are related to one another.
Limitation is that they do not represent performance or meta-knowledge very well.

Some Lexical Network examples -

- WordNet[2]

Wordnet is a lexical knowledge base based on conceptual look up, it organizes lexical information in terms of word meaning rather than word form. And the main use in natural language processing for wordnet is the word sense disambiguation.
Word sense disambiguation is the determination of the correct sense of the word. Eg. we have two sentences “the crane ate the fish” versus “the crane was used to live the load”, and in the first case the crane ate the fish, crane is used in the sense of the bird, and the crane was used to lift the load here the crane was used in the sense of a machine.

- MindNet[3]

MindNet is a knowledge representation project that uses Microsoft’s broad-coverage parser to build semantic networks from dictionaries, encyclopedias, and free text. MindNets are produced by a fully automatic process that takes the input text, sentence-breaks it, parses each sentence to build a semantic dependency graph (Logical Form), aggregates these individual graphs into a single large graph, and then assigns probabilistic weights to subgraphs based on their frequency in the corpus as a whole. The project also encompasses a number of mechanisms for searching, sorting, and measuring the similarity of paths in a MindNet.

- VerbNet[4]

VerbNet (Kipper-Schuler 2006) is the largest on-line verb lexicon currently available for English. It is a hierarchical domain-independent, broad-coverage verb lexicon with mappings to other lexical resources such as WordNet (Miller, 1990; Fellbaum, 1998), Xtag (XTAG Research Group, 2001), and FrameNet (Baker et al., 1998). VerbNet is organized into verb classes through refinement and addition of subclasses to achieve syntactic and semantic coherence among members of a class.

Applications

Network models may provide new insight into the semantic content of large text collections. For example, semantic similarity networks may be used to identify shifts in topics and identify fake or forged articles. In addition, semantic similarity networks can be used for traditional information retrieval tasks such as document clustering.

Further Understanding

1. http://www.courses.com/indian-institute-of-technology-bombay/natural-language-processing

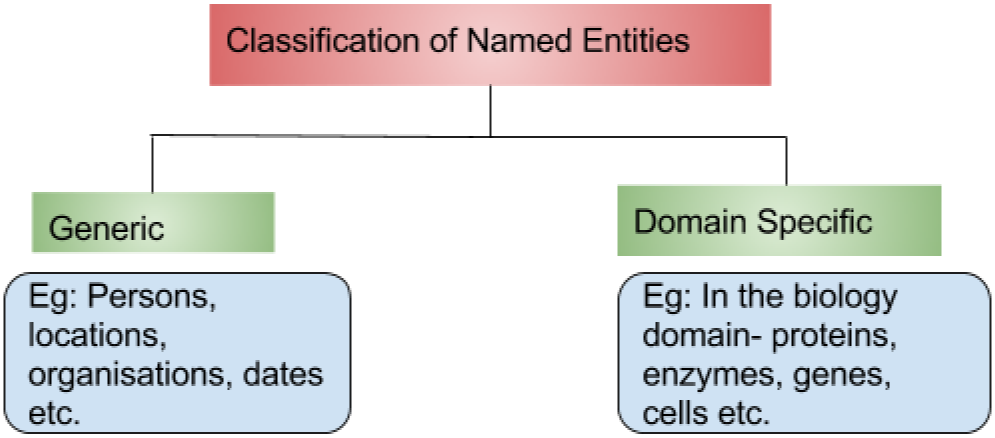
[2] http://nptel.ac.in/courses/106101007/29
[3] https://www.microsoft.com/en-us/research/project/mindnet/
[4] https://verbs.colorado.edu/verbnet/

Unit 2 - Multidisciplinary Natural Language Processing

Table of Contents

Named Entities[5]

Named Entities are basic semantic elements of text that carry a specific limited meaning.



Classification of Named Entities

Generic

Eg: Persons, locations, organisations, dates etc.

Domain Specific

Eg: In the biology domain- proteins, enzymes, genes, cells etc.

Challenges in Named Entity Recognition (NER):

1) Open nature of vocabulary
   Eg: Names of people across various regions of the world is hard to keep a track of.
2) Overlap between NE types
   Eg: Washington can refer to a person as well as a location
3) Multiword Named Entities cause a boundary identification problem
   Eg: “Bank of America” can mean “Bank of America” as an organisation entity or “America” as a location entity.
4) Abbreviations
   Eg: IBM and International Business Machines refer to the same NE.
5) Conjunctions
   Eg: “Boston Gas and Light Company” can refer to two entities or one.

[5] “Techniques for Named Entity Recognition: A Survey”, Girish Keshav Palshikar

Desirable Characteristics in an NE system:

1) Accuracy
2) Efficiency while tagging documents
3) Robustness against spelling and grammatical errors.
4) Corpus Independence
5) Language Independence
6) Ability to be extendible

Various Techniques for Named Entity Recognition:

1) Rule based Approach
2) Supervised Learning Approach
3) Unsupervised Learning Approach

Discourse Processing: Segmentation

Discourse Segmentation is an algorithm to to detect the structure of discourse in a document i.e. separating a document into a linear sequence of subtopics. This process finds applications in information retrieval, hypertext display and text summarization.

Discourse Segmentation can be categorized as :

i) Unsupervised Discourse Segmentation

Linear discourse segmentation is the task of segmenting text into multiple paragraph units that represent subtopics of the original text. One important class of unsupervised algorithms for linear discourse segmentation relies on a cohesion based approach. Cohesion is the use of certain linguistic devices to link or tie together textual units. Cohesion based approach uses the intuition that sentences or paragraphs in one subtopic are cohesive with each other but no with paragraphs in a neighbouring subtopics.

Eg: TextTiling algorithm uses cohesion-based approach (Steps: tokenization, lexical scores determination, and boundary identification)

ii) Supervised Discourse Segmentation

This segmentation is done for tasks which have boundary-labelled training data, eg. broadcast news. Any type of classifier can be used for this task ranging from svm, decision tree to hmm,crf. This method uses cohesion features for segmentation and has an additional presence of discourse markers or cue words.

EM Algorithm[6]

Stands for Expectation-Maximization algorithm. It is a way to find maximum-likelihood estimates for model parameters when your data is incomplete, has missing data points, or has unobserved (hidden) variables. It is an iterative way to approximate the maximum likelihood function.
It is an inequality involving convexity of a function..

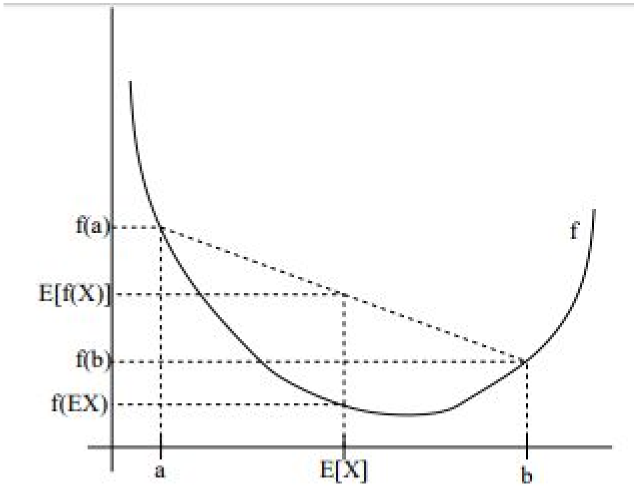
Prerequisite - Jensen’s Inequality
Convex Function - A function is convex on an interval I if the segment between any two points taken on its graph in lies above the graph. An example of a convex function is $f(x)=x^2$.

Mathematical Statement of Jensen’s Inequality -

[6] http://cs229.stanford.edu/notes/cs229-notes8.pdf

Let f be a convex function, and let X be a random variable.
Then: E[f(X)] ≥ f(EX).
Moreover, if f is strictly convex, then E[f(X)] = f(EX) holds true if and only if X = E[X] with probability 1 (i.e., if X is a constant).

Interpretation -



Here, f is a convex function shown by the solid line. Also, X is a random variable that has a 0.5 chance of taking the value a, and a 0.5 chance of taking the value b (indicated on the x-axis). Thus, the expected value of X is given by the midpoint between a and b. We also see the values f(a), f(b) and f(E[X]) indicated on the y-axis. Moreover, the value E[f(X)] is now the midpoint on the y-axis between f(a) and f(b). From our example, we see that because f is convex, it must be the case that E[f(X)] ≥ f(EX).

It is an Iterative approach to compute maximal likelihood when data is missing or hidden. The process happens in 2 steps:
1. E-step (Expected Step)
Missing data is estimated given the observed data and current estimate of model parameters.

2. M-step (Maximization Step)
Likelihood function is maximized under assumption that missing data is known from the E-step.

Likelihood function;
$L(\theta)=\ln P(X|\theta)$
Where,
X-random vector for parameterized family
θ-parameter
To maximize $L(\theta)$, M-step makes sure that $L(\theta)>L(\theta n)$

Example -[7]

[7] http://ai.stanford.edu/~chuongdo/papers/em_tutorial.pdf

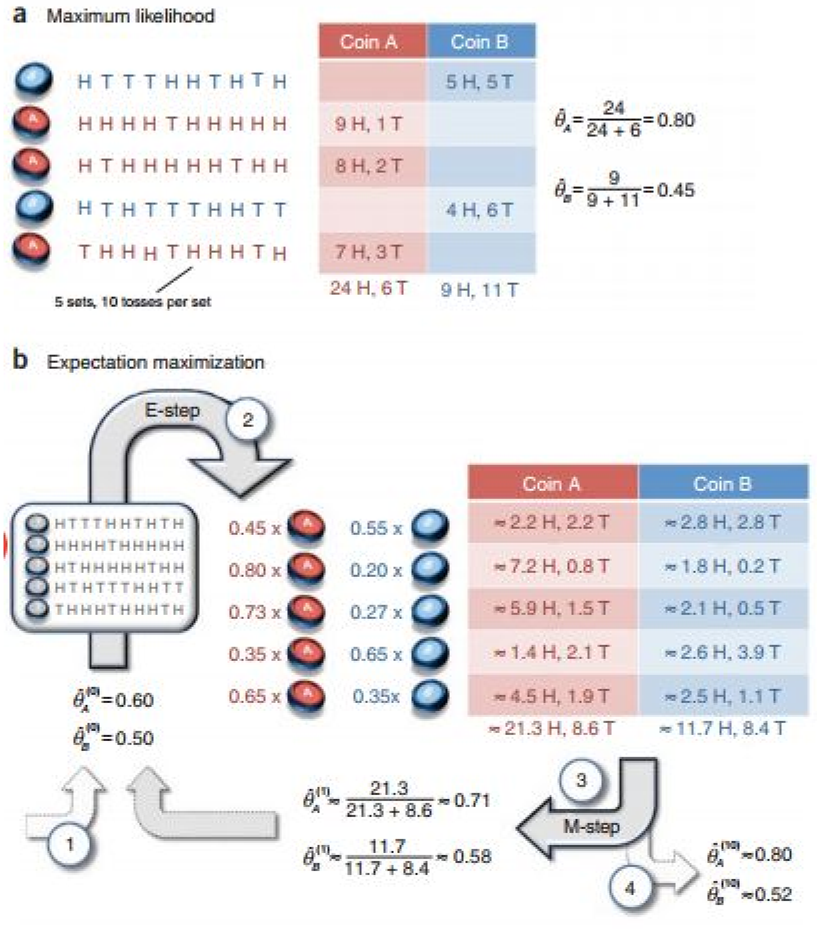


Figure 1 Parameter estimation for complete and incomplete data. **(a)** Maximum likelihood estimation. For each set of ten tosses, the maximum likelihood procedure accumulates the counts of heads and tails for coins A and B separately. These counts are then used to estimate the coin biases. **(b)** Expectation maximization. 1. EM starts with an initial guess of the parameters. 2. In the E-step, a probability distribution over possible completions is computed using the current parameters. The counts shown in the table are the expected numbers of heads and tails according to this distribution. 3. In the M-step, new parameters are determined using the current completions. 4. After several repetitions of the E-step and M-step, the algorithm converges.

Clustering

The method of identifying similar groups of data in a data set is called clustering. Entities in each group are comparatively more similar to entities of that group than those of the other groups. It is an unsupervised learning approach.

There are 2 types of clusterings broadly -

1. **Hard Clustering:**

Each data point either belongs to a cluster completely or not. For example, each customer is put into one group out of the 10 groups which are separated in terms of their purchasing habits.

2. **Soft Clustering**:

Instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, each customer is assigned a probability to be in either of 10 clusters of the retail stores.

Clustering can be classified as following based on whether they are nested or unnested:

1. Partitioning[8]

Division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset. Its an iterative clustering algorithm that aims to find local maxima in each iteration.

Steps:

1.1 Choose k.

1.2 Randomly assign each data point to a cluster.

[8] https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/

1.3 Compute cluster centroids.
1.4 Re-assign each point to the closest cluster centroid.
1.5 Re-compute cluster centroids.
1.6 Repeat steps 4 and 5 until no improvements are possible.

Advantages:
- Simple.
- Efficient.
- Linear Algorithm.

Disadvantages:
- Cannot deal with outliers.

Complexity: O(KIN), where, K-k, number of clusters, I-number of iterations and N-number of objects to cluster.

2. Hierarchical
Set of nested clusters that are organized as a tree.

Features:
2.1 No value of k needed.
2.2 The generated tree may correspond to a meaningful taxonomy.
2.3 Only a distance or "proximity" matrix is needed to compute the hierarchical clustering. Clusters are merged using factors of "proximity matrix" which can be calculated as MIN (Single Link), MAX (Complete Link), AVG (Entire Group) Proximities.

Types:
2.1 Agglomerative ("Bottom-up")[9]
Start with the points as individual clusters and, at each step, merge the closest pair of clusters.
2.2 Divisive ("Top-down")
Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain.

Log-Linear Model[6]

We have some input domain X , and a finite label set Y.
Aim is to provide a conditional probability P(y | x) for any $x \in X$ and $y \in Y$.
A feature is a function $f : X \times Y \rightarrow R$ (Often binary features or indicator functions $f_k : X \times Y \rightarrow \{0, 1\}$). Say we have m features f_k for k = 1 . . . m $\Rightarrow$ A feature vector $f(x, y) \in R$ m for any $x \in X$ and $y \in Y$.
We also have a parameter vector $v \in R$ m.
We define $P(y \mid x; v) = e^{v \cdot f(x,y)} / \Sigma_{y' \in Y}\, e^{v \cdot f(x,y')}$

Why the term log-linear?

$\text{Log } P(y \mid x; v) = v \cdot f(x,y) - \log \Sigma_{y' \in Y}\, e^{v \cdot f(x,y')}$

Where, $v \cdot f(x,y)$ is the linear term and $\log \Sigma_{y' \in Y}\, e^{v \cdot f(x,y')}$ is the normalization term.

Machine Translation[10]

In the modern world, there is an increased need for language translations owing to the fact that language is an effective medium of communication. Machine translation (MT), a subfield under Artificial Intelligence, is the application of computers to the task of translating texts from one natural (human) language to another. To process any translation, human or automated, the meaning of a text in the original (source) language must be

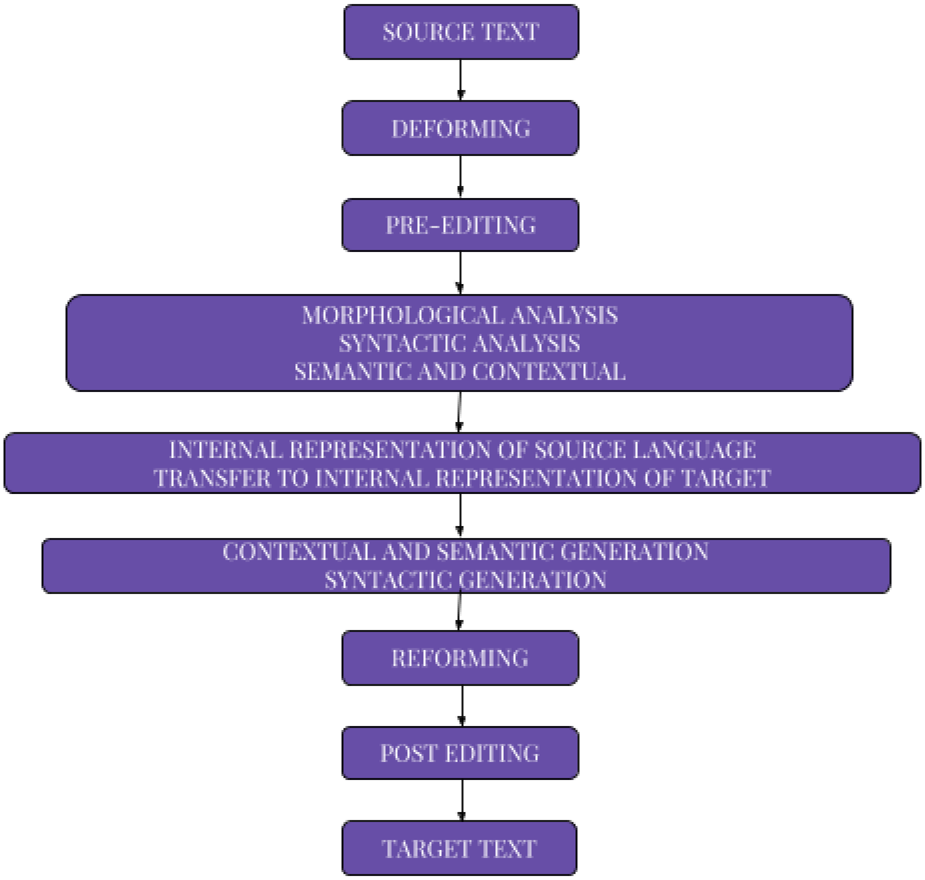
[9] https://www3.nd.edu/~rjohns15/cse40647.sp14/www/content/lectures/13%20-%20Hierarchical%20Clustering.pdf
[10] Machine Translation Approaches: Issues and Challenges, M. D. Okpor

fully restored in the target language, i.e., the translation. While on the surface, this seems straightforward, it is far more complex.

The human translation process, for instance, may be described as:

- Decoding the meaning of the source text
- Re-encoding this meaning in the target language

Behind this ostensibly simple procedure lies a complex cognitive operation. To decode the meaning of the source text in its entirety, the translator must interpret and analyse all the features of the text, a process that requires in-depth knowledge of the grammar, semantics, syntax, idioms, etc., of the source language, as well as the culture of its speakers. The translator needs the same in-depth knowledge to re-encode the meaning in the target language.



A machine translation (MT) system first analyses the source language input and creates an internal representation. This representation is manipulated and transferred to a form suitable for the target language. Then at last output is generated in the target language. MT systems can be classified according to their core methodology. Under this classification, two main paradigms can be found: **the rule-based approach** and **the corpus-based approach**. In the rule-based approach, human experts specify a set of rules to describe the translation process, so that an enormous amount of input from human experts is required. On the other hand, under the corpus-based approach the knowledge is automatically extracted by analysing translation examples from a parallel corpus built by human experts. Combining the features of the two major classifications of MT systems gave birth to the **Hybrid Machine Translation Approach**.

Language Modelling[11]

Language models were originally developed for the problem of speech recognition; they still play a central role in modern speech recognition systems. They are also widely used in other NLP applications.

w_i is the i'th word in a document.

Estimate a distribution $P(w_i |w_1, w_2, . . . w_{i-1})$ given previous "history" $w_1, . . . , w_{i-1}$.

[11] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-864-advanced-natural-language-processing-fall-2005/lecture-notes/lec09.pdf

E.g., $w_1, \ldots, w_{i-1}$ = Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical
y is an “outcome” $w_{i.}$
Aim is to provide a conditional probability P (y | x) for “decision” y given “history” x.
A feature is a function f (x, y) ≤ R (Often binary features or indicator functions f (x, y) ε { 0, 1 }).
Say we have m features ϕ_k for k = 1 . . . m.
A feature vector ϕ (x, y) ≤ R^m for any x,y.

Example of feature vectors are:

$$\phi_1(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_2(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model} \text{ and } w_{i-1} = \texttt{statistical} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_3(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-2} = \texttt{any},\ w_{i-1} = \texttt{statistical} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_4(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-2} = \texttt{any} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_5(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-1} \text{ is an adjective} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_6(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-1} \text{ ends in “ical”} \\ 0 & \text{otherwise} \end{cases}$$

Each possible y gets a different score:

$$\sum_k \mathbf{W_k}\phi_k(x, model) = 5.6 \qquad \sum_k \mathbf{W_k}\phi_k(x, the) = -3.2$$

$$\sum_k \mathbf{W_k}\phi_k(x, is) = 1.5 \qquad \sum_k \mathbf{W_k}\phi_k(x, of) = 1.3$$

$$\sum_k \mathbf{W_k}\phi_k(x, models) = 4.5 \qquad \ldots$$

So, at the end the estimate of likelihood of different words in a sentence or phrases in a paragraph are found.

Applications:
- Speech Recognition
- Machine Translation

Markov Model[12]

Markov Models for Fixed-length Sequences : Consider a sequence of random variables, X1, X2, . . . , Xn. Each random variable can take any value in a finite set *V*. For now we will assume that the length of the sequence, n, is some fixed number (e.g., n = 100). Our goal is as follows: we would like to model the probability of any sequence x1...xn, where n ≥ 1 and xi ∈ V for i = 1...n, that is, to model the joint probability :

$$P(X1 = x1,X2 = x2,...,Xn = xn)$$

There are |*V* |n possible sequences of the form x1 . . . xn : so clearly, it is not feasible for reasonable values of |*V*| and n to simply list all |*V*|n probabilities. We would like to build a much more compact model.
In a first-order Markov process, we make the following assumption, which considerably simplifies the model:

$$P(X1 = x1,X2 = x2,...Xn = xn)$$

$$= P(X1 =x1)\ \pi\ \overset{n}{\underset{i=2\ n}{P}}(Xi =xi|X1 =x1,...,Xi-1 =xi-1) \quad (1.1)$$

$$= P(X1 = x1)\ \pi_{i=2}^{n}\ P(Xi = xi|Xi-1 = xi-1) \quad (1.2)$$

The first step, in Eq. 1.1, is exact: by the chain rule of probabilities, *any* distribution P(X1 = x1 ...Xn = xn) can be written in this form. So we have made no assumptions in this step of the derivation. However, the second step, in Eq. 1.2, is not necessarily exact: we have made the assumption that for any i ∈ {2 . . . n}, for any x1 . . . xi,

$$P(Xi = xi|X1 = x1 ...Xi-1 = xi-1) = P(Xi = xi|Xi-1 = xi-1)$$

This is a (first-order) *Markov assumption*. We have assumed that the identity of the i'th word in the sequence depends only on the identity of the previous word, xi−1. More formally, we have assumed that the value of Xi is conditionally independent of X1 . . . Xi−2, given the value for Xi−1. In a second-order Markov process, which will form the basis of trigram language models, we make a slightly weaker assumption, namely that each word depends on the previous *two* words in the sequence:

$$P(Xi = xi|X1 = x1,...,Xi-1 = xi-1) = P\ (Xi = xi|Xi-2 = xi-2,\ Xi-1 = xi-1)$$

It follows that the probability of an entire sequence is written as :

$$P(X1 = x1,X2 = x2,...Xn \overset{n}{\underset{i=1}{=}} xn) = \ \pi\ \ P\ (Xi = xi|Xi-2 = xi-2,\ Xi-1 = xi-1) \quad (1.3)$$

For convenience, we will assume that x0 = x−1 = * in this definition, where * is a special "start" symbol in the sentence.

Markov Sequences for Variable-length Sentences : In the previous section, we assumed that the length of the sequence, n, was fixed. In many applications, however, the length n can itself vary. Thus n is itself a random variable. There are various ways of modeling this variability in length: in this section we describe the most common approach for language modeling.
The approach is simple: we will assume that the nth word in the sequence, Xn, is always equal to a special symbol, the STOP symbol. This symbol can only appear at the end of a sequence.

The process that generates sentences would be as follows:
1. Initialize i=1,and x0 =x−1 =*
2. Generate xi from the distribution : P (Xi = xi|Xi−2 = xi−2, Xi−1 = xi−1)
3. If xi = STOP then return the sequence x1 ...xi. Otherwise, set i = i + 1 and return to step 2.

[12] http://www.cs.columbia.edu/~mcollins/lm-spring2013.pdf

Thus we now have a model that generates sequences that vary in length.

Hidden Markov Model[13]

A Markov chain is useful when we need to compute a probability for a sequence of events that we can observe in the world. In many cases, however, the events we are interested in may not be directly observable in the world. For example, in part-of- speech tagging (Ch. 5 of textbook) we didn’t observe part of speech tags in the world; we saw words, and had to infer the correct tags from the word sequence. We call the part-of-speech tags hidden because they are not observed. A **Hidden Markov Model** (HMM) allows us to talk about both *observed* events (like words that we see in the input) and *hidden* events (like part-of-speech tags) that we think of as causal factors in our probabilistic model.

Let’s begin with a formal definition of a Hidden Markov Model, focusing on how HMM it differs from a Markov chain. An HMM is specified by the following components:

$Q = q1q2 \ldots qN$	a set of N states
$A = a11a12 \ldots an1 \ldots ann$	a transition probability matrix A, each $ai\,j$ representing the probability of moving from state i tostate$j$, s.t.$\Sigma^{n}_{j} =1aij=1$ $\forall i$
$O = o1o2 \ldots oT$	a sequence of T observations, each one drawn from a vocabulary $V = v1,v2,\ldots,vV$.
$B = bi(ot\ )$	a sequence of observation likelihoods:, also called emission probabilities, each expressing the probability of an observation ot being generated from a state i.
$q0,qF$	a special start state and end (final) state which are not associated with observations, together with transition probabilities $a01a02..a0n$ out of the start state and $a1F\ a2F \ldots anF$ into the end state.

To exemplify this model, we’ll use a task conceived of by Jason Eisner (2002). Imagine that you are a climatologist in the year 2799 studying the history of global warming. You cannot find any records of the weather in Baltimore, Maryland, for the summer of 2007, but you do find Jason Eisner’s diary, which lists how many ice creams Jason ate every day that summer. Our goal is to use these observations to estimate the temperature every day. We’ll simplify this weather task by assuming there are only two kinds of days: cold (C) and hot (H). So the Eisner task is as follows:

Given a sequence of observations O, each observation an integer corresponding to the number of ice creams eaten on a given day, figure out the correct ‘hidden’ sequence Q of weather states (H or C) which caused Jason to eat the ice cream.

A first-order Hidden Markov Model instantiates two simplifying assumptions.

First, as with a first-order Markov chain, the probability of a particular state is dependent only on the previous state: Markov Assumption: $P(qi|q1\ldots qi-1) = P(qi|qi-1)$

[13] http://stp.lingfil.uu.se/~santinim/ml/2014/JurafskyMartinSpeechAndLanguageProcessing2ed_draft%202007.pdf

Second, the probability of an output observation o_i is dependent only on the state that produced the observation *qi*, and not on any other states or any other observations:

Output Independence Assumption: $P(oi|q1...qi,...,qT,o1,...,oi,...,oT)=P(oi|qi)$

Fig. 6.3 shows a sample HMM for the ice cream task. The two hidden states (H and C) correspond to hot and cold weather, while the observations (drawn from the alphabet *O* = {1,2,3}) correspond to the number of ice creams eaten by Jason on a given day.

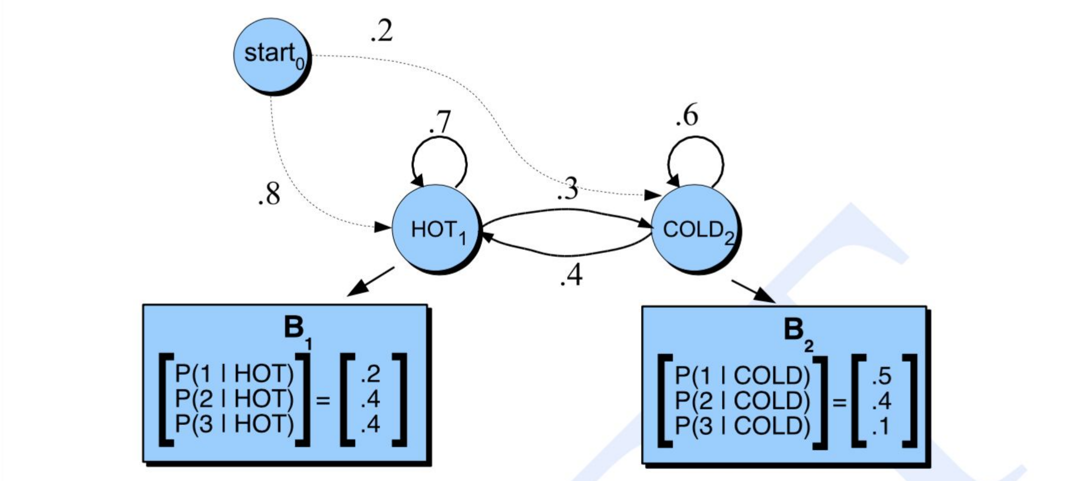


Figure 6.3 A Hidden Markov Model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables). For this example we are not using an end-state, instead allowing both states 1 and 2 to be a final (accepting) state.

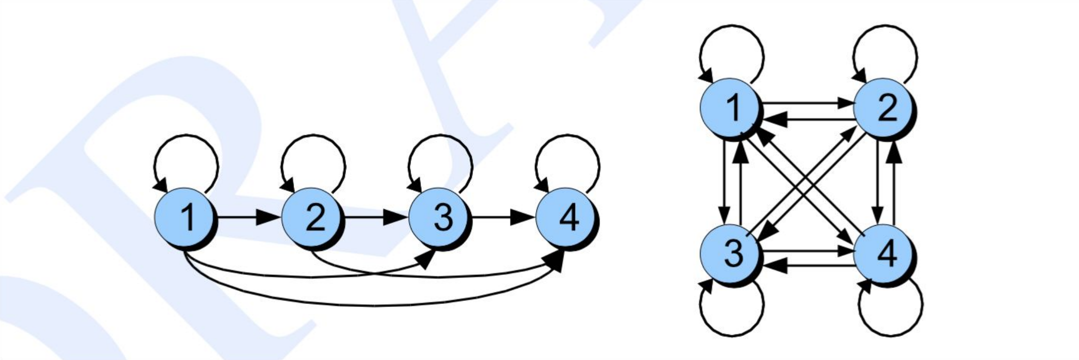


Figure 6.4 Two 4-state Hidden Markov Models; a left-to-right (Bakis) HMM on the left, and a fully-connected (ergodic) HMM on the right. In the Bakis model, all transitions not shown have zero probability.

Notice that in the HMM in Fig. 6.3, there is a (non-zero) probability of transitioning between any two states. Such an HMM is called a fully-connected or ergodic HMM. Sometimes, however, we have HMMs in which many of the transitions between states have zero probability. For example, in left-to-right (also called Bakis) HMMs, the state transitions proceed from left to right, as shown in Fig. 6.4. In a Bakis HMM, there are no transitions going from a higher-numbered state to a lower-numbered state (or, more accurately, any transitions from a higher-numbered state to a lower-numbered state have zero probability). Bakis HMMs are generally used to model temporal processes like speech.

Hidden Markov Models should be characterized by three fundamental problems:
Problem 1 (Computing Likelihood): Given an HMM $\lambda = (A,B)$ and an observation sequence *O*, determine the likelihood $P(O|\lambda)$.
Problem2(Decoding): Given an observation sequence *O* and an HMM $\lambda = (A, B)$, discover the best hidden state sequence *Q*.

Problem 3 (Learning): Given an observation sequence *O* and the set of states in the HMM, learn the HMM parameters *A* and *B*.

Stochastic Tagging[14]

Part-of-speech tagging (or just tagging for short) is the process of assigning a part- of-speech or other syntactic class marker to each word in a corpus. This section describes a particular stochastic tagging algorithm generally known as the Hidden Markov Model or HMM tagger. Use of a Hidden Markov Model to do part-of-speech-tagging, as we will define it, is a special case of Bayesian inference, a paradigm that has been known since the work of Bayes (1763).

In a classification task, we are given some observation(s) and our job is to determine which of a set of classes it belongs to. Part-of-speech tagging is generally treated as a sequence classification task. So here the observation is a sequence of words (let's say a sentence), and it is our job to assign them a sequence of part-of-speech tags. For example, say we are given a sentence like

Secretariat is expected to race tomorrow.

What is the best sequence of tags which corresponds to this sequence of words? The Bayesian interpretation of this task starts by considering all possible sequences of classes—in this case, all possible sequences of tags. Out of this universe of tag sequences, we want to choose the tag sequence which is most probable given the observation sequence of *n* words *wn*1. In other words, we want, out of all sequences of *n, tn* the single tag sequence such that *P*(*tn*|*wn*) is highest.

So, HMM taggers choose the tag sequence that maximizes P(tag|previous tag)*P(word|tag)

Eg : Secretariat is expected **to** race tomorrow.

What was the reason for **the** race.

We have to find tag for race. We know that tag for to is 'TO' and tag for the is 'DET'.

We find P(NN|TO)*P(race|NN) = x ---(1)

and P(VB|DET)*P(race|VB) = y.

The tag with a higher value is assigned. Here (1) shows the likelyness of NN (or VB) when the previous tag is TO and the probability that the word race being tagged as VB or NN. The probability of a word having a particular tag is taken from various dictionaries.

[14] TextBook

Unit 3 - Multidisciplinary Natural Language Processing

Table of Contents

Markov Model[15]

Markov Models for Fixed-length Sequences : Consider a sequence of random variables, X1, X2, . . . , Xn. Each random variable can take any value in a finite set *V*. For now we will assume that the length of the sequence, n, is some fixed number (e.g., n = 100). Our goal is as follows: we would like to model the probability of any sequence x1...xn, where $n \geq 1$ and $xi \in V$ for i = 1...n, that is, to model the joint probability :

$$P(X1 = x1,X2 = x2,...,Xn = xn)$$

There are $|V|n$ possible sequences of the form x1 . . . xn : so clearly, it is not feasible for reasonable values of $|V|$ and n to simply list all $|V|n$ probabilities. We would like to build a much more compact model.
In a first-order Markov process, we make the following assumption, which considerably simplifies the model:

$$P(X1 = x1,X2 = x2,...Xn = xn)$$

$$= P(X1 = x1)\ \pi\ \overset{n}{\underset{i=2\ n}{P}}(Xi = xi|X1 = x1,...,Xi-1 = xi-1) \quad (1.1)$$

$$= P(X1 = x1)\ \pi_{i=2}^{n}\ P(Xi = xi|Xi-1 = xi-1) \quad (1.2)$$

The first step, in Eq. 1.1, is exact: by the chain rule of probabilities, *any* distribution P(X1 = x1 ...Xn = xn) can be written in this form. So we have made no assumptions in this step of the derivation. However, the second step, in Eq. 1.2, is not necessarily exact: we have made the assumption that for any $i \in \{2 . . . n\}$, for any x1 . . . xi,

$$P(Xi = xi|X1 = x1 ...Xi-1 = xi-1) = P(Xi = xi|Xi-1 = xi-1)$$

This is a (first-order) *Markov assumption*. We have assumed that the identity of the i'th word in the sequence depends only on the identity of the previous word, xi−1. More formally, we have assumed that the value of Xi is conditionally independent of X1 . . . Xi−2, given the value for Xi−1. In a second-order Markov process, which will form the basis of trigram language models, we make a slightly weaker assumption, namely that each word depends on the previous *two* words in the sequence:

[15] http://www.cs.columbia.edu/~mcollins/lm-spring2013.pdf

$P(X_i = x_i|X_1 = x_1,\ldots,X_{i-1} = x_{i-1}) = P\ (X_i = x_i|X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$

It follows that the probability of an entire sequence is written as :

$$P(X_1 = x_1,X_2 = x_2,\ldots X_n \overset{n}{\underset{i=1}{=}} x_n) = \pi\ \ P\ (X_i = x_i|X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1}) \quad (1.3)$$

For convenience, we will assume that $x_0 = x_{-1} = *$ in this definition, where * is a special "start" symbol in the sentence.

Markov Sequences for Variable-length Sentences : In the previous section, we assumed that the length of the sequence, n, was fixed. In many applications, however, the length n can itself vary. Thus n is itself a random variable. There are various ways of modeling this variability in length: in this section we describe the most common approach for language modeling.
The approach is simple: we will assume that the n^{th} word in the sequence, X_n, is always equal to a special symbol, the STOP symbol. This symbol can only appear at the end of a sequence.

The process that generates sentences would be as follows:
1. Initialize i=1,and $x_0 = x_{-1} = *$
2. Generate x_i from the distribution : $P\ (X_i = x_i|X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$
3. If x_i = STOP then return the sequence $x_1 \ldots x_i$. Otherwise, set i = i + 1 and return to step 2.

Thus we now have a model that generates sequences that vary in length.

Hidden Markov Model[16]

A Markov chain is useful when we need to compute a probability for a sequence of events that we can observe in the world. In many cases, however, the events we are interested in may not be directly observable in the world. For example, in part-of- speech tagging (Ch. 5 of textbook) we didn't observe part of speech tags in the world; we saw words, and had to infer the correct tags from the word sequence. We call the part-of-speech tags hidden because they are not observed. A **Hidden Markov Model** (HMM) allows us to talk about both *observed* events (like words that we see in the input) and *hidden* events (like part-of-speech tags) that we think of as causal factors in our probabilistic model.

Let's begin with a formal definition of a Hidden Markov Model, focusing on how HMM it differs from a Markov chain. An HMM is specified by the following components:

$Q = q_1q_2 \ldots q_N$	a set of *N* states
$A = a_{11}a_{12} \ldots a_{n1} \ldots a_{nn}$	a transition probability matrix *A*, each a_{ij} rep-resenting the probability of moving from state *i* tostate*j*, s.t.$\Sigma^{n}_{j}=1a_{ij}=1\ \forall i$
$O = o_1o_2 \ldots o_T$	a sequence of *T* observations, each one drawn from a vocabulary $V = v_1,v_2,\ldots,v_V$.
$B = b_i(o_t)$	a sequence of observation likelihoods:, also called emission probabilities, each expressing the probability of an observation o_t being generated from a state *i*.

[16] http://stp.lingfil.uu.se/~santinim/ml/2014/JurafskyMartinSpeechAndLanguageProcessing2ed_draft%202007.pdf

q0,qF	a special start state and end (final) state which are not associated with observations, together with transition probabilities $a01a02..a0n$ out of the start state and $a1F$ $a2F$ ...anF into the end state.

To exemplify this model, we'll use a task conceived of by Jason Eisner (2002). Imagine that you are a climatologist in the year 2799 studying the history of global warming. You cannot find any records of the weather in Baltimore, Maryland, for the summer of 2007, but you do find Jason Eisner's diary, which lists how many ice creams Jason ate every day that summer. Our goal is to use these observations to estimate the temperature every day. We'll simplify this weather task by assuming there are only two kinds of days: cold (C) and hot (H). So the Eisner task is as follows:

Given a sequence of observations *O*, each observation an integer corresponding to the number of ice creams eaten on a given day, figure out the correct 'hidden' sequence *Q* of weather states (H or C) which caused Jason to eat the ice cream.

A first-order Hidden Markov Model instantiates two simplifying assumptions.

First, as with a first-order Markov chain, the probability of a particular state is dependent only on the previous state: Markov Assumption: $P(qi|q1...qi-1) = P(qi|qi-1)$

Second, the probability of an output observation o_i is dependent only on the state that produced the observation *qi*, and not on any other states or any other observations:

Output Independence Assumption: $P(oi|q1...qi,...,qT,o1,...,oi,...,oT)=P(oi|qi)$

Fig. 6.3 shows a sample HMM for the ice cream task. The two hidden states (H and C) correspond to hot and cold weather, while the observations (drawn from the alphabet *O* = {1,2,3}) correspond to the number of ice creams eaten by Jason on a given day.

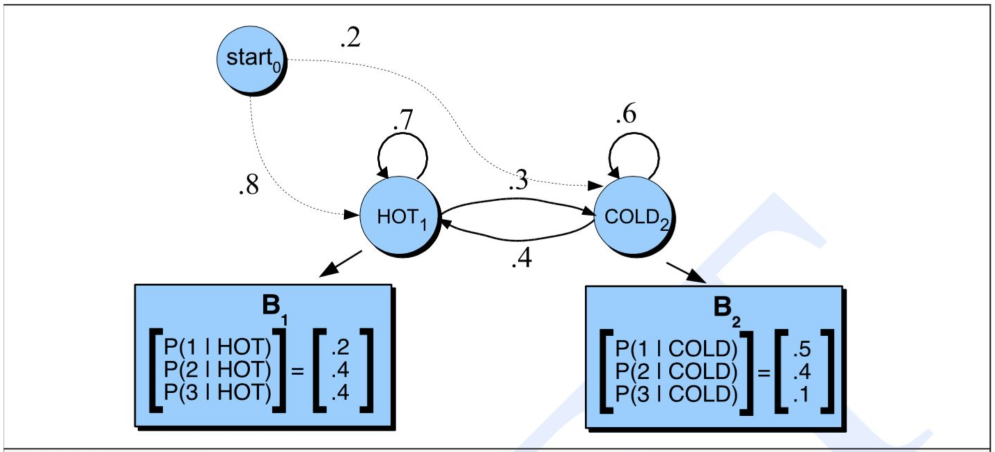


Figure 6.3 A Hidden Markov Model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables). For this example we are not using an end-state, instead allowing both states 1 and 2 to be a final (accepting) state.

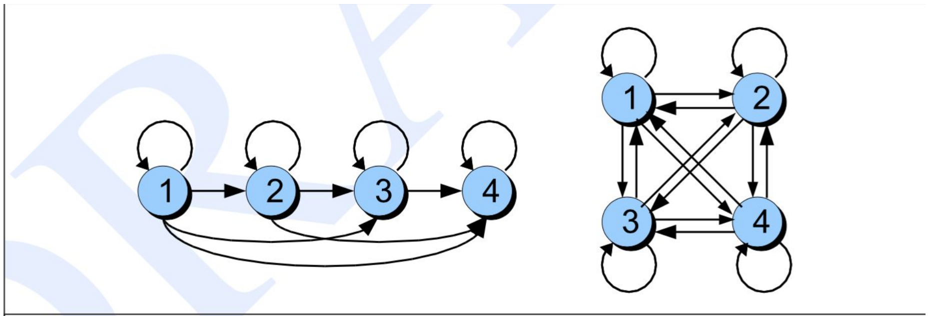


Figure 6.4 Two 4-state Hidden Markov Models; a left-to-right (Bakis) HMM on the left, and a fully-connected (ergodic) HMM on the right. In the Bakis model, all transitions not shown have zero probability.

Notice that in the HMM in Fig. 6.3, there is a (non-zero) probability of transitioning between any two states. Such an HMM is called a fully-connected or ergodic HMM. Sometimes, however, we have HMMs in which many of the transitions between states have zero probability. For example, in left-to-right (also called Bakis) HMMs, the state transitions proceed from left to right, as shown in Fig. 6.4. In a Bakis HMM, there are no transitions going from a higher-numbered state to a lower-numbered state (or, more accurately, any transitions from a higher-numbered state to a lower-numbered state have zero probability). Bakis HMMs are generally used to model temporal processes like speech.

Hidden Markov Models should be characterized by three fundamental problems:
Problem 1 (Computing Likelihood): Given an HMM $\lambda = (A,B)$ and an observation sequence *O*, determine the likelihood $P(O|\lambda)$.
Problem2(Decoding): Given an observation sequence *O* and an HMM $\lambda = (A, B)$, discover the best hidden state sequence *Q*.
Problem 3 (Learning): Given an observation sequence *O* and the set of states in the HMM, learn the HMM parameters *A* and *B*.

Viterbi Algorithm[17]

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events. For example, in speech-to-text (speech recognition), the acoustic signal is treated as the observed sequence of events, and a string of text is considered to be the "hidden cause" of the acoustic signal. The Viterbi algorithm finds the most likely string of text given the acoustic signal.

Formal representation of Viterbi Algorithm:

Input

- Observation space O = $\{o_1,o_2,...o_n\}$
- State Space S = $\{s_1,s_2,...,s_k\}$
- Initial probabilities $\Pi = (\Pi_1, \Pi_2, .. \Pi_k)$ where Π_i stores the probability of the first hidden state (x_1) will be equal to s_i
- Sequence of observations Y = $(y_1,y_2,...,y_T)$ such that y_t == i if observation at time t is o_i.
- Transition matrix A of size KxK where A_{ij} stores the probability of transitioning from state s_i to s_j.
- Emission probabilty matrix B f size KxN such that B_{ij} stores the probability of observing o_j from state s_i.

Output

- The most likely hidden state sequence X =$(x_1,x_2,...,x_N)$

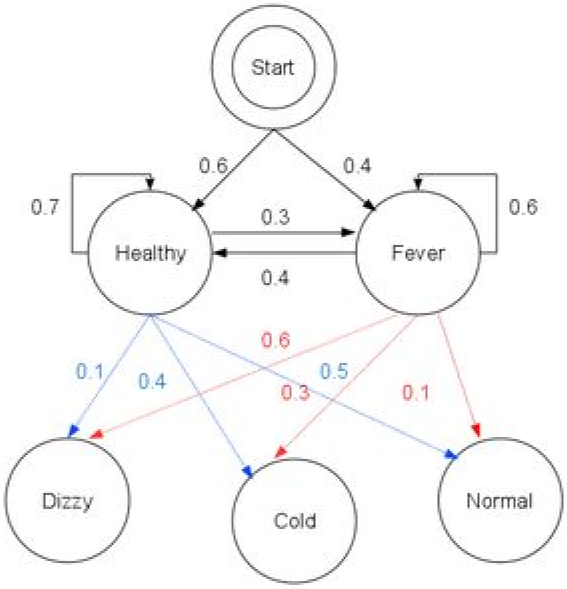
[17] https://en.wikipedia.org/wiki/Viterbi_algorithm

Example:

Consider a village where all villagers are either healthy or have a fever and only the village doctor can determine whether each has a fever. The doctor diagnoses fever by asking patients how they feel. The villagers may only answer that they feel normal, dizzy, or cold.

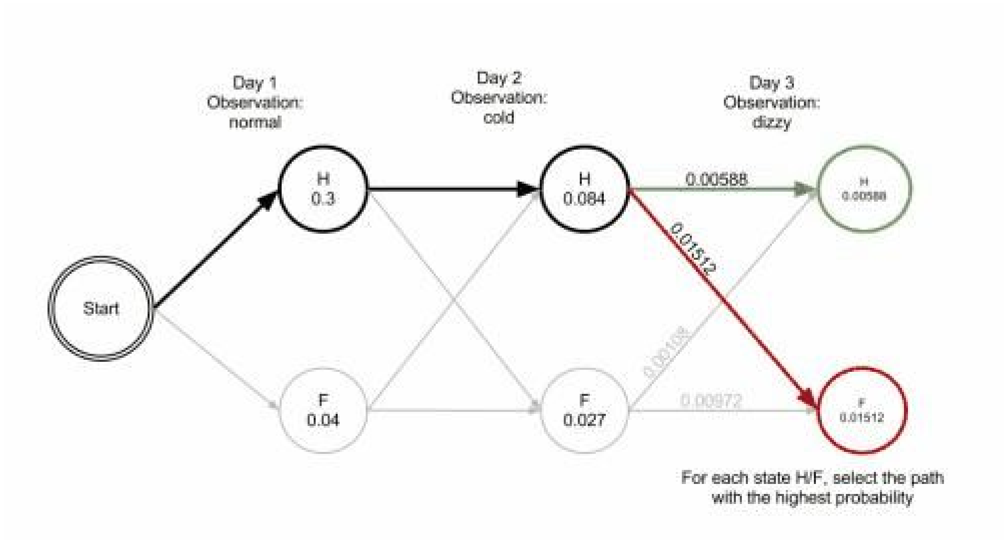
The doctor believes that the health condition of his patients operate as a discrete Markov chain. There are two states, "Healthy" and "Fever", but the doctor cannot observe them directly; they are *hidden* from him. On each day, there is a certain chance that the patient will tell the doctor he/she is "normal", "cold", or "dizzy", depending on their health condition.

The *observations (O)* (normal, cold, dizzy) along with a *hidden* state (X) (healthy, fever) form a hidden Markov model (HMM). Start Probability Π represents the doctor's belief about which state the HMM is in when the patient first visits (all he knows is that the patient tends to be healthy). The transition_probability represents the change of the health condition in the underlying Markov chain. The emission_probability represents how likely the patient is to feel on each day based on the hidden state.



The patient visits three days in a row and the doctor discovers that on the first day she feels normal, on the second day she feels cold, on the third day she feels dizzy. The doctor has a question: what is the most likely sequence of health conditions of the patient that would explain these observations? This is answered by the Viterbi algorithm.

The operation of Viterbi's algorithm can be visualized by means of a trellis diagram. The Viterbi path is essentially the shortest path through this trellis. The trellis for the clinic example is shown below; the corresponding Viterbi path is in bold:



For each state H/F, select the path with the highest probability

After Day 3, the most likely path is ['Healthy', 'Healthy', 'Fever']

Forward and Backward Probability[18]

These algorithms cater to the third problem of HMM i.e. Learning.
Learning: Given an observation sequence *O* and the set of possible states in the HMM, learn the HMM parameters *A* and *B*.
The input to such a learning algorithm would be an unlabeled sequence of observations *O* and a vocabulary of potential hidden states *Q*. Thus for the ice cream task, we would start with a sequence of observations *O* = {1, 3, 2, ..., }, and the set of hidden states *H* and *C*.
For the part-of-speech tagging task we would start with a sequence of observations *O* = {*w*1,*w*2,*w*3 ...} and a set of hidden states *NN, NNS, VBD, IN,...* and so on.
The standard algorithm for HMM training is the forward-backward or Baum- Welch algorithm (Baum, 1972), a special case of the Expectation-Maximization or EM algorithm (Dempster et al., 1977). The algorithm will let us train both the transition probabilities *A* and the emission probabilities *B* of the HMM.
In order to understand the algorithm, we need to define a useful probability related to the forward probability, called the backward probability.
The backward probability β is the probability of seeing the observations from time *t* + 1 to the end, given that we are in state *i* at time *t* (and of course given the automaton λ):
$\beta t(i) = P(ot+1,ot+2 ...oT \mid qt = i, \lambda)$

It is computed inductively in a similar manner to the forward algorithm.

1. Initialization:

$$\beta T(i) = ai,F,\ 1 \le i \le N$$

2. Recursion (again since states 0 and *qF* are non-emitting):

$$\beta t(i)= {}^{N}\sum_{j=1} (aij\ bj(ot+1)\,\beta\, t+1(j),\ 1 \le i \le N, 1 \le t < T)$$

3. Termination:

$$N\ P(O \mid \lambda)= \alpha\, T(qF)= \beta\, 1(0)= \mathrm{N}\sum j=1\ (a0j\ bj(o1)\,\beta\, 1(j))$$

Figure below illustrates the backward induction step.

[18] http://stp.lingfil.uu.se/~santinim/ml/2014/JurafskyMartinSpeechAndLanguageProcessing2ed_draft%202007.pdf

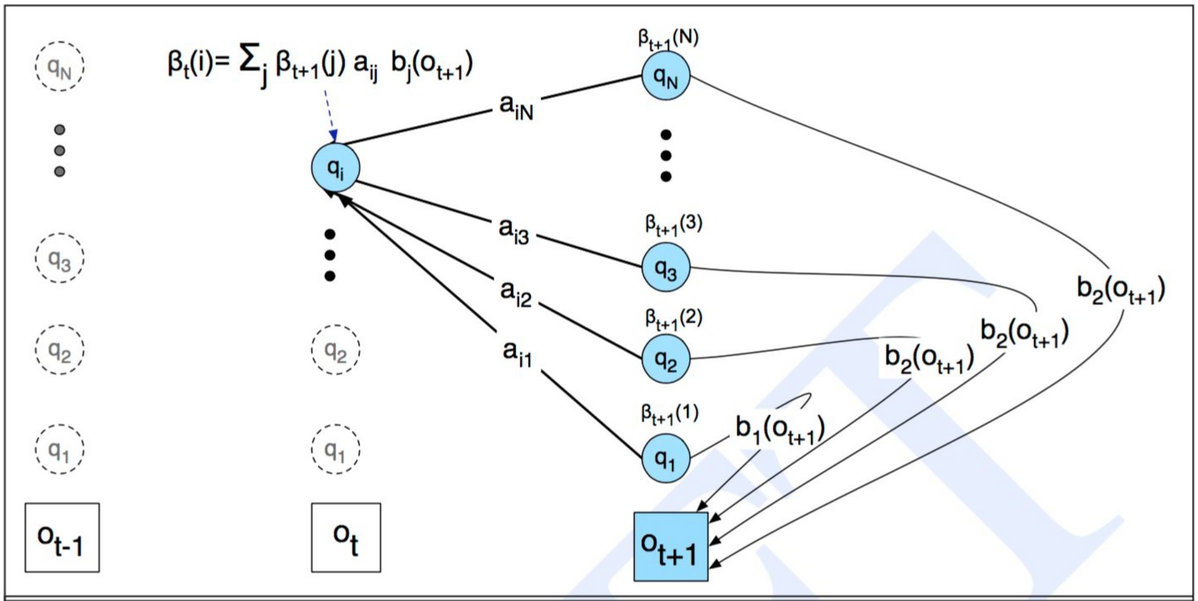
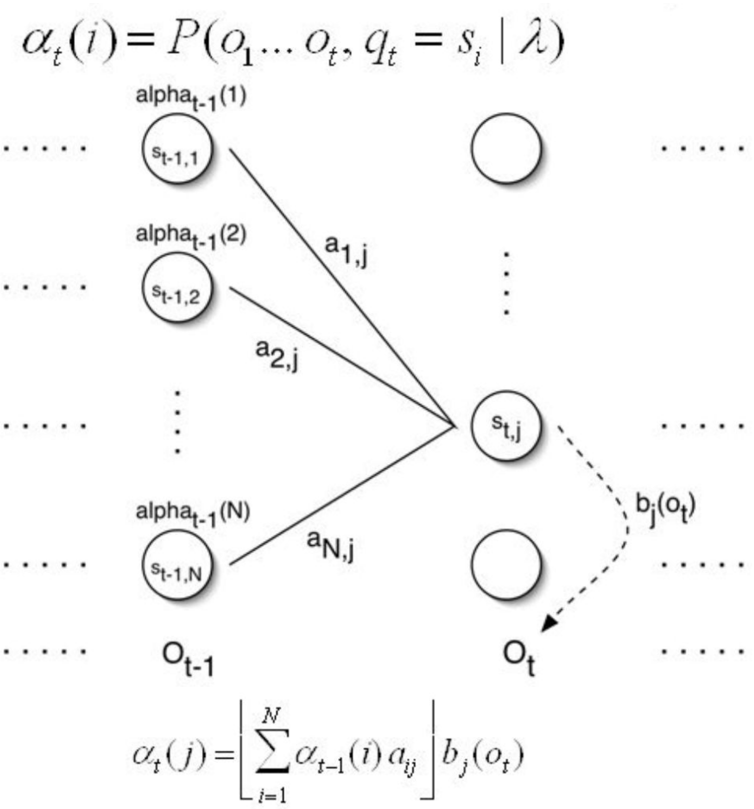


Figure 6.13 The computation of $\beta_t(i)$ by summing all the successive values $\beta_{t+1}(j)$ weighted by their transition probabilities a_{ij} and their observation probabilities $b_j(o_{t+1})$. Start and end states not shown.

And the image below represents forward probability.


$$\alpha_t(i) = P(o_1 \ldots o_t, q_t = s_i \mid \lambda)$$

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij}\right] b_j(o_t)$$

In the first pass, the forward–backward algorithm computes a set of forward probabilities which provide, for all k ϵ {1…,t}, the probability of ending up in any particular state given the first k observations in the sequence, i.e.

$P(X_k \mid o_{1:k})$, In the second pass, the algorithm computes a set of backward probabilities which provide the probability of observing the remaining observations given any starting point k, i.e. $P(o_{k+1:t} \mid X_k)$. These two sets of probability distributions can then be combined to obtain the distribution over states at any specific point in time given the entire observation sequence:

$$P(X_k \mid o_{1:t}) = P(X_k \mid o_{1:k}, o_{k+1:t}) \propto P(o_{k+1:t} \mid X_k) P(X_k|o_{1:k})$$

The last step follows from an application of the Bayes' rule and the conditional independence of $o_{k+1:t}$ and $o_{1:k}$ given X_k.

Unit 4 - Multidisciplinary Natural Language Processing

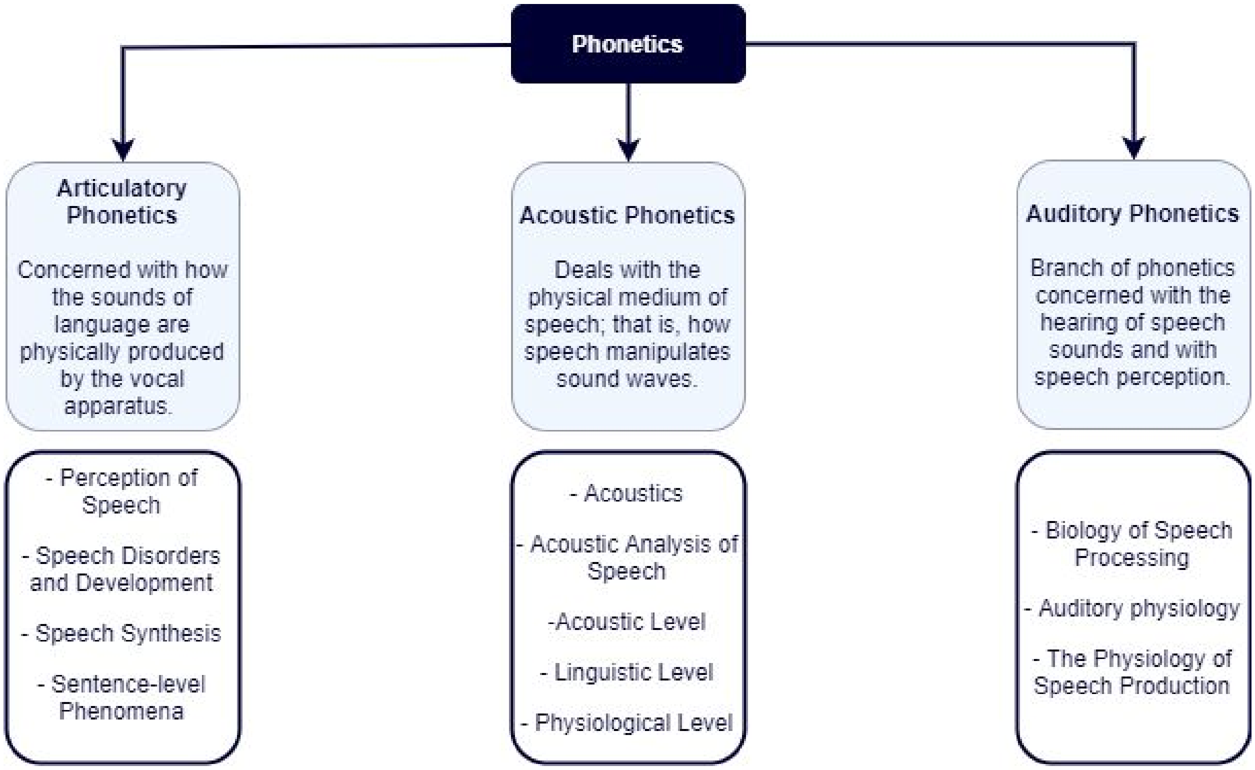
Table of Contents

Phonology -

Phonology is about patterns of sounds, especially different patterns of sounds in different languages, or within each language, different patterns of sounds in different positions in words etc.
Example - Dogs - sounds like “z” and Cats - sounds like “s”.

Phonetics -

It is the study of linguistic sounds, how they are produced by the articulators of the human vocal tract, how they are realized acoustically, and how this realization can be digitized and processed.
Example - Phones) i: - sheep and I-ship.


Phonetics
Articulatory Phonetics
Concerned with how the sounds of language are physically produced by the vocal apparatus.
Acoustic Phonetics
Deals with the physical medium of speech; that is, how speech manipulates sound waves.
Auditory Phonetics
Branch of phonetics concerned with the hearing of speech sounds and with speech perception.
- Perception of Speech
- Speech Disorders and Development
- Speech Synthesis
- Sentence-level Phenomena
- Acoustics
- Acoustic Analysis of Speech
-Acoustic Level
- Linguistic Level
- Physiological Level
- Biology of Speech Processing
- Auditory physiology
- The Physiology of Speech Production

Acoustics-

It is the branch of physics concerned with the properties of sound.

Acoustic Phonetics -[19]

Acoustic Analysis is based on sine and cosine functions.

The equation can be represented as:

$y=A*\sin(2\pi ft)$

Where, A is amplitude and f is frequency measured in cycles per second or Hertz.

Representation of sound waves in speech is by plotting change in air pressure over time.

Representation needs to be made in digital form and it needs following steps:

1. Analog-to-digital Conversion

1.1 Sampling

1.2 Quantization

Example - Telephone Speech is sampled at 16kHz and stored as 16-bit samples.

The waveform of the sentence "She just had a baby" is shown -

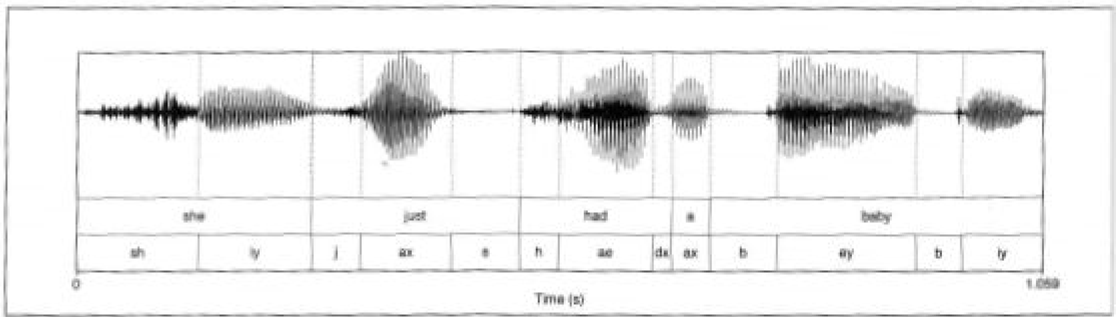


Figure 7.17 A waveform of the sentence "She just had a baby" from the Switchboard corpus (conversation 4325). The speaker is female, was 20 years old in 1991, which is approximately when the recording was made, and speaks the South Midlands dialect of American English.

A representation of the last vowel "iy" in the word "baby" of the above sentence is shown and explained below -

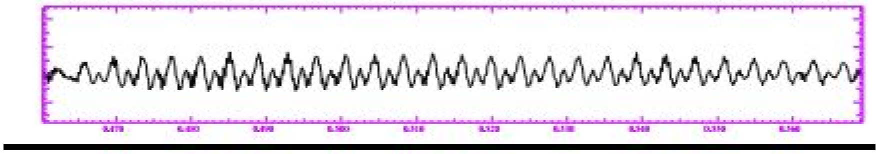


Figure 7.19 A waveform of the vowel [iy] from the utterance shown in Figure 7.20. The y-axis shows the changes in air pressure above and below normal atmospheric pressure. The x-axis shows time. Notice that the wave repeats regularly.

The vertical axis measures the amount of air pressure variation. A high value on the vertical axis (a high amplitude) indicates that there is more air pressure at that point in time, a zero value means there is normal

[19] Speech and Language Processing, Second Edition, Daniel Jurafsky, James H. Martin. Chapter - Phonetics

(atmospheric) air pressure, while a negative value means there is lower than normal air pressure (rarefaction).
Other properties related to frequency and amplitude are -

1. Pitch -

   Its a perceptual correlate of frequency, that is if a sound has a higher-frequency we perceive it as having a higher pitch. This uses either amplitude or root mean square of amplitude that is power or intensity that is normalized power to attain the pitch.

   Pitch plot is shown below-

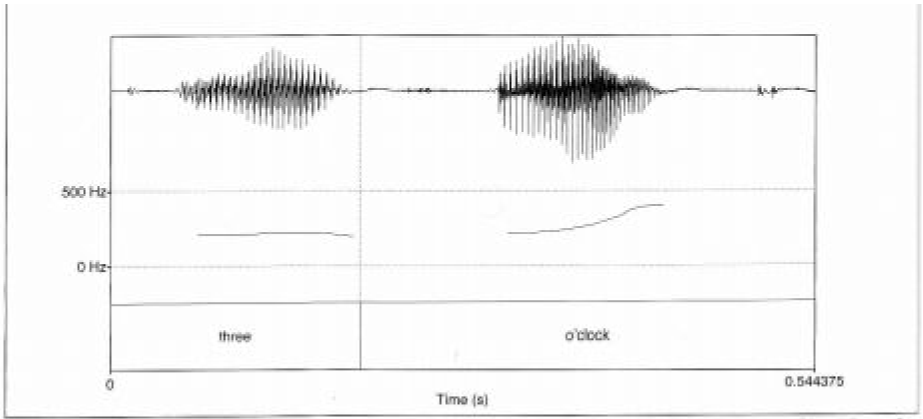
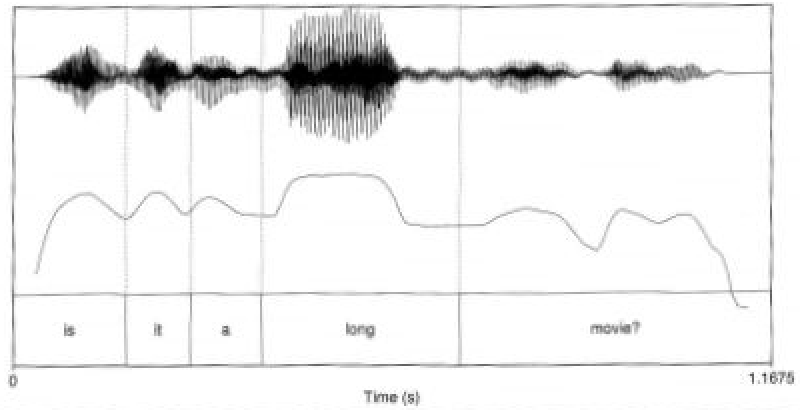


Figure 7.15 Pitch track of the question "Three o'clock?", shown below the wavefile. Note the rise in F0 at the end of the question. Note the lack of pitch trace during the very quiet part (the "o'" of "o'clock"; automatic pitch tracking is based on counting the pulses in the voiced regions, and doesn't work if there is no voicing (or insufficient sound).

   Intensity plot is shown below-



igure 7.16 Intensity plot for the sentence "Is it a long movie?". Note the intensity peaks a ach vowel and the especially high peak for the word *long*.

2. Loudness -

   Its a perceptual correlate of the power, which is related to the square of the amplitude. So sounds with higher amplitudes are perceived as louder.

Acoustic features are :

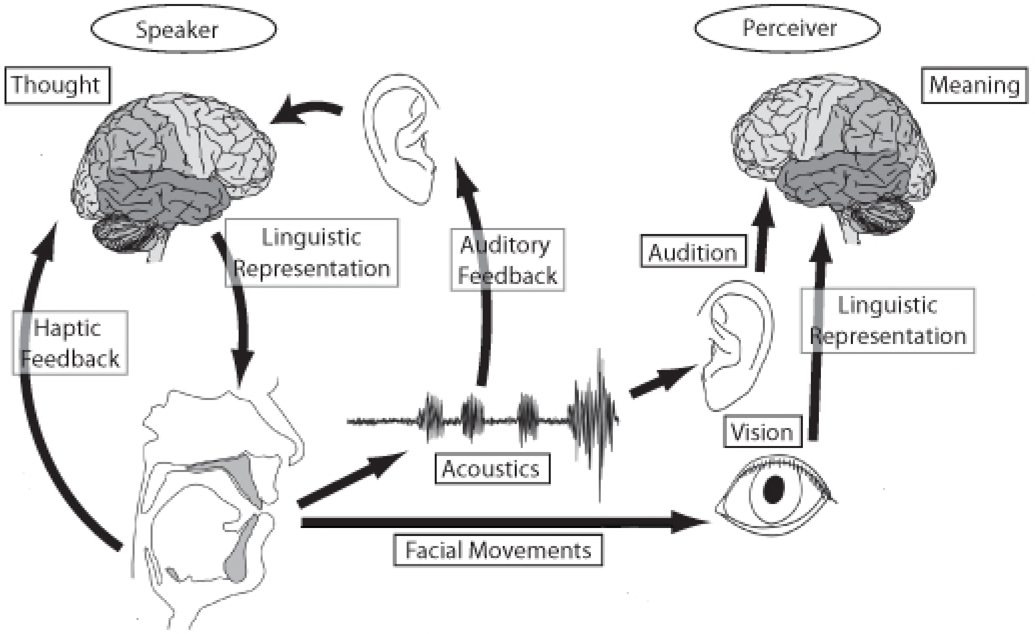
1. fundamental frequency (measured in hertz, or cycles per second)
2. duration (measured in time units such as milliseconds or seconds)

3. intensity, or sound pressure level (measured in decibels)

4. spectral characteristics (distribution of energy at different parts of the audible frequency range)

The Speech Chain[20]

The speech chain describes the stages in speech communication when a message moves between the mind of the speaker and the mind of the listener. Through the idea of the speech chain we see that information which is communicated linguistically to achieve some goal is encoded by the speaker into a sequence of articulatory gestures which generate sound, that sound is communicated to the listener, processed by the hearing mechanism into a neural code that is decoded to extract the meaning of the utterance and the intention of the communicative act.



Linguistic Level (Speaker Side)

Human will select, combine, and order suitable words into suitable sentences. Words are comprised of syllables and syllables are comprised of phonemes.

Physiological Level (Articulatory Speaker Side)

Involves neural and muscular activity, based on the input signal (phoneme) from the previous linguistic state. The muscular activity will form the articulator in a certain shape, so that the vocal tract for a specific phoneme is formed.

Acoustic Level (Transmission)

The ‘state’ moves to acoustic level when there is air flowing through the vocal tract from the lungs. At this level, speech sound wave is generated and then transmitted on the medium of air.

Physiological Level (Auditory Listener Side)

When the speech sound wave reaches listener’s ears the incoming wave will activate the hearing mechanism, which then involve neural activity in the hearing and perceptual mechanism.

Linguistic Level (Listener Side)

The speech chain is completed when the listener recognizes the words and sentences produced by the speaker.

[20] http://www.phon.ucl.ac.uk/courses/spsci/iss/week1.php

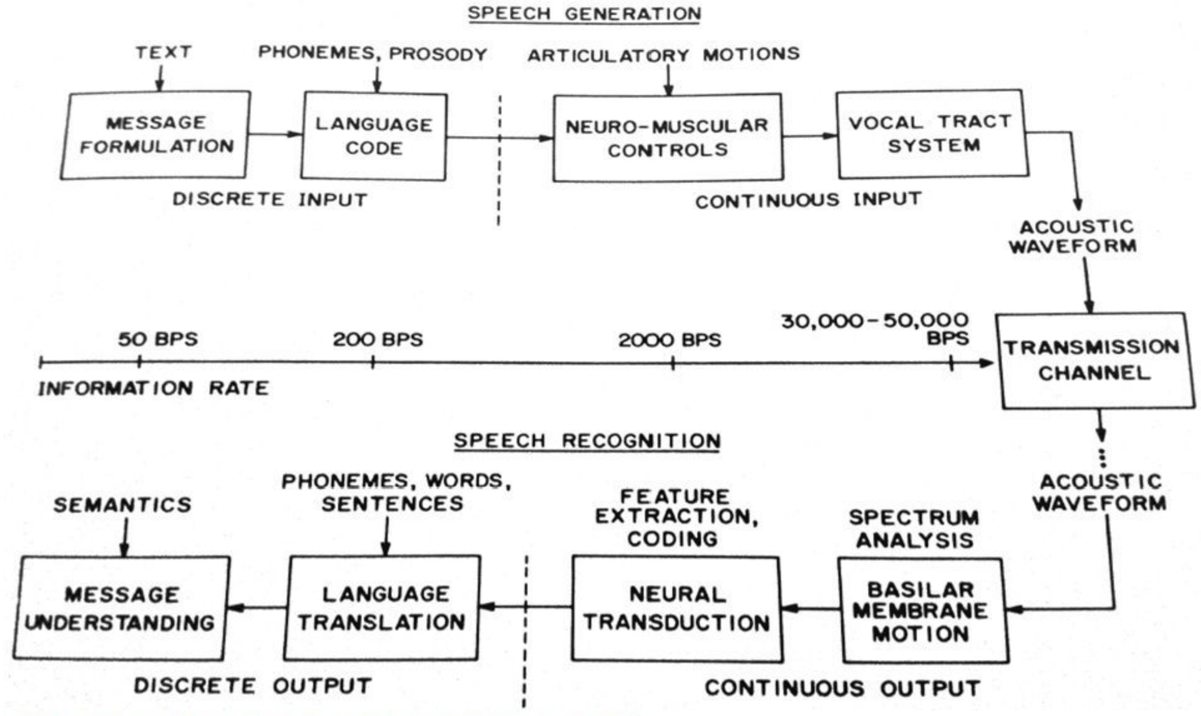
The steps can be roughly put across as below -

1. Encoding of pronunciation elements of the message as articulations (Linguistic level Speaker)
2. Aeroacoustic processes that generate sound from articulation (Physiological level Speaker)
3. Transmission of sound (Acoustic Level)
4. Audition of sound (Physiological level Listener)
5. Interpretation of auditory sensations in terms of pronunciation elements (Linguistic level Listener)

Speech Synthesis [21]

The process of speech synthesis (production) and speech perception by the acoustic phonetic method can be summarised by the following diagram:

Speech Production and Perception Process



Fundamentals of Speech Recognition". L. Rabiner & B. Juang. 1993

Note: Observe the discrete symbol information rate in various steps.

The speech-perception mechanism follows the inverse pattern of the speech synthesis process.

[21] "Fundamentals of Speech Recognition", L. Rabinar & B. Juang. 1993

The Perception of Speech[22]

Three approaches for automatic speech recognition (ASR):

1) **The acoustic-phonetic approach**

   Machine attempts to decode the speech signal in a sequential manner based on the observed acoustic features of the signal and the relations between acoustic and phonetic symbols.

2) **The pattern recognition approach**

   Four steps:

   i) Feature measurement

   Feature measurement is the output of any spectral analysis technique like DFT, filter bank analyzer, linear predictive coding analysis etc.

   ii) Pattern Training

   One or more test patterns corresponding to speech sounds of the same class are used to create a pattern representative (a.k.a. Reference pattern or exemplar or template) of the features of that class.

   In this step the machine learns which acoustic properties of the speech are reliable and repeatable across all training tokens of the pattern belonging to a class..

   iii) Pattern classification

   Unknown pattern is compared with each class reference pattern and a measure of similarity between the test pattern and each reference pattern is computed. Foe the comparison of speech patterns we require a local distance measure(which measures the spectral distance between two spectral vectors) and a global time alignment procedure (compensates for different rates of speaking of two patterns).

   iv) Decision logic

   Reference pattern similarity scores are used to decide which reference pattern best matches the unknown test pattern.

3) **The artificial intelligence approach**

   This approach compiles and incorporates knowledge from various knowledge sources and use it to solve the problem at hand (knowledge sources (KS) : acoustic, lexical, syntactic, semantic, pragmatic). The AI approach cn be implemented with neural networks.

   Knowledge sources can be integrated with the following methods:

   - Bottom-up approach : lowest level processes (feature detection, phonetic decoding) precede higher level processes (lexical decoding, language model etc) sequentially

[22] "Fundamentals of Speech Recognition", L. Rabinar & B. Juang. 1993

Sec. 2.5 Approaches to Automatic Speech Recognition by Machine 55

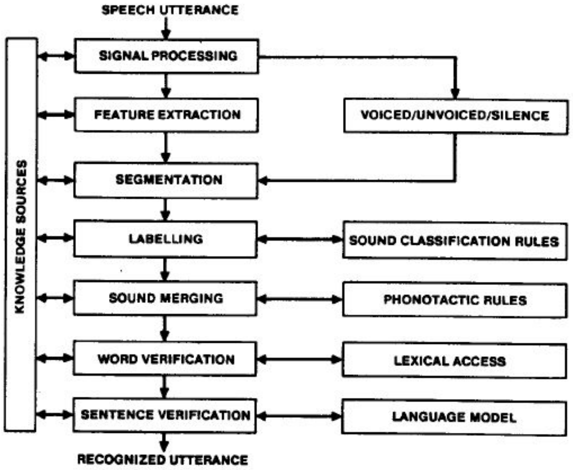


Figure 2.39 A bottom-up approach to knowledge integration for speech recognition.

- Top-down approach: language model generates word hypothesis that are matched with the speech signal and meaningful sentences are developed on basis of word match score

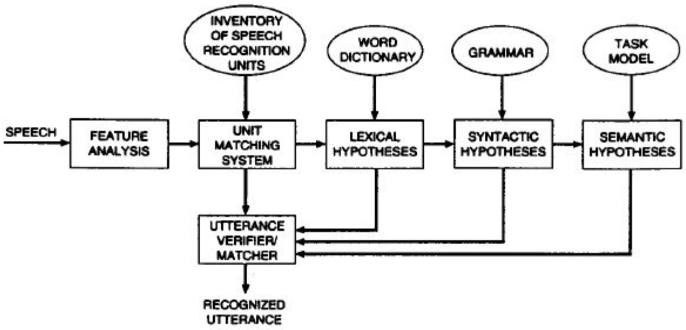


Figure 2.40 A top-down approach to knowledge integration for speech recognition.

- Blackboard approach: every knowledge source (KS) is independent; a hypothesis and test paradigm serves as a basic medium of communication between KSs; each KS is data driven - based on occurrence of patterns on the blackboard that matches the templates specified by the KS; the system activity operates asynchronously

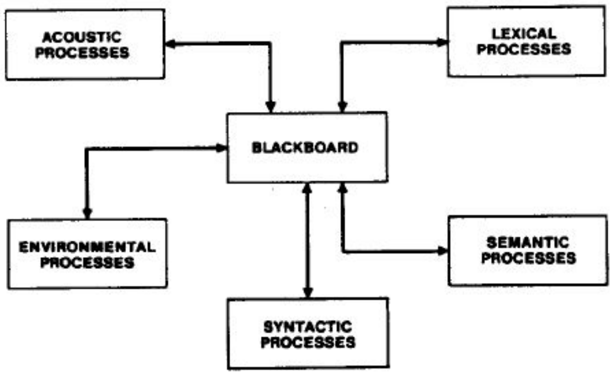
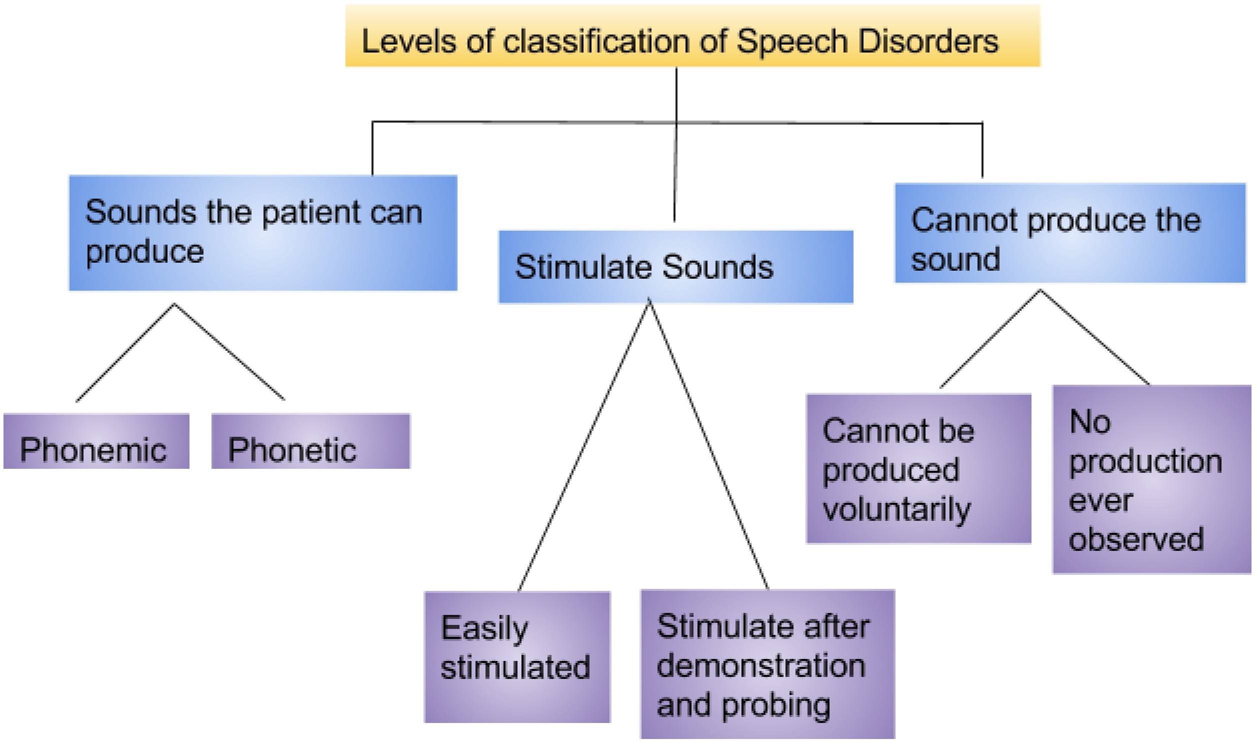


Figure 2.41 A blackboard approach to knowledge integration for speech recognition (after Lesser et al. [11]).

Speech Disorders and Development[23]


Levels of classification of Speech Disorders
Sounds the patient can produce
Stimulate Sounds
Cannot produce the sound
Phonemic
Phonetic
Easily stimulated
Stimulate after demonstration and probing
Cannot be produced voluntarily
No production ever observed

Types of speech disorders

- Apraxia of speech may result from stroke or progressive illness, and involves inconsistent production of speech sounds and rearranging of sounds in a word ("potato" may become "topato" and next "totapo"). Production of words becomes more difficult with effort, but common phrases may sometimes be spoken spontaneously without effort.
- Cluttering, a speech and fluency disorder characterized primarily by a rapid rate of speech, which makes speech difficult to understand.
- Developmental verbal dyspraxia also known as childhood apraxia of speech.
- Dysarthria is a weakness or paralysis of speech muscles caused by damage to the nerves or brain. Dysarthria is often caused by strokes, Parkinson's disease, ALS, head or neck injuries, surgical accident, or cerebral palsy.
- Dysprosody is the rarest neurological speech disorder. It is characterized by alterations in intensity, in the timing of utterance segments, and in rhythm, cadence, and intonation of words. The changes to the duration, the fundamental frequency, and the intensity of tonic and atonic syllables of the sentences spoken, deprive an individual's particular speech of its characteristics. The cause of dysprosody is usually associated with neurological pathologies such as brain vascular accidents, cranioencephalic traumatisms, and brain tumors.
- Muteness is complete inability to speak.
- Speech sound disorders involve difficulty in producing specific speech sounds (most often certain consonants, such as /s/ or /r/), and are subdivided into articulation disorders (also called phonetic disorders) and phonemic disorders. Articulation disorders are characterized by difficulty learning to produce sounds physically. Phonemic disorders are characterized by difficulty in learning the sound

[23] https://en.wikipedia.org/wiki/Speech_disorder

distinctions of a language, so that one sound may be used in place of many. However, it is not uncommon for a single person to have a mixed speech sound disorder with both phonemic and phonetic components.

- Stuttering affects approximately 1% of the adult population.[1]
- Voice disorders are impairments, often physical, that involve the function of the larynx or vocal resonance.

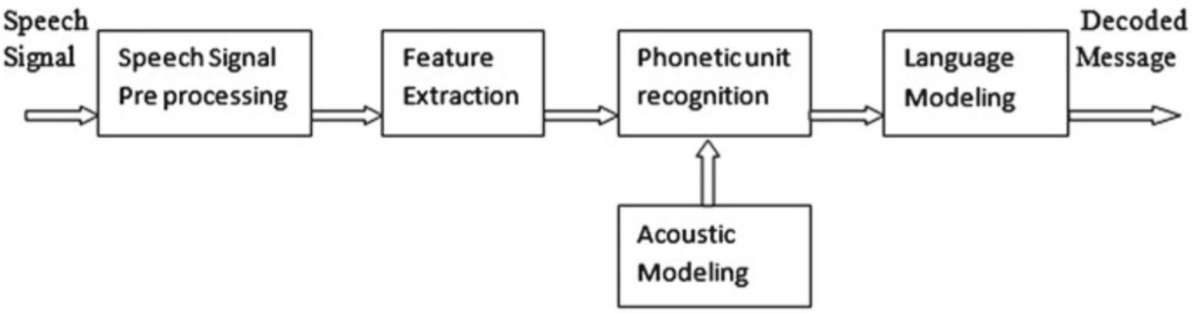
Biology of speech processing[24]

Speech processing is the process by which speech signals are interpreted, understood, and acted upon. It specifically refers to the processing of human speech by computerized systems, as in voice recognition software or voice-to-text programs. Speech processing is important to many fields for both theoretical and practical uses, ranging from voice activation and control in phones to development of functional artificial intelligence in computer science. Interpretation and production of coherent speech are both important in the processing of speech.

Speech Recognition

Speech recognition is special case of speech processing. It deals with the analysis of the linguistic contents of a speech signal. Speech recognition is a method that uses an audio input for data entry to a computer or a digital system in place of a keyboard.

Speech recognition research is interdisciplinary in nature, drawing upon work in fields as diverse as biology, computer science, electrical engineering, linguistics, mathematics, physics, and psychology. Within these disciplines, pertinent work is being done in the areas of acoustics, artificial intelligence, computer algorithms, information theory, linear algebra, linear system theory, pattern recognition, phonetics, physiology, probability theory, signal processing, and syntactic theory.



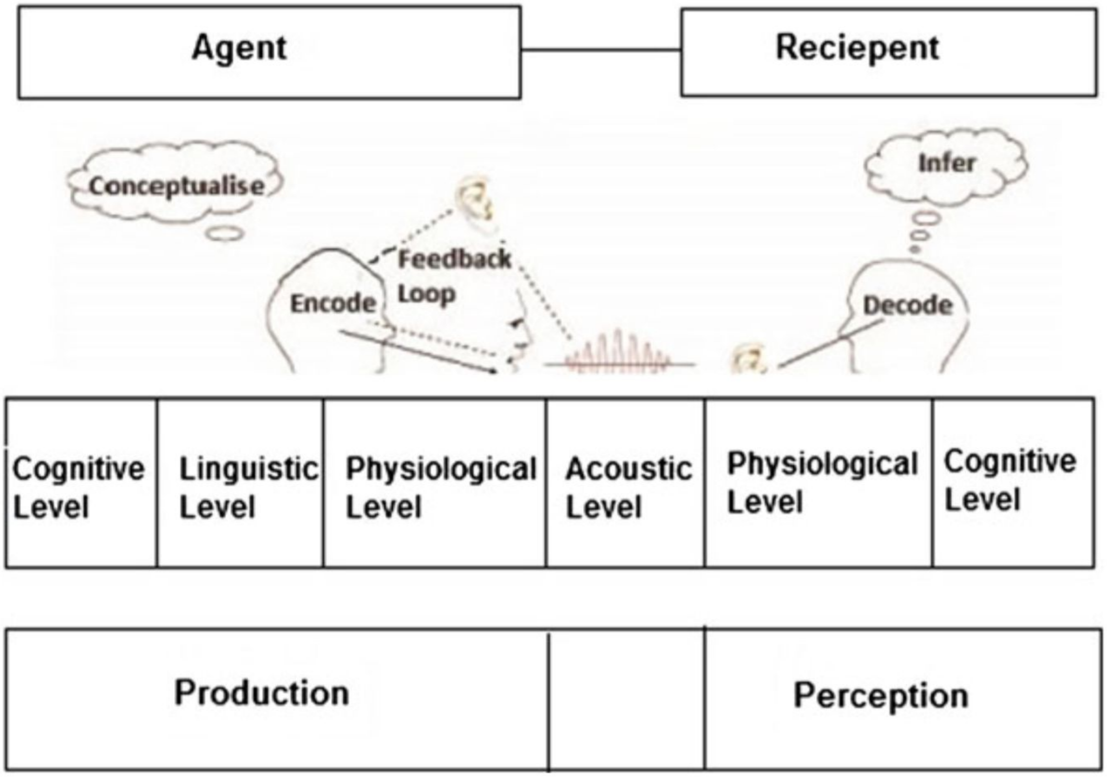
Structure of a standard speech recognition system

Speech Communication Chain

The fundamental purpose of generation of speech is communication. Speech communication involves a chain of physical and psychological events. The speaker initiates the chain by setting into motion the vocal apparatus, which propels a modu- lated airstream to the listener. The process culminates with the listener receiving the fluctuations in air pressure through the auditory mechanisms and subsequently parsing the signal into higher-level linguistic units. The three elements of the speech chain are production, transmission, and perception which depends upon the function of the cognitive, linguistic, physiological, and acoustic levels for both the speaker and recipient listener.

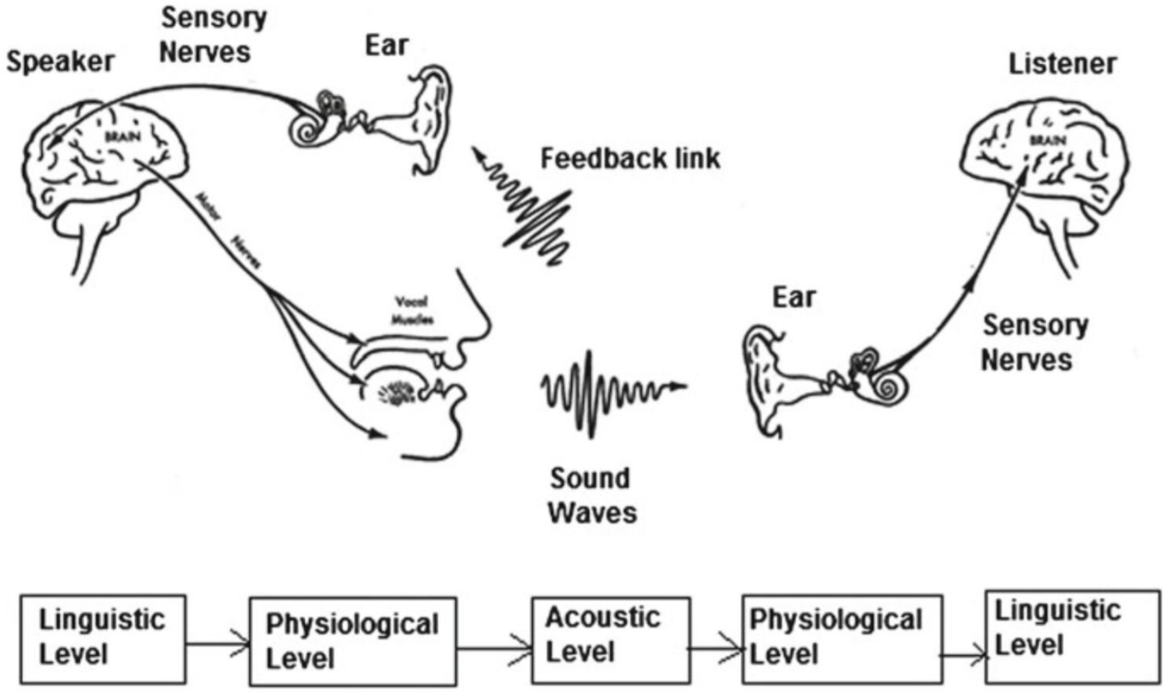
[24] http://www.springer.com/cda/content/document/cda_downloaddocument/9788132218616-c2.pdf?SGWID=0-0-45-1452213-p176639035

Figure below shows the communication chain underpinned by Shannon and Weaver's Information Theory and highlights the linguistic, physiological, and acoustic mechanisms by which humans encode, transmit, and decode meanings.



Communication Model

This model is an extension of the Speech Chain, originally outlined by Denes and Pinson (1973) [8], as shown in the figure below.



The Speech Chain

There are three main links in the communication chain:

1. Production: It is the process by which a human expresses himself or herself through first deciding what message is to be communicated (cognitive level). Next a plan is prepared and encoded with appropriate linguistic utterance to represent the concept (linguistic level) and, finally, produce this utterance through the suitable coordination of the vocal apparatus (physiological level). Production of a verbal utterance, therefore, takes place at three levels:

a. Cognitive: When two people talk together, the speaker sends messages to the listener in the form of words. The speaker has first to decide what he or she wants to say and then to choose the right words to put together in order to send the message. These decisions are taken in a higher level of the brain known as the cortex.

b. Linguistic:According To Connectionist Model,there are four layers processing at the linguistic level: semantic, syntactic, morphological, and phonological. These work in parallel and in series, with activation at each level. Interference and mis activation can occur at any of these stages. Production begins with concepts and continues down from there. Human has a bank of words stored in brains known as lexicon. It is built up over time, and the items we store are different from person to person. The store is largely dependent upon what one have been exposed to, such as the job of work it does, where one have lived, and so on. Whenever human needs to encode a word, a search is made within this lexicon in order to determine whether or not it already contains a word for the idea which is to be conveyed.

c. Physiological: Once the linguistic encoding has taken place, the brain sends small electrical signals along nerves from the cortex to the mouth, tongue, lips, vocal folds (vocal cords), and the muscles which control breathing to enable us to articulate the word or words needed to communicate our thoughts. This production of the sound sequence occurs at what is known as the physiological level, and it involves rapid, coordinated, sequential movements of the vocal apparatus.
2. **Transmission** is the sending of the linguistic utterance through some medium the recipient. As we are only concerned with oral verbal communication here, there is only one medium of consequence, and that is air, i.e., the spoken utterance travels through the medium of air to the recipient’s ear. There are, of course, other media through which a message could be transmitted. For example, written messages may be transmitted with ink and paper. However, because here, the only concern is the transmission of messages that use a so-called vocal-auditory channel, then transmission is said to occur at the acoustic level.

3.**Reception** is the process by which the recipient of a verbal utterance detects the utterance through the sense of hearing at physiological level and then decodes the linguistic expression at linguistic level. Then, the recipient infers what is meant by the linguistic expression at cognitive level [8]. Like the mirror image of production, reception also operates at the same three levels:

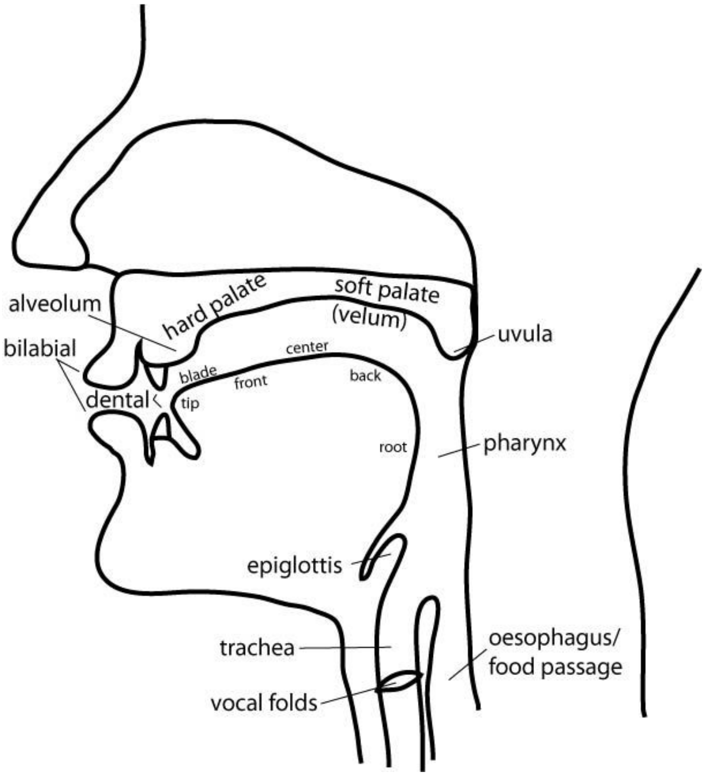
a. Physiological: When the speaker’s utterance transmitted acoustically as a speech sound wave arrives at the listener’s ear, it causes his or her eardrum to vibrate. This, in turn, causes the movement of three small bones within the middle ear. Their function is to amplify the vibration of the sound wave. One of these bones, the stapes, is connected to a membrane in the wall of nerve bundle called the cochlea. The cochlea is designed to convert the vibrations into electrical signals. These are subsequently transmitted along the 30,000 or so fibres that constitute the auditory nerve to the brain of the listener. Again, this takes place at the physiological level.

b. Linguistic:The listener subsequently decodes the electrical impulses in the cortex and reforms them into the word or words of the message, again at the linguistic level. The listener compares the decoded words with other words in its own lexicon. The listener is then able to determine that the word is a proper word. In order for a recipient to decode longer utterances and to interpret them as meaningful, it must also make use of the grammatical rules stored in the brain. These allow the recipient to decode an utterance such as “I have just seen a cat” as indicating that the event took place in the recent past, as opposed to an utterance such as “I will be seeing a cat” which grammatically indicates that the event has not yet happened but will happen some time in the future. Consequently, as with the agent, the recipient also needs access to a lexicon and a set of grammatical rules in order to comprehend verbal utterances.

c. Cognitive: In this level, the listener must infer the speaker’s meaning. A linguistic utterance can never convey all of a speaker’s intended meaning. A linguistic utterance is a sketch, and the listener must fill in the sketch by inferring the meaning from such things as body language, shared knowledge, tone of voice, and so on. Humans must, therefore, be able to infer meanings in order to communicate fully.

The Physiology of Speech Production[25]

Speech sounds rely on air supplied by the lungs enclosed by the rib cage and the diaphragm – a dome-shaped muscle that supports the base of the lungs and separates the abdomen from the thorax (chest).
Before we can start producing speech sounds, the air passes through several stages of the respiratory tract, such as the trachea and the larynx, after which it goes through the vocal tract.



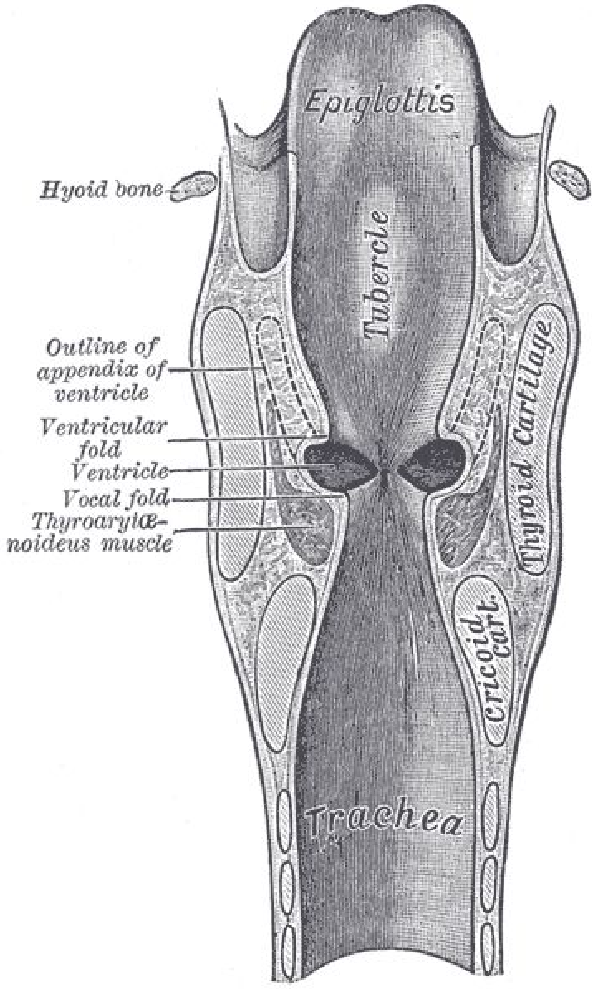
human vocal tract

The larynx is also known as the phonatory system and is located at the upper end of the trachea (or windpipe). The larynx is popularly known as the Adam’s Apple. It has both speech and physical (i.e. anatomical) functions. What concerns us is the speech function of the larynx as it acts as the sound producing organ of the vocal apparatus. For this purpose, we need to know its three major components. The larynx includes two big cartilages.The lower one is called the cricoid cartilage, which is situated just above the highest ring in the trachea. It has a typical signet-ring shape. Resting on this cartilage is the thyroid cartilage, which is made up of two square-shaped cartilages that are joined at the front of the larynx. At the top of where the two surfaces

[25] http://educypedia.karadimov.info/library/phon6.pdf

join together, there is a small V-shaped gap. At the top and bottom, each wall of the thyroid cartilage extends into horn-shaped protrusions – the cornua.

Inside the 'box' formed by the thyroid and cricoid cartilages we find the vocal folds (- lips/ - cords), i.e. two three-sided pyramid-shaped muscles, which project into the cavity of the larynx. The vocal folds are stretched across the larynx from front to back. The opening between the vocal folds is called the glottis.



Coronal section of larynx and upper part of trachea

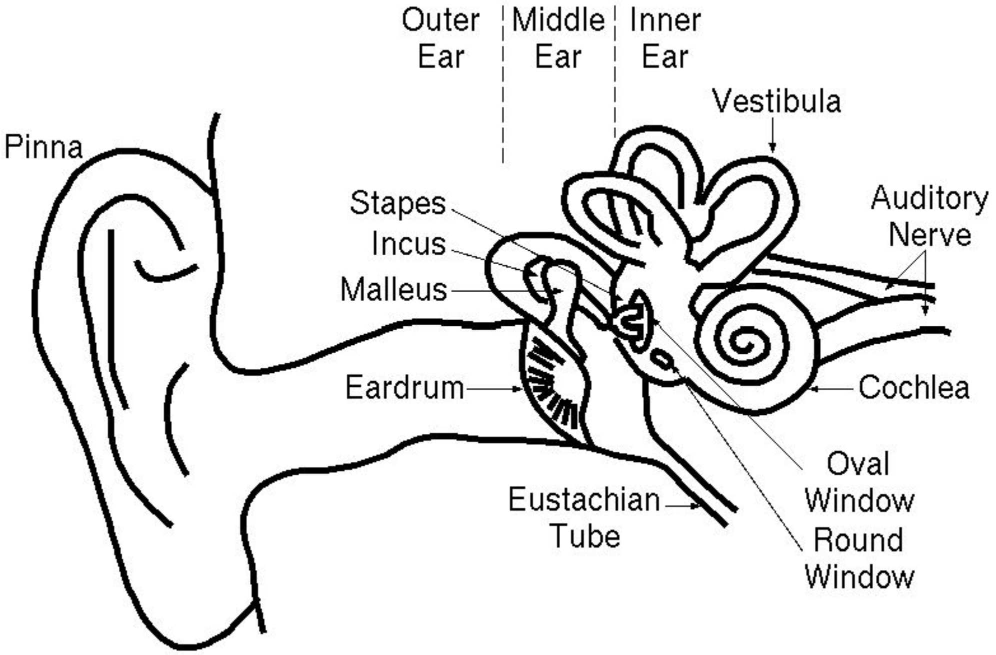
Let us now take a look at the sounds that correspond to the various positions of the vocal folds. When you are breathing normally, or producing a sound like /f/, they are wide apart (**abducted**), they can also be tightly closed (adducted), for instance, when we are lifting heavy objects. The process involving the vibration of the vocal folds can be distinguished into four stages. The first is closure (or **adduction**), during which the laryngeal muscles bring the vocal folds together. As the flow of air from the lungs continues, there is a build-up of air below the folds (subglottal pressure). This stage is known as **compression**. At some point, the compressed air will force apart the vocal folds so that a little air escapes for a brief moment. This phase is the **release**. As the air flows past, the vocal folds are brought back together as a result of two forces: the elasticity of the folds, and the Bernouilli effect. When the vocal folds are brought back together again, the subglottal pressure again builds up, and the process repeats itself. The entire process of opening and closing of the vocal folds is called the glottal cycle by phoneticians. On average, this occurs between 200 and 300 times per second (expressed as cycles per second) in women, and about 100 to 150 times per second in men. It is important to add that the vibration rate does not remain constant in speech, and is controlled by the speaker who changes it in order to achieve certain perceptual effects. An increase in the frequency of this process leads to an increase in the pitch of the voice, and vice versa.

This rapid vibration of the vocal folds results in a typical buzzing sound, which we call voice, or phonation. This can be heard in consonants like /z/ or /b/, which are then said to be voiced. In English, for instance, all vowels and nasal consonants (e.g. /m/) are voiced. When the folds are apart specialists use the term nil phonation, and any speech sounds produced without the vibration of the vocal folds are voiceless (or aphonic): e.g. /p/, /t/.

There are a number of types of phonation. Up until now, we have only talked about true (or modal) phonation, i.e. that used in the production of voiced speech sounds in English. However, the vocal folds can operate in a number of different ways, resulting in different types of phonation. Such as murmur, creaky voice, whisper.

Auditory physiology[26]

The auditory system is comprised of three components; the outer, middle, and inner ear, all of which work together to transfer sounds from the environment to the brain.



THE OUTER EAR : Includes the portion of the ear that we see—the pinna/auricle and the ear canal.
Pinna : The pinna or auricle is a concave cartilaginous structure, which collects and directs sound waves traveling in air into the ear canal or external auditory meatus.

Ear Canal : The inner two-thirds of the ear canal is imbedded in the temporal bone. The outer one-third of the canal is cartilage. Although the shape of each ear canal varies, in general the canal forms an elongated "s" shape curve. The ear canal directs airborne sound waves towards the tympanic membrane (eardrum). The ear canal resonates sound waves and increases the loudness of the tones in the 3000-4000 Hz range. The ear canal maintains the proper conditions of temperature and humidity necessary to preserve the elasticity of the tympanic membrane. Glands, which produce cerumen (earwax) and tiny hairs in the ear canal, provide added protection against insects and foreign particles from damaging the tympanic membrane.

MIDDLE EAR : The middle ear is composed of the tympanic membrane and the cavity, which houses the ossicular chain.

Tympanic Membrane : The tympanic membrane or eardrum serves as a divider between the outer ear and the middle ear structures. The eardrum is very sensitive to sound waves and vibrates back and forth as the

[26] http://www.workplaceintegra.com/hearing-articles/Ear-anatomy.html

sound waves strike it. The eardrum transmits the airborne vibrations from the outer to the middle ear and also assists in the protection of the delicate structures of the middle ear cavity and inner ear.

Middle Ear Cavity : The middle ear cavity extends from the tympanic membrane to the inner ear. It is approximately two cubic centimeters in volume and is lined with mucous membrane. The middle ear cavity is actually an extension of the nasopharynx via the eustachian tube.

Eustachian Tube : The eustachian tube acts as an air pressure equalizer and ventilates the middle ear. Normally the tube is closed but opens while chewing or swallowing. When the eustachian tube opens, the air pressure between the outer and middle ear is equalized. The transmission of sound through the eardrum is optimal when the air pressure is equalized between the outer and middle ear. When the air pressure between the outer and middle ear is unequal, the eardrum is forced outward or inward causing discomfort and the ability of the eardrum to transmit sound is reduced.

Ossicular Chain : The middle ear is connected and transmits sound to the inner ear via the ossicular chain. The ossicular chain amplifies a signal approximately 25 decibels as it transfers signals from the tympanic membrane to the inner ear.

THE INNER EAR

The inner ear is composed of the sensory organ for hearing—the cochlea, as well as for balance—the vestibular system. The systems are separate, yet both are encased in the same bony capsule and share the same fluid systems.

Vestibular or Balance System : The balance part of the ear is referred to as the vestibular apparatus. It is composed, in part, of three semicircular canals located within the inner ear. The vestibular system helps to maintain balance, regardless of head position or gravity, in conjunction with eye movement and somatosensory input. The semicircular canals are innervated by the VIIIth cranial nerve.

Cochlea : The hearing part of the inner ear is the cochlea. The cochlea is spiral-shaped, similar to the shape of a snail. The cochlea is composed of three fluid-filled chambers that extend the length of the structure. The two outer chambers are filled with a fluid called perilymph. Perilymph acts as a cushioning agent for the delicate structures that occupy the center chamber. The third fluid filled chamber is the center chamber, called the cochlear duct. The cochlear duct secretes a fluid called endolymph, which fills this chamber. The cochlear duct contains the Basilar membrane upon which lies the Organ of Corti. The Organ of Corti is a sensory organ essential to hearing. It consists of approximately 30,000 finger-like projections of cilia that are arranged in rows. These cilia are referred to as hair cells. Each hair cell is connected to a nerve fiber that relays various impulses to the cochlear branch of the VIIIth cranial nerve or auditory nerve. The "pitch" of the impulse relayed is dependent upon which areas of the basilar membrane, and hence, which portions of the Organ of Corti are stimulated. The apical portion of the basilar membrane (the most curled area of the cochlea) transfers lower frequency impulses. The basal end relays higher frequency impulses.

The VIII cranial nerve (VIII C.N.) or auditory C.N. carries the impulses generated from the Organ of Corti to the brainstem. From the brainstem, nerve pathways extend through numerous nuclei to the cerebral cortex in the temporal lobes of the brain. It is in the temporal lobes of the brain that meaning is associated with the various patterns of nerve impulses.

THE PHYSIOLOGY OF HEARING

The process of hearing begins with the occurrence of a sound. Sound is initiated when an event moves and causes a motion or vibration in air. When this air movement stimulates the ear, a sound is heard.

In the human ear, a sound wave is transmitted through four separate mediums along the auditory system before a sound is perceived: in the outer ear—air, in the middle ear— mechanical, in the inner ear liquid and to the brain—neural.

Sound Transmission through the Outer Ear

Air transmitted sound waves are directed toward the delicate hearing mechanisms with the help of the outer ear, first by the pinna, which gently funnels sound waves into the ear canal, then by the ear canal.

Sound Transmission through the Middle Ear

When air movement strikes the tympanic membrane, the tympanic membrane or eardrum moves. At this point, the energy generated through a sound wave is transferred from a medium of air to that which is solid in the middle ear. The ossicular chain of the middle ear connects to the eardrum via the malleus, so that any motion of the eardrum sets the three little bones of the ossicular chain into motion.

Sound Transmission through the Inner Ear

The ossicular chain transfers energy from a solid medium to the fluid medium of the inner ear via the stapes. The stapes is attached to the oval window. Movement of the oval window creates motion in the cochlear fluid and along the Basilar membrane. Motion along the basilar membrane excites frequency specific areas of the Organ of Corti, which in turn stimulates a series of nerve endings.

Sound Transmission to the Brain

With the initiation of the nerve impulses, another change in medium occurs: from fluid to neural. Nerve impulses are relayed through the VIII C.N., through various nuclei along the auditory pathway to areas to the brain. It is the brain that interprets the neural impulses and creates a thought, picture, or other recognized symbol.

Unit 5 - Multidisciplinary Natural Language Processing

Table of Contents

Lexical Knowledge Networks

They are networks that represent semantic relations between concepts. This is often used as form of the knowledge representation and is similar to a graph (directed or undirected). A semantic network is used when one has knowledge that is best understood as a set of concepts that are related to one another.
Limitation is that they do not represent performance or meta-knowledge very well.

Some Lexical Network examples -
- WordNet[27]
Wordnet is a lexical knowledge base based on conceptual look up, it organizes lexical information in terms of word meaning rather than word form. And the main use in natural language processing for wordnet is the word sense disambiguation.
Word sense disambiguation is the determination of the correct sense of the word. Eg. we have two sentences "the crane ate the fish" versus "the crane was used to live the load", and in the first case the crane ate the fish, crane is used in the sense of the bird, and the crane was used to lift the load here the crane was used in the sense of a machine.

- MindNet[28]
MindNet is a knowledge representation project that uses Microsoft's broad-coverage parser to build semantic networks from dictionaries, encyclopedias, and free text. MindNets are produced by a fully automatic process that takes the input text, sentence-breaks it, parses each sentence to build a semantic dependency graph (Logical Form), aggregates these individual graphs into a single large graph, and then assigns probabilistic weights to subgraphs based on their frequency in the corpus as a whole. The project also encompasses a number of mechanisms for searching, sorting, and measuring the similarity of paths in a MindNet.

- VerbNet[29]
VerbNet (Kipper-Schuler 2006) is the largest on-line verb lexicon currently available for English. It is a hierarchical domain-independent, broad-coverage verb lexicon with mappings to other lexical resources such as WordNet (Miller, 1990; Fellbaum, 1998), Xtag (XTAG Research Group, 2001), and FrameNet (Baker et al., 1998). VerbNet is organized into verb classes through refinement and addition of subclasses to achieve syntactic and semantic coherence among members of a class.

Applications
Network models may provide new insight into the semantic content of large text collections. For example, semantic similarity networks may be used to identify shifts in topics and identify fake or forged articles. In addition, semantic similarity networks can be used for traditional information retrieval tasks such as document clustering.

[27] http://nptel.ac.in/courses/106101007/29
[28] https://www.microsoft.com/en-us/research/project/mindnet/
[29] https://verbs.colorado.edu/verbnet/

Metaphors

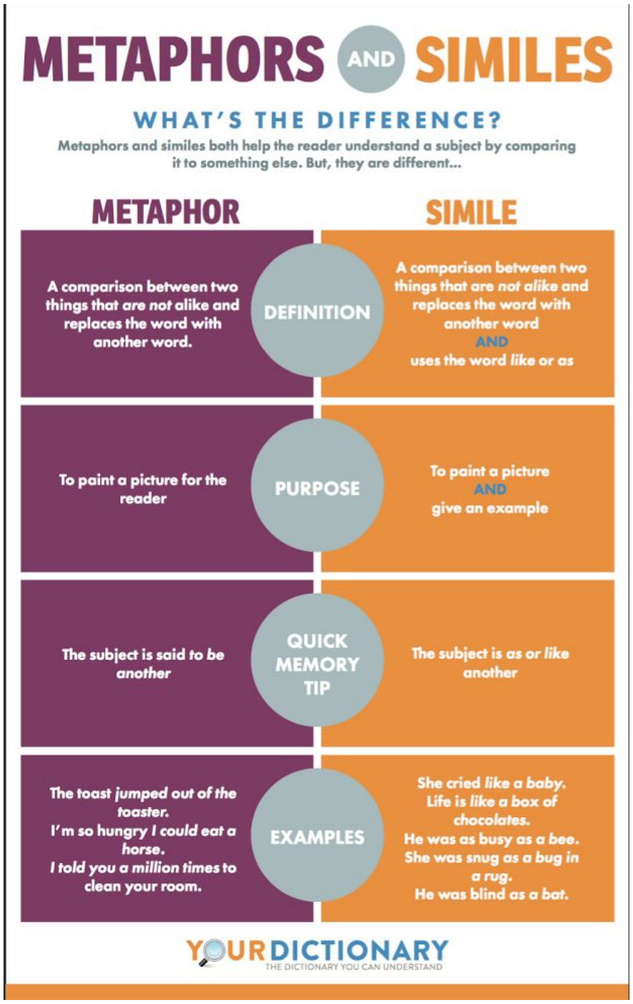
Metaphor is a rhetorical figure of speech that compares two subjects without the use of “like” or “as.” Metaphor is often confused with simile, which compares two subjects by connecting them with “like” or “as” (for example: “She’s fit as a fiddle”). While a simile states that one thing is like another, a metaphor asserts that one thing is the other, or is a substitute for the other thing.

A metaphor asserts a correlation or resemblance between two things that are otherwise unrelated. The English word “metaphor” originates from the Greek metaphorá, which means “to transfer” or “to carry over.” Indeed, a metaphor transfers meaning from one subject on to another so that the target subject can be understood in a new way.

Rhetoricians have further elaborated on the definition of metaphor by separating and naming the two key elements. There are a few different sets of names for these two parts: they can be called the “tenor” and the “vehicle”, the “ground” and the “figure”, or the “target” and the “source”. Consider this famous example of a metaphor from Shakespeare’s “As You Like It”:

All the world’s a stage,
And all the men and women merely players.

In this example, the world is the primary subject, and it gains attributes from the stage (ie, from theater). Thus, in the binary pairs, the world is the “tenor,” the “ground,” and the “target,” while the stage is the “vehicle,” the “figure,” and the “source.”



METAPHORS AND SIMILES

WHAT’S THE DIFFERENCE?

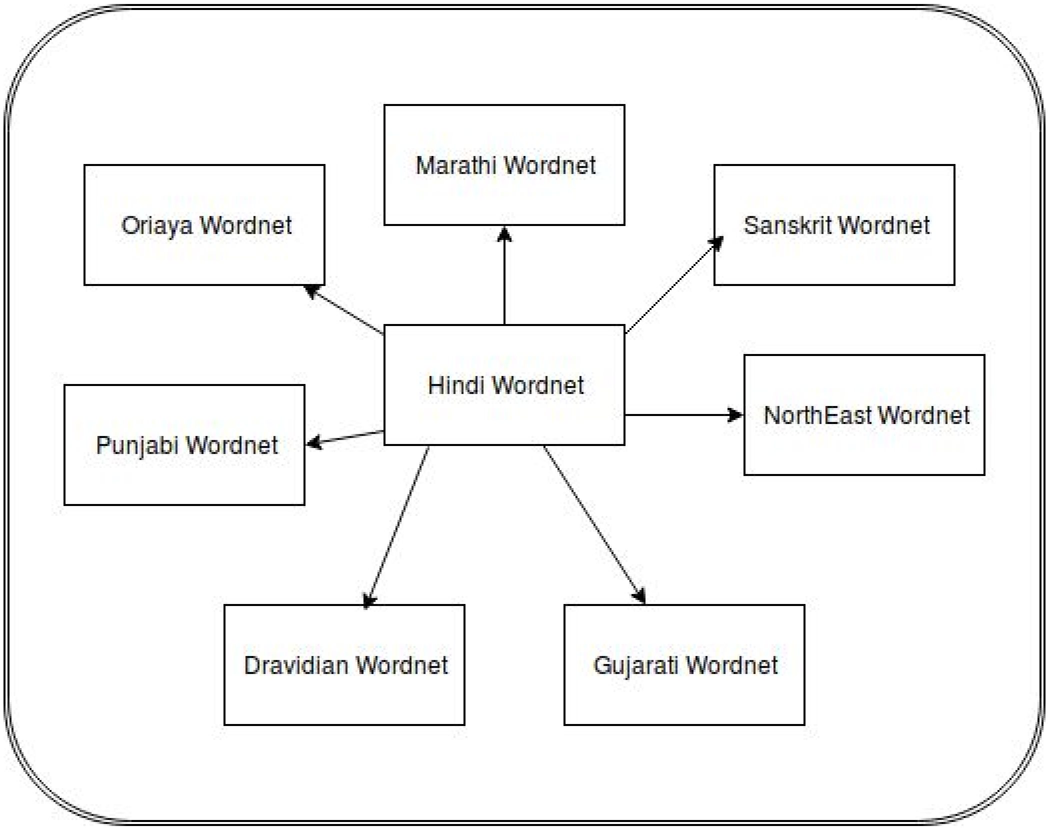
Metaphors and similes both help the reader understand a subject by comparing it to something else. But, they are different...

	METAPHOR	SIMILE
DEFINITION	A comparison between two things that *are not* alike and replaces the word with another word.	A comparison between two things that are *not alike* and replaces the word with another word AND uses the word *like* or *as*
PURPOSE	To paint a picture for the reader	To paint a picture AND give an example
QUICK MEMORY TIP	The subject is said *to be another*	The subject is *as* or *like* another
EXAMPLES	The toast *jumped out of the toaster.* I’m so hungry *I could eat a horse.* *I told you a million times* to clean your room.	She cried *like a baby.* Life is *like a box of chocolates.* He was *as busy as a bee.* She was *snug as a bug in a rug.* He was *blind as a bat.*

YOURDICTIONARY
THE DICTIONARY YOU CAN UNDERSTAND

Indian Language WordNets and Multilingual Dictionaries

India is a multilingual country where machine translation and cross lingual search are highly relevant problems. This led the formation of Indian Language WordNets.
i) IndoWordNet[30] http://www.cfilt.iitb.ac.in/indowordnet/index.jsp
IndoWordNet is a linked lexical knowledge base of wordnets of 18 scheduled languages of India, viz., Assamese, Bangla, Bodo, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Malayalam, Meitei (Manipuri), Marathi, Nepali, Odia, Punjabi, Sanskrit, Tamil, Telugu and Urdu.



The first step in IndoWordNet creation was the construction of synsets for most common concepts which are universal across languages. The WordNet was created using MultiDict tool developed at the Center for Indian Language Technology, Computer Science Department, IIT Bombay.
Right from the beginning, IWN insisted on storing lexical links expressing relationship of derivational morphology. Indian languages are rich in morphology . In Sanskrit wordnet, for example, the theory that all words are derived from verbal roots- dhaatus- is being seriously examined for its use as a fundamental guiding principle for storing and linking word.
Causative verb form sare a typically occurring phenomenon in Indian languages. For example, khaanaa(to eat), khilaana (to feed)and khilwaanaa (to cause to feed)are forms derived from the same root khaanaa. It has been decided to take special care to store causative forms in IWN and link them to their basic roots.

ii) Hindi WordNet http://www.cfilt.iitb.ac.in/wordnet/webhwn/

- Hindi WordNet is an on-line lexical database for Hindi language
- Design has been inspired by the famous English WordNet
- Unique features
    - Graded antonyms and meronymy relationships
    - Efficient underlying database design
    - Cross part of speech linkage

[30] https://www.cse.iitb.ac.in/~pb/papers/lrec2010-indowordnet.pdf

Word Sense Disambiguation Multilinguality[31]

Ambiguity is inherent to human language. In particular, word sense ambiguity is prevalent in all natural languages, with a large number of the words in any given language carrying more than one meaning. For instance, the English noun plant can mean green plant or factory; similarly the French word feuille can mean leaf or paper. The correct sense of an ambiguous word can be selected based on the context where it occurs, and correspondingly the problem of word sense disambiguation is defined as the task of automatically assigning the most appropriate meaning to a polysemous word within a given context. Despite the large number of word sense disambiguation methods that have been proposed so far, targeting the resolution of word ambiguity in different languages, there are only a few methods that try to explore more than one language at a time.

Approaches:

1) Bootstrapping method (Li and Li, 2002)

It iterates between two languages to select the correct translation for a given target word. Word translations are automatically disambiguated using information iteratively drawn from two languages.

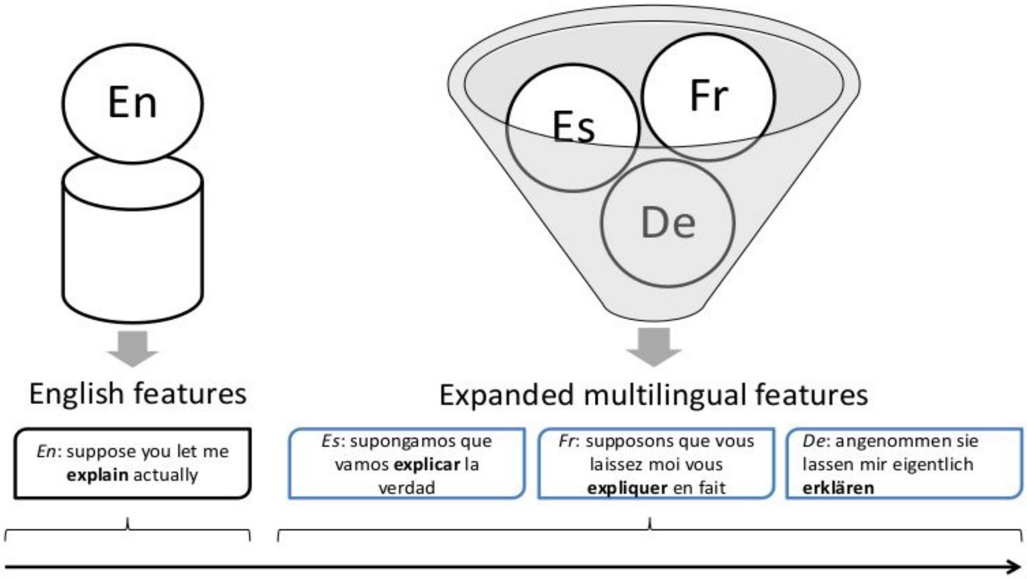
2) Word Sense Disambiguation with Multilingual Features (Banea and Mihalcea, 2011)

This work seeks to explore the expansion of a monolingual feature set with features drawn from multiple languages in order to generate a more robust and more effective vector-space representation that can be used for the task of word sense disambiguation. While traditional monolingual representations allow a supervised learning systems to achieve a certain accuracy, this work tries to surpass this limitation by infusing additional information in the model, mainly in the form of features extracted from the machine translated view of the monolingual data. A statistical machine translation (MT) engine does not only provide a dictionary-based translation of the words surrounding a given ambiguous word, but it also encodes the translation knowledge derived from very large parallel corpora, thus accounting for the contextual dependencies between the words.

Example: Consider the word **build** translated to language French. Build can have up to 12 different meanings. For such words, multilingual representation attempts to disambiguate the target ambiguous word by assigning it a different translation depending on the context where it occurs. Eg- build can be used for construction or as a verbal expression (build support) depending on the context. Being able to infer from the co-occurrence of additional words appearing the context, the MT engine differentiates the two usages in French, translating the first occurrence as acquis and the second one as accumuler.

The multilingual representation also significantly enriches the feature space, by adding features drawn from multiple languages. While this multilingual representation can sometime result in redundancy when there is a one-to-one translation between languages, in most cases however the translations will enrich the feature space, by either indicating that two features in English share the same meaning (e.g., the words manufactory and factory will both be translated as usine in French), or by disambiguating ambiguous English features using different translations

[31] http://web.eecs.umich.edu/~mihalcea/papers/banea.iwcs11.pdf


En
Es
Fr
De
English features
Expanded multilingual features
En: suppose you let me **explain** actually
Es: supongamos que vamos **explicar** la verdad
Fr: supposons que vous laissez moi vous **expliquer** en fait
De: angenommen sie lassen mir eigentlich **erklären**

Unit 6 - Multidisciplinary Natural Language Processing

Table of Contents

Sentiment Analysis[32]

Using NLP, statistics, or machine learning methods to extract, identify, or otherwise characterize the sentiment content of a text unit.

The process of identifying the orientation of opinion in a piece of text.

In simple terms to understand its polarity - whether it is positive, negative or neutral.

Example - “The movie was fabulous.” - Positive - :-)
- “The movie stars Mr. X.” - Neutral - :-|
- “The movie was horrible!” - Negative - :-(

Sometimes called opinion mining or emotion mining.

SA Levels

Example - 'The camera takes great quality pictures but is expensive. It feels like a professional one'

1. Word-level SA

Called Attribute-level SA.

Provides a sentiment for each object in a sentence. Attribute analysis identifies the objects of a sentence and any sentiment expressed regarding those objects.

Example - Sentiment expressed with regard to "camera" and "quality pictures." -

```
Sentence |    attribute          | sentiment_score
-------------+-----------------------+-----------------
         1 | camera              |            1
         1 | quality pictures  |            1
```

2. Sentence-level SA

Provides the overall sentiment of each sentence in a document. If a sentence is contains both positive and negative sentiments, it may appear as mixed or neutral.

Example - Shows two sentences, the first of which is neutral. As a mixed sentiment, the sentiment score is 0, or neutral, and the mixed value is true. The second sentence is entirely positive. Its sentiment is 1, or positive, and the mixed value is false.

```
sentence | sentiment_score | mixed
-------------+-----------------------+-------
        1 |                       0 | true
        2 |                       1 | false
```

3. Document-level SA

Provides the overall sentiment of an entire document. If you wanted to know if a movie review was positive, negative, or

[32] Pushpak Bhattacharya - IIT Bombay

mixed, a document level analysis could provide that information. Document level analysis gives both the overall sentiment score and a mixed rating if the sentiment is not exclusively positive or negative.

Example - Shows that overall, the writer is positive but does express some negative sentiments.

```
sentiment_score | mixed
-----------------------+----------
                     1 | true
```

Challenges in SA
1. Word Sense Disambiguation
Correct meaning of word based on the context needs to be extracted as word can have different meanings for different domains. For example small size can be positive opinion for mobile phones but negative for hotels.
2. Sarcasm Detection
That's just what I need, great! Terrific! In a movie review with a one star.
3. Comparisons
To determine the polarity for comparative sentences can be a challenge. For example Battery life of phone X is better than phone Y. This review has positive word 'better' but the author's preferred object is not easy to determine which is the key piece of information in a comparative reviews.
4. Handling negations
Negations if not handled properly can give completely wrong results. For example There is a good chance that this phone will not break easily. This review shows positive polarity but presence of negation changes the effect completely.

Text Entailment[33] [34]

It's a concept that corresponds to "common sense" reasoning. It is defined as a relation between two natural language sentences (a premise P and a hypothesis H) that holds if a human reading P would infer that H is most likely true.

A text T is said to entail a textual hypothesis H if the truth of H can be inferred from T.

Textual entailment (TE) is a directional relation between text fragments. The relation holds whenever the truth of one text fragment follows from another text. In the TE framework, the entailing and entailed texts are termed text (t) and hypothesis (h), respectively.
It is defined as - "t entails h"(t —> h) if, typically, a human reading t would infer that h is most likely true.

The relation is directional because even if "t entails h", the reverse "h entails t"is much less certain.
Example - T : Green cards are becoming more difficult to obtain.
H : Green card is now difficult to receive.
Entailment : YES

How one can say that the hypothesis is entailed by the given text? There are various approaches by which one can determine the result of entailment.

Triggers

Entailment triggers help to decide entailment between premise and hypothesis. These triggers can be synonyms (T: India won the world cup in 2011. H: India got the world cup in 2011.), hypernyms (T: Ram ate breads. H: Ram ate food.), hyponyms (T: He is interested in a game. H: He is interested in cricket.), quantifiers (T: Every employee must file income tax return. H: An employee must file income tax return.), coreference (T: Michael Dell announced a new strategy for the company. He is the founder of Dell. H: Michael Dell is the founder of Dell.), etc.

Approaches to detect entailment

1. Bag of Words
2. Natural Logic

[33] https://web.stanford.edu/~jurafsky/wmt09_pado.pdf
[34] http://www.cfilt.iitb.ac.in/resources/surveys/te-mt_shubham-june14.pdf

3. Probabilistic Entailment Model
4. Lexical Entailment Model
5. Machine Learning Approach
6. Graph Matching

Applications

Question Answering (QA), Information Extraction (IE), (multi-document) summarization and machine translation (MT) evaluation, need to recognize that a particular target meaning can be inferred from different text variants.

Robust and Scalable Machine Translation[35]

Robustness in handling heterogeneous data need to be evaluated on data from several different sources. Scalability in handling a large amount of data needs to be evaluated for speed and complexity.

Robustness -

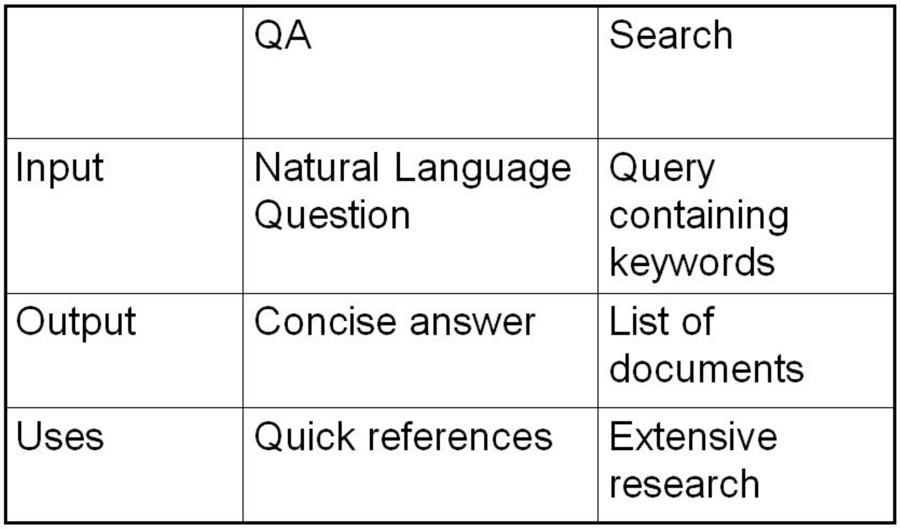
1. POS tagger: by building a generalized model.
2. Dependency parser: by bootstrapping parse information.
3. Semantic role labeler: by applying higher-order argument pruning.

Scalability -
1. POS tagger: by adapting dynamic model selection.
2. Dependency parser: by optimizing the engineering of transition- based parsing algorithms.
3. Semantic role labeler: by applying conditional higher-order argument pruning.

Question Answering in Multilingual Setting

Type of information retrieval. Given a collection of documents ,the system should be able to retrieve answers to questions posed in natural language.

Difference between search and QA -



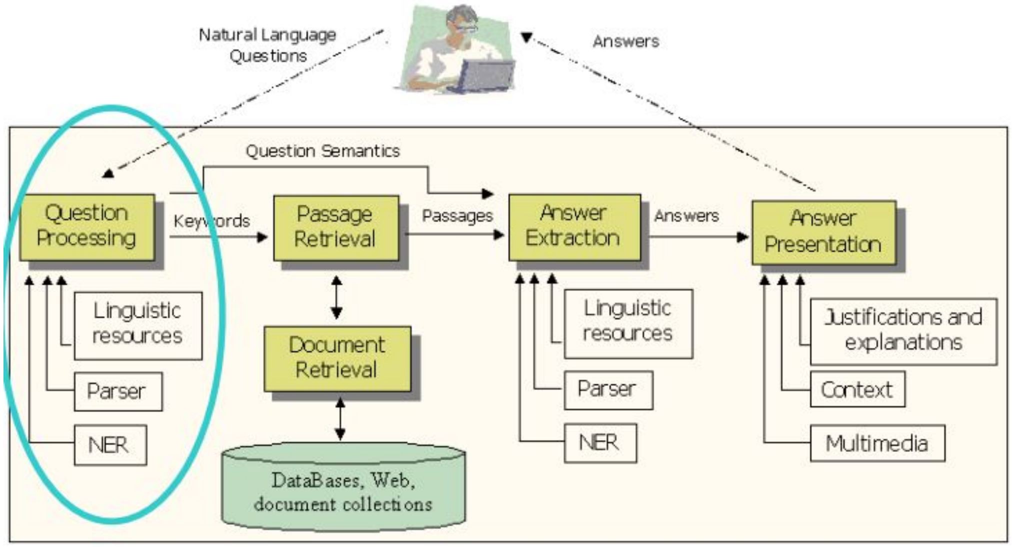
	QA	Search
Input	Natural Language Question	Query containing keywords
Output	Concise answer	List of documents
Uses	Quick references	Extensive research

Aim is to answer natural language questions from any type of document.
- Domain: {Open domain: newspaper articles (close corpora), the web} {Close domain: Aerospatiale domain, medical domain, etc.}
- Documents: {Structured -> Database QA} {Heterogeneous Plain text (web pages, multimedia documents) containing errors, contradictions, inconsistencies -> Text based QA} {Web based QA}
- NL questions : {Factoid questions : who, when, where, how much, ... }{Definition, How, why questions}

[35] Optimization of NLP Components For Robustness and Scalability Dissertation Defense, Department of Computer Science University of Colorado Boulder

{Opinion, comparative and evaluative questions }
- Answers {Exact answers, Passages, Multimedia, etc.}

Architecture Components



Question Processing captures the semantic of the question and selects keywords forPassage Retrieval.
Passage Retrieval extracts and ranks passages.
Answer Extraction extracts and ranks answers using NLP techniques.

T-expression (Ternery Expression)
<subject, relation, object>
Eg -
Sentence: “Bill surprised Hillary with his answer”
T-expression: <<Bill surprise Hillary> with answer>
<answer related-to Bill>

Solution by T-expression
Q: Whom did Bill surprise with his answer?
Step1: Bill surprised whom with his answer
Step2: <<Bill surprise whom> with answer>
<answer related-to Bill>
Step3: It matches to the earlier T-expresion
Step4: produce the answer
A: Bill surprised Hillary with his answer.

Evaluation -
1. TREC (Text Retrieval conference)
2. FIRE (Forum for Information Retrieval Evaluation)
3. CLEF (Cross Language Evaluation Forum)

Cross Lingual Information Retrieval[36]

Information Retrieval (IR) refers to the task of searching relevant documents and information from the contents of a data set such as the World Wide Web (WWW). A web search engine is an IR system that is designed to search for information on the World Wide Web.
Component of IR system -

[36] Cross-lingual Information
Retrieval." Electronic Journal of Computer Science and Information Technology (eJCSIT) 2 (2010

Crawling: Documents from web are fetched and stored.
Indexing: An index of the fetched documents is created.
Query: Input from the user.
Ranking: The systems produces a list of documents, ranked according to their relevance to the query.

Information on the web is growing in various forms and languages. Though English dominated the web initially, now less than half the documents on the web are in English. The popularity of internet and availability of networked information sources have led to a strong demand for Cross Lingual Information Retrieval (CLIR) systems.

Cross-Lingual Information Retrieval (CLIR) refers to the retrieval of documents that are in a language different from the one in which the query is expressed. This allows users to search document collections in multiple languages and retrieve relevant information in a form that is useful to them, even when they have little or no linguistic competence in the target languages. Cross lingual information retrieval is important for countries like India where very large fraction of people are not conversant with English and thus don't have access to the vast store of information on the web.

Approaches to CLIR

1. Query translation approach
   The query is translated into the language of the document. Many translation schemes could be possible like dictionary based translation or more sophisticated machine translations. The dictionary based approach uses a lexical resource like bi-lingual dictionary to translate words from source language to target document language. This translation can be done at word level or phrase level. The main assumption in this approach is that user can read and understand documents in target language. In case, the user is not conversant with the target language, he/she can use some external tools to translate the document in foreign language to his/her native language. Such tools need not be available for all language pairs.
2. Document translation approach
   It translates the documents in foreign languages to the query language. Although this approach alleviates the problem stated above, this approach has scalability issues. There are too many documents to be translated and each document is quite large as compared to a query. This makes the approach practically unsuitable.
3. Interlingua based approach
   The documents and the query are both translated into some common Interlingua (like UNL - Universal Networking Language). This approach generally requires huge resources as the translation needs to be done online.

A possible solution to overcome the problems in query and document translations is to use query translation followed by snippet translation instead of document translation. A snippet generally contains parts of a document containing query terms. This can give a clue to the end user about usability of document. If the user finds it useful, then document translation can be used to translate the document in language of the user.

Challenges in CLIR

1. Translation ambiguity
   While translating from source language to target language, more than one translation may be possible. Selecting appropriate translation is a challenge.
   For example, the word मान (maan, respect/neck) has two meanings neck and respect.
2. Phrase identification and translation
   Identifying phrases in limited context and translating them as a whole entity rather than individual word translation is difficult.
3. Translate/transliterate a term
   There are ambiguous names which need to be transliterated instead of translation.
   For example, भास्कर (Bhaskar, Sun) in Marathi refers to a person's name as well as sun. Detecting these cases based on available context is a challenge.

4. Transliteration errors
   Errors while transliteration might end up fetching the wrong word in target language.
5. Dictionary coverage
   For translations using bilingual dictionary, the exhaustiveness of the dictionary is important criteria for performance on system.
6. Font
   Many documents on web are not in Unicode format. These documents need to be converted in Unicode format for further processing and storage.
7. Morphological analysis (different for different languages)
8. Out-of-Vocabulary (OOV) problems
   New words get added to language which may not be recognized by the system.

Factors affecting the performance of CLIR systems

1. Limited size of Dictionary
   New words, compounds and phrases get added to the language quite frequently and maintaining the dictionary up to date with these new words and phrases is difficult. Thus morphological analysis becomes essential.
2. Query translation/transliteration performance
   During the translation process, extraneous senses may be added to the query due to the fact that the translation alternatives may also have more than one sense. Thus lexical ambiguity appears in both source and target language.

Natural Language ToolKit (NLTK)[37] [38]

Natural Language ToolKit (NLTK) is a comprehensive Python library for natural language processing and text analytics. The following is a description of using NLTK for common NLP tasks.

i) Tokenization

- The sent_tokenize function uses an instance of PunktSentenceTokenizer from the nltk.tokenize.punkt module.

```
>>> para = "Hello World. It's good to see you. Thanks for buying this book."

>>> from nltk.tokenize import sent_tokenize

>>> sent_tokenize(para)

['Hello World.', "It's good to see you.", 'Thanks for buying this book.']
```

- Splitting sentence into individual words can be done with the word_tokenize() function

```
>>> from nltk.tokenize import word_tokenize

>>> word_tokenize('Hello World.')

['Hello', 'World', '.']
```

ii) Filtering stop words

[37] http://www.nltk.org/
[38] Python Text Processing with NLTK 2.0 Cookbook Jacob Perkins

Stopwords are common words that generally do not contribute to the meaning of a sentence, at least for the purposes of information retrieval and natural language processing. NLTK comes with a stopwords corpus that contains word lists for many languages.

```
>>> from nltk.corpus import stopwords

>>> english_stops = set(stopwords.words('english'))

>>> words = ["Can't", 'is', 'a', 'contraction']

>>> [word for word in words if word not in english_stops]

["Can't", 'contraction']
```

iii) Synsets(synonymous words) of a word using WordNet

```
>>> from nltk.corpus import wordnet

>>> syn = wordnet.synsets('cookbook')[0]

#wordnet.synset(word) returns a list of synonymous words of the word supplied to the function

>>>syn.pos()    # retrieve pos tag of cookbook in this case

‘n’
```

- Lammas (canonical or morphological form of a word) can be looked up with syn.lemmas()

iv) Stemming words (removing affixes from a wrod, resulting with a stem)

- This can be done using an instance of the Porter Stemming Algorithm.

```
    >>> from nltk.stem import PorterStemmer

    >>> stemmer = PorterStemmer()

    >>> stemmer.stem('cooking')

            'cook'

    >>> stemmer.stem('cookery')

            'cookeri'
```

v) POS tagging

Part-of-speech tagging is the process of converting a sentence, in the form of a list of words, into a list of tuples, where each tuple is of the form (word, tag). The tag is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

- Default tagging

Default tagging provides a baseline for part-of-speech tagging. It simply assigns the same part-of-speech tag to every token.

```
    >>> from nltk.tag import DefaultTagger

    >>> tagger = DefaultTagger('NN')
```

```
>>> tagger.tag(['Hello', 'World'])

[('Hello', 'NN'), ('World', 'NN')]
```

- Unigram Tagger

A unigram generally refers to a single token. Therefore, a unigram tagger only uses a single word as its context for determining the part-of-speech tag.

```
>>> from nltk.tag import UnigramTagger

>>> from nltk.corpus import treebank

>>> train_sents = treebank.tagged_sents()[:3000]

>>> tagger = UnigramTagger(train_sents) #training a unigram tagger

>>> treebank.sents()[0]

['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', 'will', 'join','the', 'board', 'as', 'a', 'nonexecutive','director',
'Nov.', '29','.']

>>> tagger.tag(treebank.sents()[0])

[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ('61', 'CD'),('years', 'NNS'), ('old', 'JJ'), (',', ','), ('will', 'MD'),
('join','VB'), ('the', 'DT'), ('board', 'NN'), ('as', 'IN'), ('a', 'DT'),('nonexecutive', 'JJ'), ('director', 'NN'),
('Nov.', 'NNP'), ('29','CD'), ('.', '.')]
```

Applications

Machine Translation

[39]Machine Translation (MT) is the task of automatically converting one natural language into another, preserving the meaning of the input text, and producing fluent text in the output language.
Oldest sub-field of Artificial Intelligence.
Challenges in MT -
1. Word order
English word order is subject-verb-object whereas Japanese order is subject-object-verb.
Eg - English: IBM bought Lotus and Japanese: IBM Lotus bought

2. Word sense
Different word senses will likely translate into different words in another language.
Eg - Bank as in river and Bank as in financial institution

3. Pronouns
Japanese is an example of a pro-drop language.
Eg - Shiranai. Ki ni itta? know-NEGATIVE. liked? I don't know. Do you like it?
Spanish drops pronouns altogether.
Eg - -o = I, -as = you, -a = he/she/it, -amos = we, -an = they

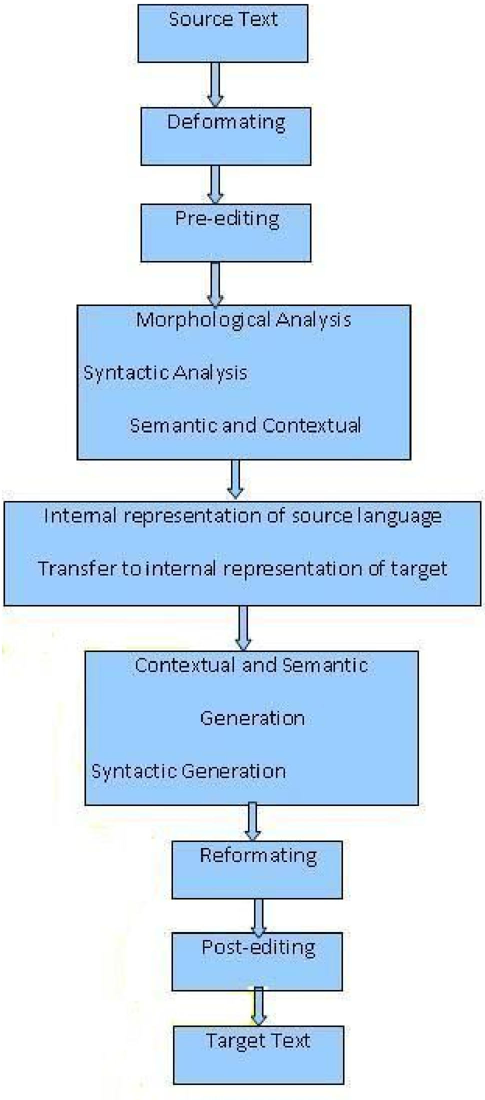
4. Tense
Eg - Spanish has two versions of the past tense: one for a definite time in the past, and one for an unknown

[39] https://nlp.stanford.edu/projects/mt.shtml

time in the past. When translating from English to Spanish we need to choose which version of the past tense to use

5. Idioms
Eg - “to kick the bucket” means “to die”

Process of MT -



Types of MT -[40]
1. Rule-based MT
Consists of collection of rules called grammar rules, lexicon and software programs to process the rules. It is the first strategy ever developed in the field of machine translation. Rules are written with linguistic knowledge gathered from linguists that is manually written. Rules play major role in syntactic processing, semantic interpretation, and contextual processing of language.

Eg of rules -
S -> NP VP
VP -> V NP
NP -> Name
NP -> ART N
Where S stands for sentence, V for verb, N for noun and ART for article.

[40] Speech and Language Processing, Jurafsky and Martin, ch. 21

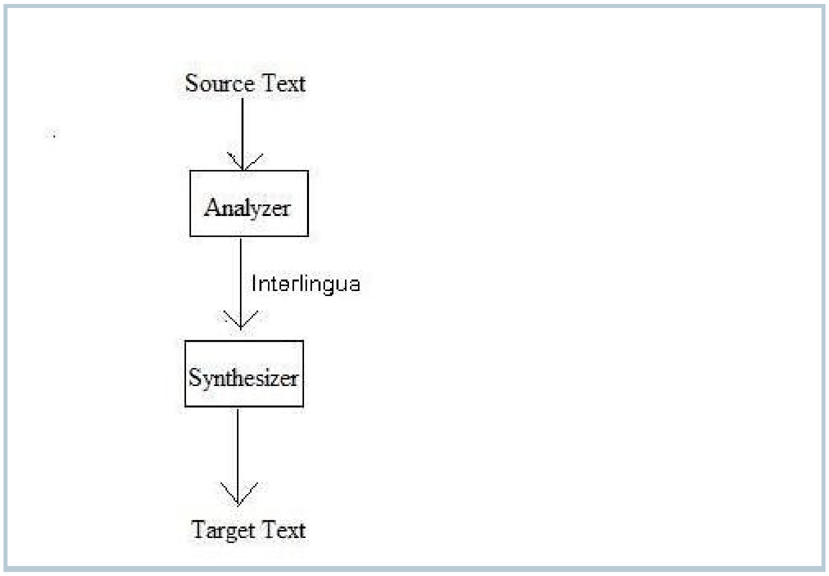
Advantage -
It can deeply analyze at syntax and semantic levels.
Disadvantage -
Requirement of huge linguistic knowledge and very large number of rules to cover all the features of a language.

2. Example-based MT
The basic idea of **Example-Based Machine Translation** (EBMT) is to reuse examples of already existing translations as the basis for for new translation. The process of **EBMT** is broken down into three stages:
2.1 Matching
Finds examples that are going to contribute to the translation on the basis of their similarity with the input. The way matching stage should be implemented is based on how the examples are stored. Ex - tress, sequence comparison.

2.2 Alignment
Identify which parts of the corresponding translation are to be reused. Alignment is done by using bilingual dictionary or comparing with other examples.

2.3 Recombination
Makes sure that the reusable parts in example identified during alignement are putting together in a legitimate way.

3. Interlingual MT
Interlingua is an intermediate form between two or more languages.
Interlingual Machine Translation is a methodology that employs interlingua for translation. The disadvantage is that the design of interlingua is too complex.



A typical interlingual MT system has analyzer and synthesizer for each language. The analyzer produces interlingual represention of the meaning of the given text. The synthesizer produces one or more sentences with the meaning given by the analyzer.

Example - KANT

4. Statistical MT
Automatically align words and phrases within sentence pairs in a parallel corpus using probabilities. Find the most probable English sentence given a foreign language sentence.

$\hat{}e = \arg\max_e p(e|f)$
$= \arg\max_e p(f|e)p(e)\ p(f)$
$= \arg\max_e p(f|e)p(e)$

$p(f|e)$ is the translation model
– assigns a higher probability to English sentences that have the same meaning as the foreign sentence.
– needs a bilingual (parallel) corpus for estimation.
$p(e)$ is the language model
– assigns a higher probability to fluent/grammatical sentences.
– only needs a monolingual corpus for estimation (which are plentiful).
$p(f|e)$ - the probability of some foreign language string given a hypothesis English translation

Eg - f = Ces gens ont grandi, vecu et oeuvre des dizaines d'annees dans le domaine agricole.
e = Those people have grown up, lived and worked many years in a farming district.
e = I like bungee jumping off high bridges.

Database Interface[41]
The natural language to database interface shifts a user's burden of learning a structured database language to describe his or her need for information to the system.
Retrieval of information from the databases requires the knowledge of databases language like the Structured Query Language (SQL). But humans understand general langugage and it is not possible for everyone to enter SQL to retrieve the information they want. The idea of using natural language instead of SQL has led to the development of new type of processing method called Natural Language Interface to Database systems.

Methods of NLDBI -
1. Symbolic Approach (Rule-based)
2. Empirical Approach (Corpus-based)

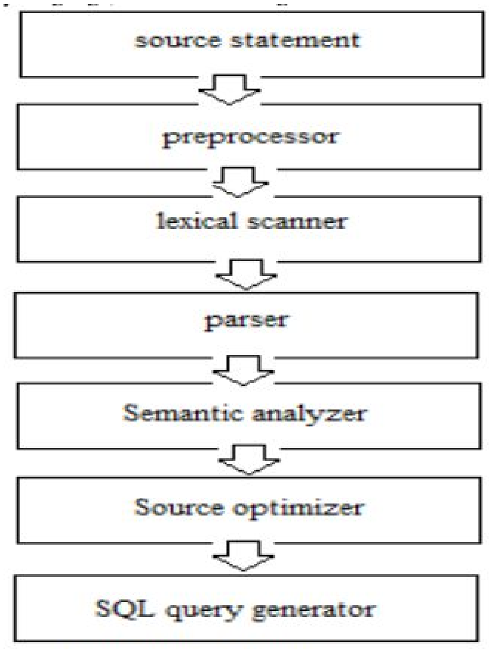


Fig 1: System Architecture

Web Mining[42] [43]
Web is a collection of inter-related files on one or more Web servers. Web mining is the application of data mining techniques to extract knowledge from web data.

[41] https://www.ijarcce.com/upload/2014/february/IJARCCE5A____s_aarti_natural.pdf
[42] http://dmr.cs.umn.edu/Papers/P2004_4.pdf
[43] http://www.ieee.org.ar/downloads/Srivastava-tut-pres.pdf

Web data is - web content - text, image, records
- web structure - hyperlinks, tags
- web usage - http logs, app server logs.

Divided into 3 types -

1. Web usage mining
   Discovering interesting usage patterns from Web data in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a website.

   Classified based on what is the usage data -
   1.1 Web Server Data
   User logs are collected by the web server and typically include IP address, page reference and access time.
   1.2 Application Server Data
   Commercial application servers have features to enable E-commerce applications to be built on top of them. A key feature is the ability to track various kinds of business events and log them in application server logs.
   1.3 Application Level Data
   New kinds of events can be defined in an application, and logging can be turned on for them — generating histories of these events.

   2. Web content mining
   Web content mining is the process of extracting useful information from the contents of web documents. Content data is the collection of facts a web page is designed to contain. It may consist of text, images, audio, video, or structured records such as lists and tables.

   3. Web structure mining
   The structure of a typical web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Web structure mining is the process of discovering structure information from the web. It uses graph theory to analyze the node and connection structure of a web site. Web structure mining can be divided into two kinds:
1. Patterns from Hyperlinks
   A hyperlink is a structural component that connects the web page to a different location either on the same page or different page.
2. Document
   Analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

Text Mining[44]

Text mining is the process of analyzing collections of textual materials in order to capture key concepts and themes and uncover hidden relationships and trends without requiring that you know the precise words or terms that authors have used to express those concepts.

Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The overarching goal is, essentially, to turn text into data for analysis, via application of NLP and analytical methods.

[44] Kao, A., and Poteet, S. (Editors). *Natural Language Processing and Text Mining*. Springer. ISBN 1-84628-175-X

A typical application is to scan a set of documents written in a natural language and either model the document set for predictive classification purposes or populate a database or search index with the information extracted.

Applications -
1. Social media monitoring
2. Tracking terrorist activities
3. Business and marketing applications