UNIT-3

Natural Language Processing

- Language Models
  - N-gram character models
  - N-gram word models
- Text classification
  - Classification by data compression
- Information retrieval
  - The page rank algorithm
  - The HITS algorithm
- Information extraction
  - Finite state automata for information extraction
  - Probabilistic model for information extraction
- Examples: Applications of Natural Language Processing.
- Case Study: Automated Voice Assistants, Chat bots.

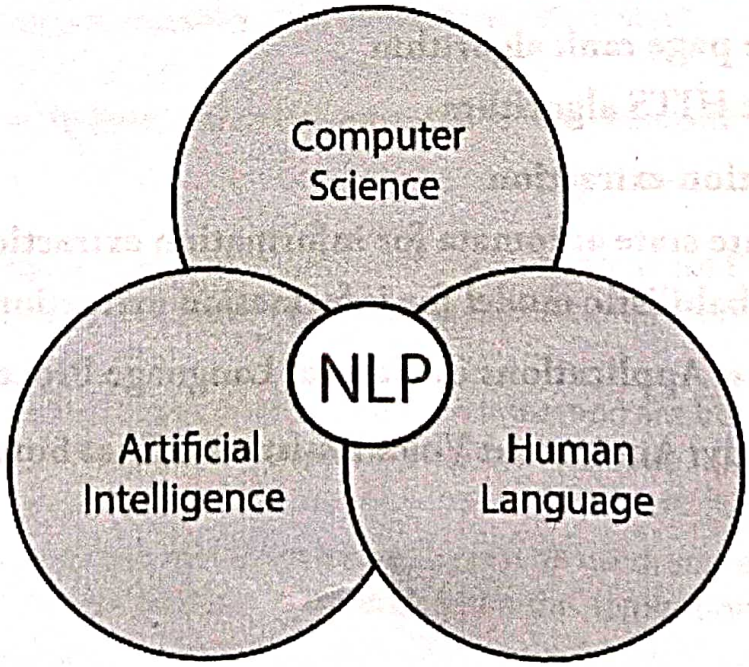
Unit –3 Natural Language Processing

❖ Introduction:

NLP: the ability to understand and process human languages. It is important in order to fill the gap in communication between humans and machines.

Natural Language Processing (NLP), by definition, is a method that enables the communication of humans with computers or rather a computer program by using human languages, referred to as natural languages, like English. These include both text and speech input. It helps computers to understand and interpret the languages and reply validly in a valid manner.

It is a part of **Computer Science, Human language,** and **Artificial Intelligence**. It is the technology that is used by machines to understand, analyze, manipulate, and interpret human's languages. It helps developers to organize knowledge for performing tasks such as **translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction,** and **topic segmentation**.It is used to create automated software that helps understand human spoken languages to extract useful information it gets within the style of audio. Techniques in NLP allow computer systems to process and interpret data in the form of natural languages.



Natural language processing (NLP) drawbacksare often divided into two tasks:

- Processing written text, using lexical, syntactic and semantic information of th language as well as the required real world information.

- Processing spoken language, using all the information required higher than and extra knowledge about phonology as well as enough added information to handle the additional ambiguities that arise in speech.
- NLP is one of the major and most addressed parts of Artificial Intelligence. It is very commonly used in our day-to-day lives, in applications like Google Assistant, Siri, Google Translate, Alexa, etc. NLP includes dividing the input into smaller items and playing tasks to understand the connection between them and then produce significant output.

3.1 Language Models:

Formal languages, such as the programming languages Java or Python have exactly defined language models. A **language** can be defined as a collection of strings; "print (2 + 2)" is a legal program in the language Python, whereas "2) + (2 print" is not. Since there are an infinite number of legal programs, they cannot be enumerated; instead they are such that by a collection of rules called a **grammar**. Formal languages also have rules that define the meaning or **semantics** of a program; as an example, the rules say that the "meaning" of "2 + 2" is 4, and the meaning of "1/0" is that an error is signaled.

Natural languages are also **ambiguous**. We cannot speak of a single meaning for a sentence, but rather of a probability distribution over possible meanings.

3.1.1 N-gram character models:

An n-gram model is **a technique of counting sequences of characters or words that allows us to support rich pattern discovery in text**. In other words, it tries to capture patterns of sequences (characters or words next to each other) while being sensitive to contextual relations (characters or words near each other).

Ultimately, a written text is consist of **characters**—letters, digits, punctuation, and spaces in English (and additional exotic characters in some other languages). Thus, one of the simplest language models is a probability distribution over sequences of characters. We have tendency to write P (c1:N) for the probability of a sequence of N characters, c1 through cN.

A sequence of written symbols of length n is called an n-gram, with special case "unigram" for 1-gram, "bigram" for 2-gram, and "trigram" for 3-gram. A model of the probability distribution of n-letter sequences is thus called an **n-gram model**.

An n-gram model is defined as a **Markov chain** of order n − 1.

$$P(c_i \mid c_{1:i-1}) = P(c_i \mid c_{i-2:i-1})$$

We can define the probability of a sequence of characters P (c1: N) under the trigram model by first factoring with the chain rule and then using the Markov assumption:

$$P(c_{1:N}) = \prod_{i=1}^{N} P(c_i \mid c_{1:i-1}) = \prod_{i=1}^{N} P(c_i \mid c_{i-2:i-1}) .$$

For a trigram character model in a language with 100 characters, **P** (Ci|Ci−2:i−1) has a million entries, and can be accurately estimated by counting character sequences in a body of text of 10 million characters or more. We call a body of text a **corpus.**

What can we do with n-gram character models? One task for which they are well suited is **language identification**: given a text, determine what natural language it is written in. This is a relatively easy task; even with short texts such as "Hello, world" or "Wiegehtesdir," it is easy to identify the first as English and the second as German. Computer systems identify languages with greater than 99% accuracy; sometimes, closely related languages, such as Swedish and Norwegian are confused.

Other tasks for character models include spelling correction, genre classification, and named-entity recognition. **Genre classification** means deciding if a text is a news story, alegal document, a scientific article, etc. While many features help make this classification,counts of punctuation and other character n-gram features go a long way. Named-entity recognition is the task of finding names of things in a document and deciding what class they belong to.

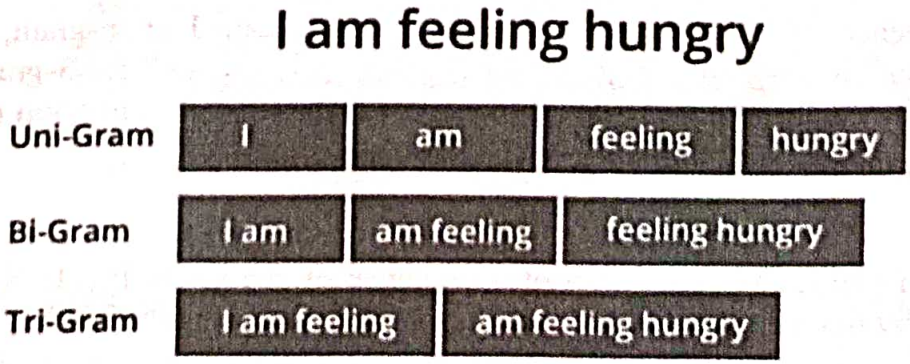
For example, in the text "Mr. Sopersteen was prescribedaciphex," we should recognize that "Mr. Sopersteen" is the name of a person and "aciphex" isthe name of a drug. Character-level models are good for this task because they can associatethe character sequence "ex" ("ex" followed by a space) with a drug name and "Steen" witha person name, and thereby identify words that they have never seen before.

3.1.2 N-gram word models:

When we have a tendency to analyze a sentence one word at a time, then it is referred to as a unigram. The sentence parsed two words at a time isa bigram.

When the sentence is parsed three words at a time, then it is a trigram. Similarly, n-gram refers to the parsing of *n* words at a time.

Example: To understand unigrams, bigrams, and trigrams, you can refer to the below diagram:



Therefore, parsing allows machines to understand the individual meaning of a word in a sentence. Also, this type of parsing helps predict the next word and correct spelling errors.

Consider two sentences: "There was heavy rain" vs. "There was heavy flood". From experience, we all know that the previous sentence sounds better. An N-gram model will tell us that "heavy rain" occurs much more often than "heavy flood" within the training corpus. Thus, the first sentence is more probable and can be elected by the model.

A model that simply depends on however a word occurs without looking at previous words is called **unigram**. If a model considers only the previous word to predict the current word, then it's called **bigram**. If two previous words are considered, then it's a **trigram** model.

An n-gram model for the above example would calculate the following probability:

P('There was heavy rain') = P('There', 'was', 'heavy', 'rain') =
P('There')P('was'|'There')P('heavy'|'There was')P('rain'|'There was heavy')

Since it's impractical to calculate these conditional probabilities, using *Markov assumption*, we approximate this to a bigram model:

P('There was heavy rain') ~ P('There')P('was'|'There')P('heavy'|'was')P('rain'|'heavy')

Now we turn to n-gram models over words instead of characters. All the same mechanismapplies equally to word and character models. The main difference is that the **vocabulary**the set of symbols that make up the corpus and the modelis larger. There are only about100 characters in most languages, and sometimes we build character models that are evenmore restrictive, for example by treating "A" and "a" as the same symbol or by treating allpunctuation as the same symbol. But with word models we have at least tens of thousands ofsymbols, and sometimes millions. The wide range is because it is not clear what constitutes aword. In English a sequence of letters enclosed by spaces is a word, but in some languages,like Chinese, words are not separated by spaces, and even in English many decisions must bemade to have a clear policy on word boundaries: how many words are in "ne'er-do-well"? Orin "(Tel:1-800-960-5660x123)"?

3.2 Text classification:

Text classification is the process of classifying documents into predefined categories based on their content. It is the machine controlled (automated) assignment of natural language texts to predefinedclasses (or categories). Text classification is the primary requirement of text retrieval systems thatretrieve texts in response to a user query, and text understanding systems, which transform text in some way such as producing summaries, questions-answers or extract data.

We now consider in depth the task of **text classification**, also known as **categorization**: given a text of some kind, decide that of a predefined set of categories it belongs to.

Text classification is that the method of (process of) classifying documents into predefined categories by their content.

Language identification and genre classification are examples of text classification, as issentiment analysis(classifying a movie or product review as positive or negative) and **spam detection** (classifyingan email message as spam or not-spam). Since "not-spam" is awkward, researchers havecontained the term **ham** for not-spam. We can treat spam detection as a problem in supervisedlearning. A training set is readily available: the positive (spam) examples are in my spamfolder, the negative (ham) examples are in my inbox. Here is an excerpt:

Spam: Wholesale Fashion Watches -57% today. Designer watches for cheap ...

Spam: You can buy ViagraFr$1.85 All Medications at unbeatable prices! ...

Spam: WE CAN TREAT ANYTHING YOU SUFFER FROM JUST TRUST US...

Spam: Sta.rt earn*ing the salary yo,u d-eserve by o'btaining the prope,rcrede'ntials!

Ham: The practical significance of hyper tree width in identifying more...

Ham: Abstract: We will motivate the problem of social identity clustering: ...

Ham: Good to see you my friend. Hey Peter, It was good to hear from you. ...

Ham: PDS implies convexity of the resulting optimization problem (Kernel Ridge ...

From this section(part) we can start to get an idea of what might be good features to include inthe supervised learning model. Word n-grams such as "for cheap" and "You can buy" seemto be indicators of spam (although they would have a nonzero probability in ham as well).

Character-level features also seem important: spam is more likely to be all uppercase and tohave punctuation embedded in words. Apparently the spammers thought that the word bigram"you deserve" would be too indicative of spam, and thus wrote "yo,u d-eserve" instead. Acharacter model should detect this. We could either create a full character n-gram modelof spam and ham, or we could handcraft features such as "number of punctuation marksembedded in words."

Note that we have two complementary ways in which of talking about classification. In thelanguage-modeling approach, we define one n-gram language model for **P**(Message | spam)by training on the spam folder, and one model for **P**(Message | ham) by training on the inbox.

Then we can classify a new message with an application of Bayes' rule:

$$\underset{c \in \{spam, ham\}}{\operatorname{argmax}} \; P(c \mid message) = \underset{c \in \{spam, ham\}}{\operatorname{argmax}} \; P(message \mid c)\, P(c) \,.$$

Where,P(c) is estimated just by counting the total number of spam and ham messages. Thisapproach works well for spam detection,just as it did for language identification.

The data can accurately determine if it is good or not. It is necessary to constantlyupdate features, because spam detection is an **adversarial task**; the spammers modify theirspam in response to the spam detector's changes. It can be expensive to run algorithms on a very large feature vector, so often a processof **feature selection** is used to keep only the features that best discriminate between spam andham.

3.2.1 Classification by data compression:

Another way to think about classification is as a problem in **data compression**. A losslessCompression algorithm takes a sequence of symbols, detects repeated patterns in it, and writesa description of the sequence that is lot of compact than the original. For example, the text"0.1428571428571428571" might be compressed to "0. [142857]*3." Compression algorithmswork by dictionaries of subsequences of the text, and then referring to entries within thedictionary. The example here had only one dictionary entry, "142857."

In effect, compression algorithms are creating a language model. To do classificationby compression, first collect all the spam training messages and compress them as aunit. We do the same for the ham. Then when given a new message to classify, we appendit to the spam messages and compress the result. We also append it to the ham and compressthat. Whichever class compresses betteradds the fewer number of additional bytes for thenewmessageis the predicted class. The idea is that a spam message will tend to share dictionary entries with different spam messages and therefore can compress better when appended toa collection that already contains the spam dictionary.

Experiments with compression-based classification on number of the standard corpora fortext classification the 20-Newsgroups data set, the Reuters-10 Corpora, the Industry Sectorcorporaindicate that whereas running off-the-shelf compression algorithms like gzip, RAR can be quite slow; their accuracy is comparable to traditional classification algorithms.

This is interesting in its own right, and also serves to point out that there's promisefor algorithms that use character n-grams directly with no preprocessing of the text or featureselection: they appear to be capturing some real patterns.

3.3 Information retrieval :

Information retrieval is concerned with representing, searching, and manipulating large collections of electronic text and alternative human-language information.

Information retrieval systems use a very simple language model based on **bags ofwords**, yet still manage to perform well in terms of **recall** and **precision**(exactness)on very largecorpora of text. On internet corpora, link-analysis algorithms improve performance.

Information retrieval is the task of finding documents that are relevant to a user's need forinformation. The well-known examples of information retrieval systems are search enginesontheWorldWideWeb. A Web user can type a query such as: AI bookinto a search engineand see a list of relevant pages. In this section, we will see how such systems are built. Aninformation retrieval system can be characterized by

1. **A corpus of documents.** Every system should decide what it needs to treat as a document: A paragraph, a page, or a multipage text.
2. **Queries posed in a query language.** A query specifies what the user needs to understand.

   The query language will be just a list of words, such as [AI book]; or it can specifya phrase of words that has to be adjacent, as in ["AI book"]; it will contain Booleanoperators as in [AI AND book]; it can include non-Boolean operators such as [AI NEAR book] or [AI book site: www.aaai.org].
3. **A result set.** This is the subset of documents that the IR system judges to be **relevant** tothe query. By *relevant*, we tend to mean likely to be of use to the person who posed the query,for the particular information need expressed in the query.
4. **A presentation of the result set.** This will be as simple as a ranked list of documenttitles or as advanced as a rotating color map of the result set projected onto a threedimensionalspace, rendered as a two-dimensional display.

3.3.1 The page rank algorithm:

It is a scoring measure based only on the link structure of web pages or website. A web page is important if it is pointed to by other important web pages. Our first technique for link analysis assigns to every node in the web graph a numerical score between 0 and 1 known as it page rank. Given a query, a web search engine computes a composite score for everyweb page content that combines or mixes hundreds of options like cos similarity and term proximity together with the Page rank score.

PageRank was one of the two original ideas that set Google's search except from other Web Search engines when it was introduced in 1997.

```
function HITS(query) returns pages with hub and authority numbers
  pages ← EXPAND-PAGES(RELEVANT-PAGES(query))
  for each p in pages do
    p.AUTHORITY ← 1
    p.HUB ← 1
  repeat until convergence do
    for each p in pages do
      p.AUTHORITY ← ∑i INLINKi(p).HUB
      p.HUB ← ∑i OUTLINKi(p).AUTHORITY
    NORMALIZE(pages)
  return pages
```

The HITS algorithm for computing hubs and authorities with respect to aquery. RELEVANT-PAGES fetch the pages that match the query, and EXPAND-PAGES addsin each page that links to or is linked from one of the relevant pages. NORMALIZE divideseach page's score by the sum of the squares of all pages' scores (separately for both theauthority and hubs scores).

(The other innovation was the use of anchor text—the underlined text in a hyperlink—to index a page, even though the anchor text was ona*different*page than the one being indexed.) PageRank was invented to solve the problem ofthe tyranny of TF scores: if the query is [IBM], how do we make sure that IBM's home page,ibm.com, is the first result, even if another page mentions the term "IBM" more frequently?

The idea is that ibm.com has several in-links (links to the page), so it should be ranked higher:every in-link is a vote for the standard of the linked-to page. But if we only counted in links,then it would be possible for a Web spammer to create a network of pages and have all of them allpoint to a page of his selecting, increasing the score of that page.

Therefore, the PageRankalgorithm is designed to weight links from high-quality sites more heavily. What is a highqualitysite? Onethat is linked to by other high-quality sites. The definition is Algorithmic or recursive, butwe will see that the recursion bottoms out properly. The PageRank for a page p is defined as:

$$PR(p) = \frac{1-d}{N} + d\sum_i \frac{PR(in_i)}{C(in_i)},$$

Where PR (p) is the PageRank of page p, N is the total number of pages in the corpus, in_i is the pages that link in to p, and C (in_i) is the count of the total number of out-links on page in_i. The constant d is a damping factor. It can be understood through the **randomsurfer model**: imagine a Web surfer who starts at some random page and begins exploring.

With probability d (we'll assume d=0.85) the surfer clicks on one in all the links on the page (choosing uniformly among them), and with chance 1 − d she gets uninterested in the pageand restarts on a random page anywhere on the Web. The PageRank of page p is then theprobability that the random surfer will be at page p at any purpose in time. Page Rank can becomputed by an iterative procedure: starts with all pages having PR(p)=1, and iterate thealgorithm, updating ranks until they converge.

Thus PageRank could be global ranking of all web pages based on their locations in the web graphstructure. PageRank uses information that is external to the web pages – **backlinks**. Backlinks from important pages are moresignificant than backlinks from average pages.

Advantages of Page Rank:

- **FightingSpam.**A page is important if the pages pointing to it are important. Since it is hard for Web page owner to add in-links into his/her page from different important pages, it is thus not easy to influence Page Rank.
- **Page Rank is a global measure and is query independent.** Page Rank values of all the pages are computed and saved off-line rather than at the query time.

3.3.2 The HITS algorithm:

Hyperlink Induced Topic Search (HITS) Algorithm is a Link Analysis Algorithm that rates Webpages,developed by Jon Kleinberg in 1999. This algorithm is used to the web link-structures to discover and rank theWebPages relevant for a particular search.

HITS use hubs and authorities to define a recursive relationship between webpages. To get knowledge of HITS Algorithm, we first have to get knowledge about Hubs and Authorities.

- Given a query to a Search Engine, the set of highly relevant web pages are called **Roots**. They arepotential **Authorities**.
- Pages that are not very relevant but point to pages in the Root are called **Hubs**. Thus, an Authority is apage that many hubs link and a Hub is a page that links tomany authorities.

Algorithm –

-> Let number of iterations be k.
->each node is assigned a Hub score = 1 and an Authority score = 1.
-> Repeat k times:

- ***Hub update:** Each node's Hub score = (Authority score of each node it points to).*
- ***Authority update:** Each node's Authority score = (Hub score of each node pointing to it).*
- *Normalize the scores by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.*
- **Two sets of inter-related pages:**
  - ⇨ Hub Pages-good lists of links on a subject
  - ⇨ Authority pages-occur recurrently on good hubs for the subjects.

The HITS algorithm

$$H(x) \leftarrow -\sum a(y)$$

$$A(x) \leftarrow -\sum h(y)$$

The Hyperlink-Induced Topic Search algorithm, also known as "Hubs and Authorities" orHITS, is another influential link-analysis algorithm, HITS differs from Page Rank in

several ways. First, it is a query-dependent measure: it rates pages with respectto a query. That means that it must be computed anew for each querya computationalburden that most search engines have elected not to take on. Given a query, HITS first findsa set of pages that are relevant to the query. It does that by intersecting hit lists of querywords, and then adding pages in the link neighborhood of these pages—pages that link to oris linked from one of the pages in the original relevant set.

Each page in this set is considered an **authority** on the query to the degree that otherpages in the relevant set point to it. A page is considered a **hub** to the degree that it pointstoother authoritative pages in the relevant set. Just as with PageRank, we don't want tomerely count the number of links; we want to give more value to the high-quality hubs andauthorities. Thus, as with PageRank, we iterate a process that updates the authority score ofa page to be the sum of the hub scores of the pages that point to it, and the hub score to bethe sum of the authority scores of the pages it points to. If we then normalize the scores andrepeat k times, the process will converge.

Both PageRank and HITS played important roles in developing our understanding ofWeb information retrieval. These algorithms and their extensions are used in ranking billionsof queries daily as search engines steadily develop better ways of extracting yet finer signalsof search relevance.

➢ **PageRank vs. HITS**

PageRank	HITS
Computed for all web pages stored prior to the query	Performed on the subset generated by each query
Computes authorities only	Computes authorities and hubs
Fast to compute	Easy to compute, realTime execution is hard.

➢ **Information Retrieval vs.Information Extraction:**

- **Information Retrieval**: Given a set of terms and a set of document terms select only the most relevant document (precision and preferably all the relevant ones (recall).
- **Information Extraction**: Extract from the text what the document means.

3.4 Information Extraction:

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents and other electronically represented sources. In most of the cases this activity concerns processing human language texts by means of natural language processing (NLP).

- Information Extraction refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources.
- Identify phrases in language that refer to specific types of entities and relations in text
- Named entity recognition is the task of identifying names of people, places, organizations, etc. in text.
- Relation extraction identifies specific relations between entities. A man works for IBM -> PERSON works for ORGANIZATION

Information extraction is the process of acquiring knowledge by skimming a text and looking for occurrences of a particular class of object and for relationships among objects. A difficult task is to extract instances of addresses from Web pages, with database fields for street, city, state, and zip code.

3.4.1 Finite state automata for information extraction:

The simplest type of information extraction system is an **attribute-based extraction** system that assumes that the entire text refers to a single object and the task is to extract attributes of that object.

For example,the problem of extracting from the text "IBM ThinkBook970.Our price: Rs.399.00". the set of attributes

```
{
        Manufacturer=IBM,
        Model=ThinkBook970,
        Price=Rs.399.00
}
```

We can address this problem by defining a **template** (also known as **a pattern**) for each attribute we would like to extract. The template is defined by a finite state automaton, the simplest example of which is the **regular expression**, or **regex**.

Regular expressions are used in UNIX commands such as grep and in word processors such as Microsoft Word.

Templates are always defined in three parts: a prefix regex, a target regex, and a postfix regex. For prices, the target regex is as above, the prefix would look for strings such as "price:" and the postfix could be empty. The idea is that some clues about an attribute come from the attribute value itself and some come from the surrounding text.

If a regular expression for an attribute matches the text exactly once, then we can pull out the portion of the text that is the value of the attribute. If there aremiss-match, it means that was a default value or given attribute missing; but if there are several matches, we

should a process to choose among them. One strategy is to have several templates for each attribute, ordered by priority.

For example, the top-priority example for price might look for the prefix "our price:" if that is not found, we look for the prefix "price:" and if that is not found, the empty prefix. Another strategy is to require all the matches and find some way to choose among them.

One step up from attribute-based extraction systems are **relational extraction** systems, which deal with multiple objects and also the relations among them. Thus, when these systems see the text "Rs.249.99," they need to determine not just that it is a price, but also which object has that price.

There are relational-based extraction system is FASTUS, which handles news stories about corporate mergers and acquisitions. It can read the story Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan. And extract the relations:

e∈ Joint Ventures ∧ Product(e, "*golf* clubs") ∧ Date(e, "Friday")
∧ Member (e, "Bridgestone Sports Co") ∧ Member (e, "a local concern")
∧ Member (e, "a Japanese trading house").

A relational extraction system can be built as a series of **cascaded finite-state transducers.**

So the system consists of a series of small, efficient finite-state automata (FSAs), where each automaton receives text as input, transfers the text into a different format, and passes it along with the next automaton. FASTUS consists of five stages:

1. Tokenization
2. Complex-word handling
3. Basic-group handling
4. Complex-phrase handling
5. Structure merging

FASTUS's first stage is **tokenization**, which segments the stream of characters into tokens (words, numbers, and punctuation). For English, tokenization can be simple; just separating characters at white space or punctuation does a fairly good job. Some tokenizes also deal with markup languages such as HTML, SGML, and XML.

The second stage handles **complex words**, including collocations such as "set up" and "joint venture," as well as proper names such as "Bridgestone Sports Co." These are recognized by a combination of lexical entries and finite-state grammar rules. For example, a company name might be recognized by the rule

Capitalized Word+ ("Company" | "Co" | "Inc" | "Ltd").

The third stage handles **basic groups**, meaning noun groups and verb groups. The idea is to chunk these into units which will be managed by the later stages. But here we have simple rules that only approximate the complexity of English, but have the advantage of being representable by finite state automata. The example sentence would emerge from this stage as the following sequence of tagged groups:

1 NG: Bridgestone Sports Co. 2 VG: said 11 CJ: and

3 NG: Friday 12 NG: a Japanese trading house

4 NG: it 13 VG: to produce

5 VG: had set up 14 NG: golf clubs

6 NG: a joint venture 15 VG: to be shipped

7 PR: in 16 PR: to

8 NG: Taiwan 17 NG: Japan

9 PR: with

10 NG: a local concern

Here NG stands for noun group, VG stands for verb group, PR stands for preposition, and CJ stands for conjunction.

The fourth stage combines the basic groups into **complex phrases**. Again, the aim is to have rules that are finite-state and therefore it can be processed quickly, and will result in unambiguous (or nearly unambiguous) output phrases. One type of combination rule deals with domain-specific events. For example, the rule

Company+ SetUpJointVenture ("with" Company+)?

Captures one way to describe the formation of a joint venture. This stage is the first one inthe cascade where the output is placed into a database template as well as being placed in theoutput stream.

The final stage **merges structures** which were built up in the previous step. If the next sentence says "The joint venture will start production in January," then this step will notice that there are two references to a joint venture, and that they should be merged into one. This is an instance of the **identity uncertainty problem.**

In general, finite-state template-based information extraction works well for a restricted domain in which it is possible to predetermine, and how they will be mentioned. The cascaded transducer model helps modularize the necessary knowledge, easing construction of the system. These systems work especially well when they are reverse engineering text that has been generated by a program. For example, a shopping site on the Web is generated by a program that takes database entries and formats them into Webpages; a template-based extractor then recovers the original database.

Finite-state information extraction is less successful at recovering information in highly variable format, such as text written by humans on a variety of subjects.

3.4.2 Probabilistic model for information extraction.

Probabilistic language models based on n-grams recover a surprising quantity of data about a language. They can perform well on such diverse tasks as language identification, spelling correction, genre classification, and named-entity recognition.

These language models have millions of features, therefore feature selection and preprocessing of the data to reduce noise is important.

When information extraction must be attempted from noisy or varied input, simple finite state approaches fare poorly. It is too hard to get all the rules and their priorities right; it is better to use a probabilistic model rather than a rule-based model. The simplest probabilistic model for sequences with hidden state is the **hidden Markov model, or HMM**.

To apply HMMs to information extraction, we can either build one big HMM for all the attributes or build a separate HMM for each attribute. We'll do the second. The observations are the words of the text, and the hidden states are whether we are within the target, prefix, or postfix part of the attribute template, or in the background (not part of a template).

For example, here is a brief text and the most probable (Viterbi) path for that text for two HMMs, one trained to acknowledge the speaker in a talk announcement, and one trained to acknowledge dates. The "-" indicates a background state:

Text: There will be a seminar by Dr. Andrew McCallum on Friday

Speaker: - - - - PRE PRE TARGET TARGETTARGET POST -

Date: - - - - - - - - - PRE TARGET

HMMs have two big advantages over FSAs for extraction.

1. HMMs are probabilistic, and thus tolerant to noise. In a regular expression, if a single expected character is missing, the regex fails to match; with HMMs there is graceful degradation with missing characters/words, and we get a probability indicating the degree of match, not just a Boolean match/fail.

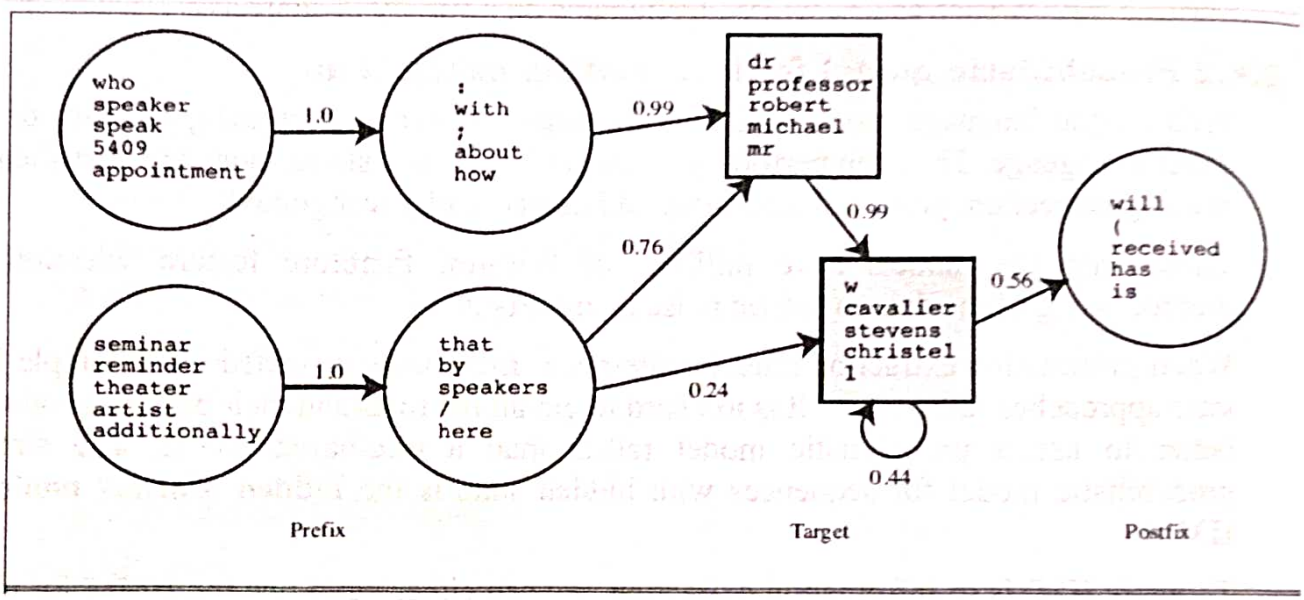


Figure: Hidden Markow model for the speaker of a talk announcement. The two aquare state are the target, the four circle to the left are the prefix, and the one on the right is the postfix. For each state, only a few of the high-probabilty words are shown. From Freitag and McCallum (2000).

2. HMMs can be trained from data; they don't require laborious engineering of templates, andthus they can more easily be kept up to date as text changes over time.

Note that we have assumed a certain level of structure in our HMM templates: they all consist of one or more target states, and any prefix states must precede the targets, postfix states most follow the targets, and other states must be background. This structure makesit easier to learn HMMs from examples.

For example, the word "Friday" would have high probability in one or more of the target states of the date HMM, and lower probability elsewhere.With sufficient training data, the HMM automatically learns a structure of dates that we find intuitive: the date HMM might have one target state in which the high-probability words are "Monday," "Tuesday," etc., and which has a high-probability transition to a target statewith words "Jan", "January," "Feb," etc.

Figure shows the HMM for the speaker of a vtalk announcement, as learned from data. The prefix covers expressions such as "Speaker:"and "seminar by," and the target has one state that covers titles and first names and anotherstate that covers initials and last names.

Once the HMMs have been learned, we can apply them to a text, using the Viterbialgorithm to find the most likely path through the HMM states.One approach is to applyeach attribute HMM separately; in this case you would expect most of the HMMs to

spendmost of their time in background states. This is appropriate when the extraction is sparse when the number of extracted words is small compared to the length of the text.

The other approach is to combine all the individual attributes into one big HMM, whichwould then find a path that wanders through different target attributes, first finding a speakertarget, then a date target, etc. Separate HMMs are better when we expect just one of eachattribute in a text and one big HMM is better when the texts are more free-form and densewith attributes. With either approach, in the end we have a collection of target attributeobservations, and have to decide what to do with them. If every expected attribute has onetarget filler then the decision is easy: we have an instance of the desired relation. If thereare multiple fillers, we need to decide which to choose, as we discussed with template-basedsystems. HMMs have the advantage of supplying probability numbers that can help makethe choice. If some targets are missing, we need to decide if this is an instance of the desiredrelation at all, or if the targets found are false positives. A machine learning algorithm can betrained to make this choice.

3.5 Examples: Applications of Natural Language Processing:

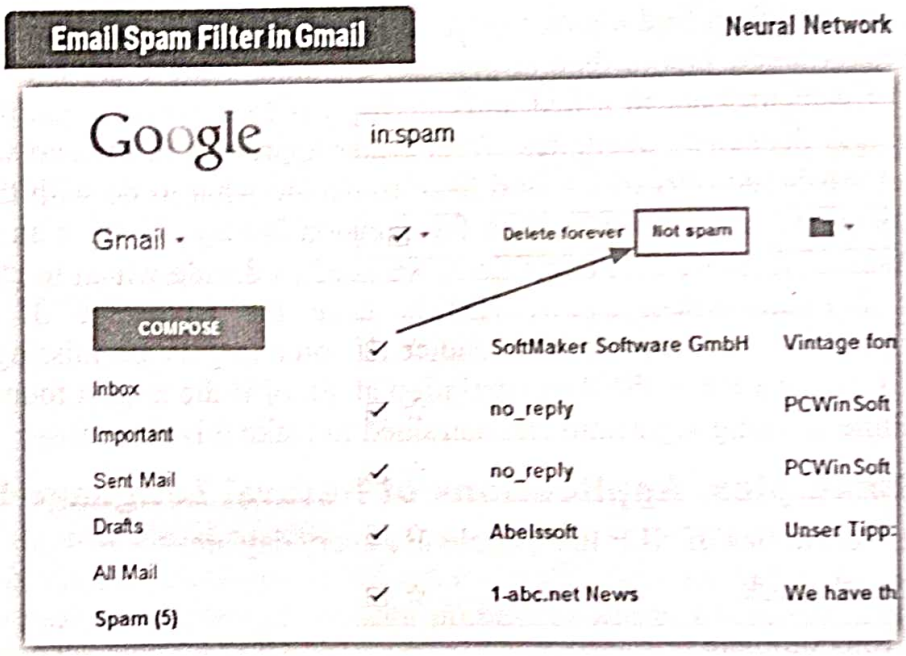
A few examples of NLP that people use every day are:

- Spell check
- Autocomplete
- Voice text messaging
- Search results
- Spam filters
- Related keywords on search engines
- Siri, Alexa, or Google Assistant

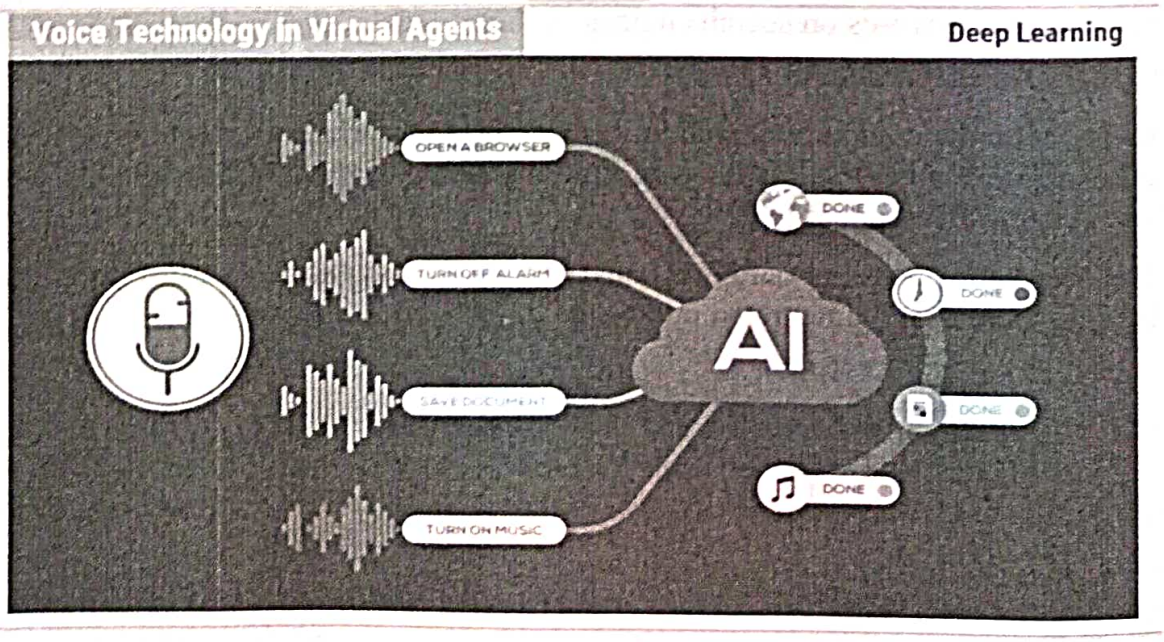
In any case, the computer is able to identify the appropriate word, phrase, or response by using context clues, the same way that any human would. Conceptually, it's a fairly straightforward technology.

The best-known **example of NLP**, smart assistants such as Siri, Alexa and Cortana have become increasingly integrated into our lives

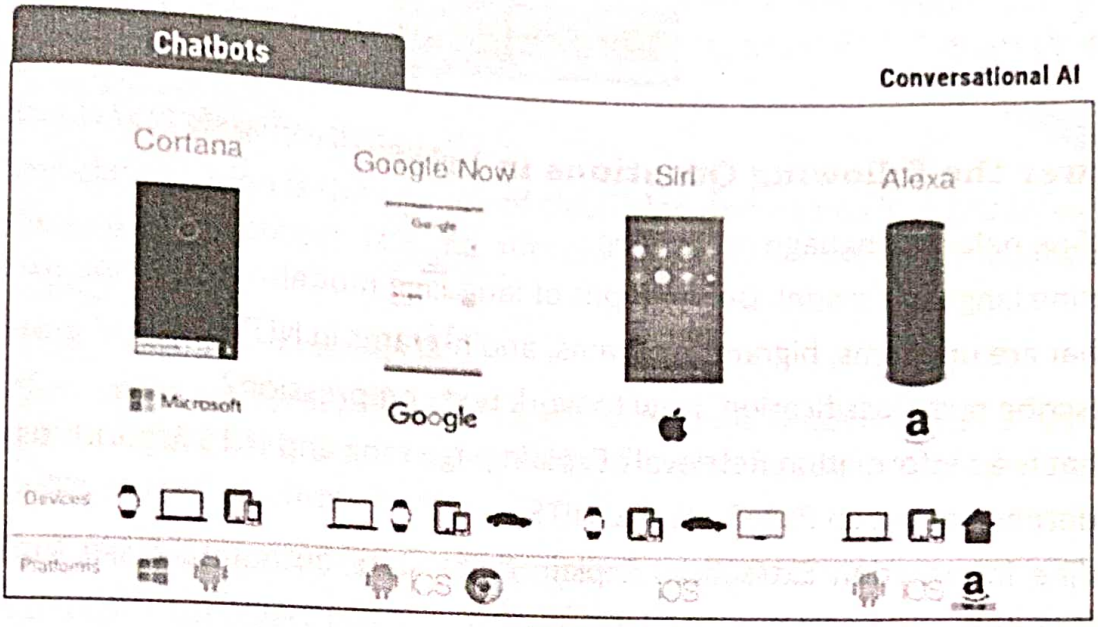
Natural Language Processing



3.6 Case Study: Automated Voice Assistants, Chat bots.



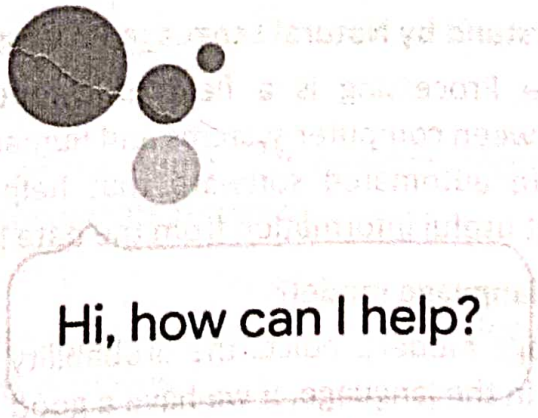
➢ **Chat bots.**



➢ **Automate support:**

Chatbots are nothing new, but advancements in NLP have increased their usefulness to the point that live agents no longer need to be the first point of communication for some customers. Some features of chatbots include being able to help users navigate support articles and knowledge bases, order products or services, and manage accounts.

- **Chatbots**: To provide a better customer support service, companies have started using chatbots for 24/7 service. Chatbots helps resolve the basic queries of customers. If a chatbot is not able to resolve any query, then it forwards it to the support team, while still engaging the customer. It helps make customers feel that the customer support team is quickly attending them. With the help of chatbots, companies have become capable of building cordial relations with customers. It is only possible with the help of Natural Language Processing.



Exercises

❖ Answer the following Questions in brief.

1. Define natural language processing.
2. Define language model. Define types of language model.
3. What are unigrams, bigrams, trigrams, and n-grams in NLP?
4. Describe text Classification. How to work text compression?
5. What is an information Retrieval? Explain page rank and HITS Algorithms.
6. Difference between PageRank and HITS.
7. Define information Extraction. Explain finite state automata (FSA) for information extraction.
8. Define Probabilistic model.

❖ Short Questions with Answer:

1. What is N-gram character model?

Ans: An n-gram model is a technique of counting sequences of characters or words that allows us to support rich pattern discovery in text. ... In other words, it tries to capture patterns of sequences (characters or words next to each other) while being sensitive to contextual relations (characters or words near each other).

2. What is the corpus in NLP?

Ans: Corpus or corpora (plural), is a collection of the text of a similar type, for example, movie reviews, social media posts, etc.

3. What do you understand by Natural Language Processing?

Ans: Natural Language Processing is a field of computer science that deals with communication between computer systems and humans.
It is used to create automated software that helps understand human spoken languages to extract useful information from the data it gets in the form of audio.

4. What is an n-gram language model?

Ans: An N-gram language model predicts the probability of a given N-gram within any sequence of words in the language. If we have a good N-gram model, we can predict p(w | h) – what is the probability of seeing the word w given a history of previous words h – where the history contains n-1 words.

N-gram refers to a number *n* of sequential items. A word 1-gram (uni-gram) for the sentence *"this is true"* would be [["this"],["is"],["true"]]; A 2-gram (bi-gram) would be: [["this", "is"], ["is", "true"]].

5. What is text classification in AI?

Ans: Text classification is the process of classifying documents into predefined categories based on their content. It is the automated assignment of natural language texts to predefined categories.

6. Define information retrieval.

Ans: Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

7. What are the areas of AI for information retrieval?

Ans: Concepts surveyed include pattern recognition, representation, problem solving and planning, heuristics, and learning. The paper concludes with an outline of areas for further research on artificial intelligence in information retrieval systems.

8. Explain about HITS algorithms

Ans:

- Hypertext induced Topic selection is a link analysis method developed by John Kleinberg in 1999 using Hub and Authority scores.

  Two sets of inter-related pages:

- Hub Pages-good lists of links on a subject
- Authority pages-occur recurrently on good hubs for the subjects.

  The HITS algorithm

  $H(x) <- \sum a(y)$

  $A(x) <- \sum h(y)$

9. What is Recall?

Ans: Recall is the ratio of the number of relevant documents retrieved to the total number of relevant documents retrieved.

10. What is precision?

Ans: Precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved.

11. Define authorities?

Ans: Authorities are pages that are recognized as providing significant, trustworthy and useful information on a topic. In-degree is one simple measure of authority. However indegree treats all links as equal.

12. Define hubs.

Ans: Hubs are index pages that provide lots of useful links to relevant content pages. Hub pages for IR are included in the home page.

❖ Multipal choice Queations - MCQs:

1. What is full form of NLP?

A. Nature Language Understanding
B. Natural Long Processed
C. Natural Language Processing
D. None of the Above

2. Natural language processing can be divided into the two subfields of:

A. context and expectations
B. generation and understanding
C. semantics of pragmatics
D. recognition and synthesis

3. The area of AI that investigates methods of facilitating communication between people and computers is:

A. natural language processing
B. symbolic processing
C. decision support
D. robotics

4. What are the input and output of an NLP system?

A. Speech and noise
B. Speech and Written Text
C. Noise and Written Text
D. Noise and value

5. Given a sound clip of a person or people speaking, determine the textual representation of the speech.

A. Text-to-speech
B. Speech-to-text
C. Both A and B
D. None of the Above

6. What is Machine Translation?

A. Converts one human language to another
B. Converts human language to machine language
C. Converts any human language to English
D. Converts Machine language to human language

7. The n-gram models includes _____.
   A. Unigram
   B. trigram
   C. Bigram
   D. All of the Above

8. Data Compression means to the file size.
   A. Increase
   B. Decrease
   C. Can't say
   D. none of the above

9. Data compression and encryption both work on binary
   A. false
   B. true

10. What is compression?
   A. To compress something by pressing it very hardly
   B. To minimize the time taken for a file to be downloaded
   C. To reduce the size of data to save space
   D. To convert one file to another Answer

11. Why data compressed?
   A. To optimise the data
   B. To reduce secondary storage space
   C. To reduce packet congestion on network
   D. Both (b) and (c)

12. What is a type of data compression?
   A. Resolution
   B. Zipping
   C. Inputting
   D. Caching

13. Data compression involves
   A. compression only
   B. Reconstruction only
   C. Both compression and Reconstruction
   D. None of the above

14. Compression is the method which eliminates the data which is not noticeable and compression does not eliminate the data which is not
   A. Lossless, Lossy
   B. Lossy, lossless
   C. None of these
   D. None of the above

15. IR objective is to retrieve _____ the relevant documents.
   A. All
   B. Specified number
   C. half
   D. None of the above

16. Full form of HITS is _____
   A. Hyperlink Indicated Topic Search
   B. Hyperlink Induced Topic Search

C. Hyphen Indeed Topic Search D. Hyperlink Induced Transfer Search

17. _____ played important roles in developing our understanding of Web information retrieval.

A. Authority B. Hub

C. HITS D. Both PageRank and HITS

18. Bag of Words in text preprocessing is a _____

A. Feature scaling technique B. Feature extraction technique

C. Feature selection technique D. None

Answer:

1. Ans.: C Natural Language Processing
2. Ans.: B. generation and understanding
3. Ans.: A. natural language processing
4. Ans.: B. Speech and Written Text
5. Ans.: B
6. Ans.: A. converts one human language to another
7. Ans.: D. All of the Above
8. Ans.: B. Decrease
9. Ans.: B. true
10. Ans.: C. To reduce the size of data to save space
11. Ans.: D. Both (b) and (c)
12. Ans.: B. Zipping
13. Ans.: C. Both compression and Reconstruction
14. Ans.: B. Lossy, lossless
15. Ans.: A. All
16. Ans.: B. Hyperlink Induced Topic Search
17. Ans.: D. Both PageRank and HIT
18. Ans.: B. Feature extraction technique