International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 6, Issue 11, November 2017, ISSN: 2278 – 1323

A Study on Natural Language Processing for Human Computer Interaction

Ms. Anmol Sajnani
B.Tech Integrated
Computer Engineering,
MPSTME, NMIMS

Mr. Harsh Kamani
B.Tech Integrated
Computer Engineering,
MPSTME, NMIMS

Mr. Hitesh Jeswani
B.Tech Integrated
Computer Engineering,
MPSTME, NMIMS

Mr. Krishna Samdani
Assistant Professor
Computer Engineering Department,
MPSTME, NMIMS

Abstract**—**We have long envisioned that one day computers will understand natural language and anticipate what we need, when and where we need it, and proactively complete tasks on our behalf. The communication between human and a computer is called Human Computer Interaction(HCI). This paper gives an overview of eminent technological perspective and appreciation of the fundamental progress on how humans can interact with computer using Natural language. Natural Language Processing System takes simple language as an input, and then processes it to give required outputs. Finally, the paper concludes with the decision on future direction for developing techniques in human computer interface and it also discusses various techniques used in each step of Natural Language Processing.

Index Terms**—**Natural Language Processing(NLP), Programming in Logic (Prolog), Natural Language Understanding (NLU), Natural Language Generation (NLG)

I. INTRODUCTION

In the last few years, IT industry has been focusing on new initiative towards developing a system which can carry out Human Computer Interaction. Interaction between human and a device is possible through use of some core set of technologies, such as machine learning, speech recognition, NLU, question answering(QA), dialogue management(DM), NLG, text-to-speech(TTS) synthesis, and data mining.

Natural Language Processing is simply a part of Artificial Intelligence which has to deal with language. Natural Language Processing broadly refers to the study and development of computer systems that can interpret human speech and text.

Human communication is indefinite at times; people on the internet use abbreviations and don't bother to correct any spelling errors. These inconsistencies make computer analysis of natural language difficult at best, but in the last few years Natural Language Processing as a field has progressed immeasurably.

Some of the applications of natural language processing are described below:

Personal Digital Assistant (PDA) Combining Natural Language Processing with Machine Learning implementation of PDA has become possible. PDAs are built to help the user get things done and provide easy access to personal/external structured data, web services, and applications. PDA's can work proactively(giving suggestions and inferences) and reactively(performing actions asked by the user) [1].

Machine translation is the technique of converting one natural language to another with the help of computer systems. Machine translation has become more accurate since the development in the field of machine learning and natural language processing. Now-a-days many IT companies use machine translation on a large scale [2].

IBM Watson System is question and answering system. IBM Watson correctly interprets the input and finds the answer. IBM Watson uses Prolog. Prolog is a general-purpose programming language which is associated with computer linguistics. Prolog uses recursion, facts, rules and grammar to find resolve the question. IBM Watson runs on 10 servers running GNU/Linux with 10 TB of RAM.

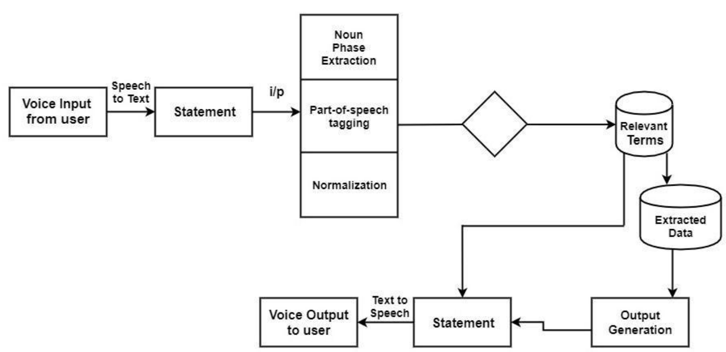

Fig. 1. General Architecture of Natural Language Processing

In this paper, the recent developments on various technologies that are used in Natural Language Processing are reviewed; starting from taking the input, processing it and giving the output in Section II. Some inferences drawn are mentioned in Section III, followed by conclusion in Section IV. Confirmed future in this topic is expressed in Section V as detailed below.

II. BRIEF DESCRIPTION

A system which is designed for human computer interaction should take the input using a microphone. By converting this speech input into text followed by machine level language, it should generate the output which is as natural as talking to a human.

The order in which this system works is given as follows:

A. Receiving the Input(speaker)
B. Processing of Input
C. Generation of Output

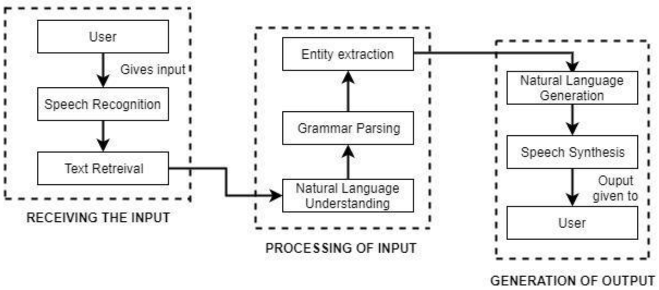

Fig. 2. Flowchart representing three phases of Human Computer Interaction

A. Receiving the Input

1) Speech Recognition: In Speech Recognition technology speech given as an input is converted to text, Automatic Speech Recognition is used for that purpose. Automatic Speech Recognition (ASR) can be defined as the independent, computer-driven transcription of spoken language intoreadable text in real time. ASR is technology that allows a computer to identify the words that a person speaks into a microphone or telephone and convert it to written text. Although ASR technology is not yet at the point where machines understand all speech, in any acoustic environment, or by any person. It is used on a day-to-day basis in a number of Natural Language Processing applications and services.

The ultimate goal of ASR research is to allow a computer to recognize in real-time, with 100% accuracy, all words that are intelligibly spoken by any person, independent of vocabulary size, noise, speaker characteristics or accent.Today, if the sys-tem is trained to learn an individual speaker's voice, then much larger vocabularies are possible and accuracy can be greater than 90%. Most commercial companies claim that recognition software can achieve between 98% to 99% accuracy if operated under optimal conditions. 'Optimal conditions' usually assume that users have speech characteristics which match the training data, can achieve proper speaker adaptation, and work in a clean noise environment [3].

Figure 3 shows the block diagram of a typical ASR system. It is composed of two major components: the front end and the decoder. The front end block extracts spectrum representation of the speech waveform. The most widely used features are Mel Frequency Cepstral Coefficients (MFCC) [5]. The decoder block searches the best match of word sequences for the input acoustic features based on acoustic model, lexicon, and language model.

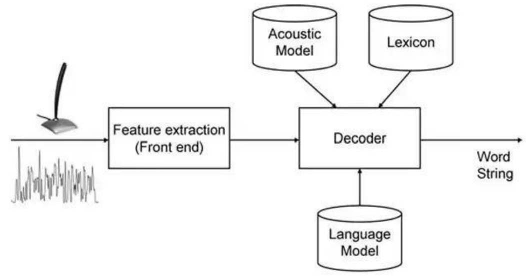

Fig. 3. Diagram of Automatic Speech Recognition System [4]

2) Text Retrieval: One of the text retrieval method used in Natural Language Processing system is Naive Bayes Classifier. Assuming the dataset on 1000 pieces of fruit. The fruit being a Banana, Orange, or some Other fruit and imagine we know 3 features of each fruit, whether its long or not, sweet or not and yellow or not, as displayed in the table below:

Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	**500**	**650**	**800**	**1000**

From the table it is seen that that 50% of the fruits are bananas, 30% are oranges and 20% are other fruits. Now based on the training set, it can be observed that from 500 bananas 400(0.8) are long, 350(0.7) are Sweet and 450(0.9) are Yellow. Out of 300 oranges 0 are long, 150(0.5) are Sweet and 300(1) are Yellow. From the remaining 200 fruits, 100(0.5) are long, 150(0.75) are Sweet and 50(0.25) are Yellow. This should provide enough evidence to predict the class of another fruit as it is introduced [6].

Given a new piece of fruit and asked to predict the class of the fruit, considering the additional fruit is long, sweet and yellow, now the naive bayes classifier can classify it using the following formula and subbing in the values for each outcome, whether it is a banana, an orange or any other fruit [6]- [9].

$$P(c|x) = \frac{P(x|c)\,P(c)}{p(x)} \quad (1)$$

P (c|x) is the posterior probability of c.
P (c) is the prior probability of class.
P (x|c) is the likelihood which is the probability of predictor given class.
P (x) is the prior probability of predictor.

Using formula 1 for above example, based on the higher score assuming this long, sweet and yellow fruit is, in fact a banana. Referring to above example, Naive Bayes Classifier can be applied in Natural Language Processing for text classification problems. It is used to classify different words into their respective categories. For example noun words into noun category, verbs into verbs category, etc.

B. Processing of Input

1) Natural Language Understanding: Natural Language Understanding is a part of Natural Language Processing in artificial intelligence that deals with the ability to read text, process it and understand the meaning. Natural Language Understanding is considered an AI-hard problem [10]. NLU involves disassembling of the input, then parsing the input into data that can be analyzed by the analysts or machine learning experts. This part is more complex than the reverse process of converting data to language because of the unexpected occurrences of symbols or words or phrases or unexpected features in any input. Human beings have a tendency to use abbreviations, or some slang word that might not even be in the language. So the Natural Language Understanding algorithm should be able to tackle such difficulties and carry out the conversion smoothly [10].

Example: I will be attending an event in the Hotel KLG

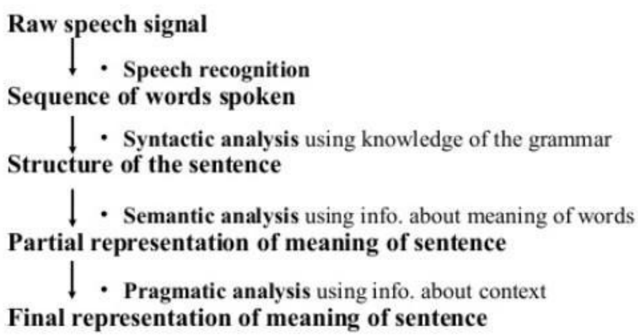

Fig. 4. Natural Language Understanding

international in the city of Chandigarh at 7.00pm and then I will be going to Chandigarh airport at 8.45 pm and I will catch a flight to Mumbai which departs at 10.45pm on 15th of this month.

Now in the above example there are many unnecessary words that need to be taken care of before the system tries to understand the statement. These words will be removed in the text pre-processing phase. Now the NLU algorithm will be run on the statement and the output will be some data on which statistics can be applied. The useful data generated is as follows:

1) There is an event in Hotel KLG international
2) The timing for the event is 7.00pm
3) There is flight to Mumbai from Chandigarh
4) Flight departs at 10.45pm

Now all of the above data is very important not to just one person but many people who are present in the area on that particular day. This is how NLU works. It tries to breakdown the input to generate useful information out of a natural language sentence. This data can then be used by the system to generate appropriate response for the user interacting with the system.

In the upcoming section, we will discuss how grammar parsing takes place; which is one of the important aspect of Natural Language Understanding.

2) Grammar Parsing: Current NLP systems uses program-ming languages such as C++, Java, Python or Rust. Pro-cedure oriented programming languages and object-oriented languages makes it mandatory to implement the data structures from the very fundamental level. Programming languages similar to Prolog can be used in NLP systems. This can make parsing effortless. This paper will mainly line up with the idea of logic programming.

In Prolog some data has to be provided for parsing natural language, this data will be declarative and will represent some form of relationship between objects [11]. An example of such statement will be a fact in Prolog. Facts in Prolog are used to reflect relation between objects.

```
article(a). noun(person). verb(observe).
```

The above example shows possible ways to represent verb, noun and articles in Prolog. Once facts have be declared in the Prolog knowledge base, we can ask questions to the system. Format of question in Prolog is as follows:

```
-? article(Article).
```

It is important to observe that variables in Prolog are denoted by uppercase letters, here Article is a variable. Depending on the facts contained in Prolog knowledge base the system will find the appropriate answer. Couple of more examples of facts:

```
likes(john, flowers). likes(john, mary).
likes(paul, mary).
?-likes(john, X).
X = flowers;
X = mary;
no
```

A very important applications area of Prolog is parsing. In fact, Prolog originated from attempts to use logic to express grammar rules and to formalize the process of parsing. DCG arsise from adding features of Prolog to context-free grammars [12]. DCG will make parsing effortless. DCG provides a way to represents grammar of natural or formal languages. Example of DCG:

```
sentence --> noun_phrase, verb_phrase. noun_phrase --> det, noun.
verb_phrase --> verb, noun_phrase
```

The above DCG will generate valid sentnce such as the boy saw the bat. Prologs searching and backtracking methods can be used to unify the fact and create valid sentences. It is not necessary that the semantics of a sentence is similar to formal language.

The process of ***lexical analysis*** revolves around tokenizer. The tokenizer handles whitespaces, hyphens, punctuations, enclitics and special expressions. Lexical analysis is a fundamental step which separates the string in such a way that everything divided into chunks, sentences and paragraphs.

The initial step in lexical analysis is converting the input into separate entities. In modern languages whitespaces are used as delimiters. Delimiters will indicate that the end of current word has been found. It is also mandatory to remove trailing spaces, quotation marks, parentheses and punctuators.

The second step in lexical analysis will be handling abbreviations. It is a common practice that a word in between will be followed by a period, such words are known as abbreviations. Words with periods attached should be tokenized with the period. To identify if the word is an abbreviation the word can be looked up in the knowledge base. Handling abbreviation is also made difficult if there are ambiguity for example: in can be used for inches and no can be used for numbers.

The third step in lexical analysis is handling hyphens. Some tokenizer includes hyphens as a part of token or some tokenizer do not consider hyphens as token or some tokenizer separates the hyphenated part.

The fourth step would be for handling special expressions such as email addresses, urls, dates, time, paper and book citation, and telephone numbers. This can cause a lot of ambiguity for the tokenizer since special expressions are combinations if alpha numerical and punctuation syntax. To avoid confusion a pre-processor must be designed for telephone numbers, dates, etc. Example of tokenization of a simple sentence for example consider the following sentence:

Input: The Cat sat on the mat.
Tokenized output: [The, Cat, sat, on, the, mat, .]

The above example uses Prolog list data structure. If any error occurs the input will discarded.

After lexical analysis next step is to analyze the syntax of input. In ***syntax analysis*** the parser determines which part of speech is applicable to the given text. Structure of sentence might have multiple interpretation. Syntax analysis determines verbs, nouns, subjects, predicates, pronouns, adjective, adverb, prepositions, Et Cetra.

To parse grammar in natural language, it is mandatory to context-free grammar. Context-free grammar is independent of context, whether the context of other elements or parts of the sentence or of the larger discourse context of the sentence [13]. As discussed DCG can be utilized to express grammar rules and to formalize the process of parsing. Context-free grammar can be implemented with Prolog.

We will be understanding the method to handle morphology and distribution of words in English. English has very simple morphology but very complex syntax, therefore many ambiguities can be created during syntax analysis of a string. Morphology is the identification, analysis and description of the structure of words, practiced by morphologists [14]. It is important to compute how the word will placed in a sentence;

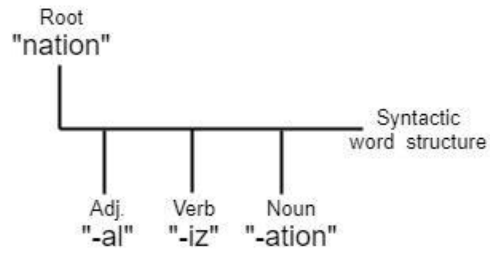

Fig. 5. Morphology of a word in standard-English.

this is done by grammar rules. Grammar rules consists of two types of symbols–terminals and non-terminals. Terminal symbols are those that are part of a generated sentence. Non-terminal symbols are those symbols that take part in the formation of a statement, but are not part of the generated statements [15]. Example of grammar rules are as follows:

S → N P V P
N P → noun | adj SN P
V P → verb VerbComp
VerbCompnull |N P | P P |N P |P P

Non-terminal symbols:
S = sentence
N P = noun phrase
SN P = simple noun phrase
V P = verb phrase
VerbComps = verb complements
P P = prepositional phrase
Terminal symbols:
det = determiner
adj = adjective
noun
prep = preposition
verb

Example of complex grammar rule and generation of valid sentence:

S | det adj SN P V P
S | det adj noun V P
S |det adj noun verb V erbComp S | det adj noun verb det | the | this | that | that | my | some | a

adj | good | new | bad | few noun | dog | man | chair desk prep | at |
before | down in verb | eat | bend | put | slept

Sentences generated by this grammar rule will be:

The good Dog slept.
*The good Chair slept.

The previous sentence has malformed semantics. The distribution of nouns can satisfy the grammar rules, therefore it is not necessary that the semantics of generated sentence will fit the standard-English. The marked sentence is semantically incorrect.

Semantics of a language provide meaning to its tokens and syntax structure. Semantics help interpret symbols, their types and their relations with each other. In natural language processing, ***semantic analysis*** is the process of relating syntactic structures, from phrase, clauses, sentences and paragraphs to writing them as a whole, to their language-independent meanings. Semantic analysis must do two important tasks:

1) It must map individual words into appropriate objects in the knowledge base or database
2) It must create the correct structures to correspond to the way the meaning of the individual words combine with each other

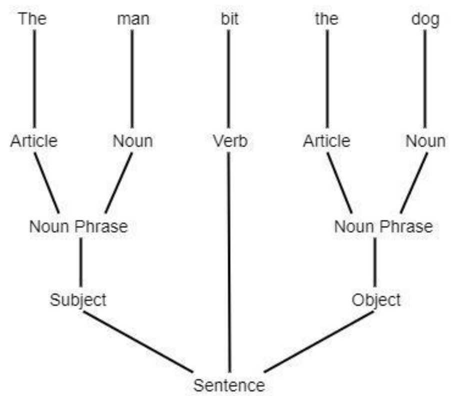

Fig. 6. Semantics Parse tree [16]

The parser only examines the syntax of the source language, i.e. whether the sequence of tokens is correct or not. Semantic analysis makes sure that the meaning is correct. For example, 'The man bit the dog', statement has the correct syntax as shown in the above parse tree, but the semantics of the statement is questionable- the man bit the dog instead of the dog biting the man.

So semantic analysis for programming languages makes sure that all the variables have the correct data types and that the argument-list and return value of every function-call agrees with the definition of the called function, etc.

3) Entity Extraction: Named-entity recognition (Entity Extraction) also known as entity chunking or entity extraction is the process of information extraction that locates and classifies named entities in text into statistical data such as names of persons, organizations, percentages, expression of times, etc.

In the above picture there are various entities that are being extracted from a single document. This document can contain names of persons, companies, addresses of the employees, their salary, certain links, phone numbers of various employ-ees, etc. The main task of entity extraction is to convert raw text data into useful statistical data with the help of labelling in machine learning. This new data can be used by analysts for statistics, predictions, changes in the companies, etc. This approach is aimed at finding out the maximum possible facts and entities that are present in the document. These entities can

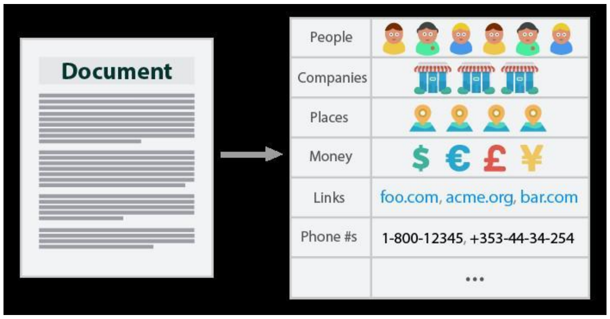

Fig. 7. Entity Extraction [16]

also be mapped to entities present in other documents on the internet with the help of relationship extraction. Relationship extraction finds the similarity between any given two entities and according gives the expected output. Although in this technique only entity extraction is done this assumes the relationship while mapping the entities of the document with the entities present on the internet [17].

Entity and Relationship Extraction: In the modern world, information extraction also known as entity extraction is gaining massive importance in the Information Technology field. Nowadays entity extraction is used to extract entities from a given document [18]. However real that document might be, there is always a chance for that document to be fake. So relationship extraction was introduced. Relationship extraction is very important in extracting information of the structured form from unstructured sources such as unorganized data.

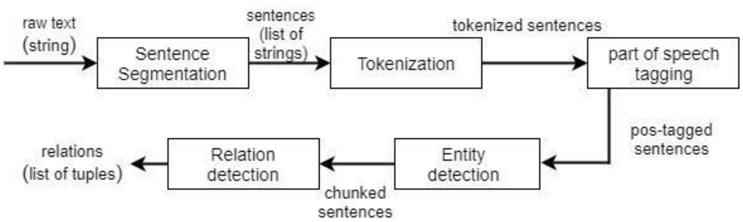

Fig. 8. Entity and Relationship Extraction [19]

As shown in the above figure entities are linked to each other with the help of relationship detection among them. This can be very useful when the user is trying to find accurate results but has the time to wait for such results. The technique of entity and relationship extraction was introduced to check if the document and the entities extracted from it are valid or not.

This kind of technique can be used in a content monitoring system to monitor the genuineness of the content available in the document. For example, nowadays many fake news articles are cooked up by the journalists, this technique can be used to detect such fake news articles available on the internet. This could be possible by finding out the relationship of the entities in the fake news article with the entities in the real news articles. If there is a relationship then the article is genuine and if not the article is termed fake by the system [20].

C. Output Generation

1) Natural Language Generation: Natural Language Generation is the task of generating language out of data analyzed by machine learning experts and data analysts. Natural Language Generation is sometimes also called as language production by the Psycholinguists. NLG software converts facts to human readable text using a series of steps [21]. This does not involve any parsing of the data or even breaking the data, it just involves translating statistically analyzed data into natural language. In NLU the system needs to disambiguate the input sentence to produce a language which can be understood by the computer system, but in NLG, the system converts data into natural language and needs to make decisions about how to put a concept into words. NLG can be very simple or very complex depending upon the implementation of the algorithm in the system [21].

Following are the stages in NLG:

1) Content determination: Deciding what information to mention in the output text. For instance, in the airport example whether to review personal details of the person or not.
2) Document structuring: Overall organization and structuring of the information that the system needs to convey. For example, suggesting setting a reminder for the event at 7.00pm to the user.
3) Aggregation: Merging of two or more similar sentences to improve readability and naturalness of the output. For example organize a schedule for the user for the 15th date of the month.
4) Lexical choice: Placement of words in place of the concepts. For example, deciding whether medium or moderate should be used to define traffic conditions on the route till the airport.
5) Referring expression generation: Creating referring expressions that identify objects and regions.
6) Realisation: This is the last and the most important phase in which the data analyzed is converted into actual text and this text should be correct according to the rules of syntax and semantic analysis.

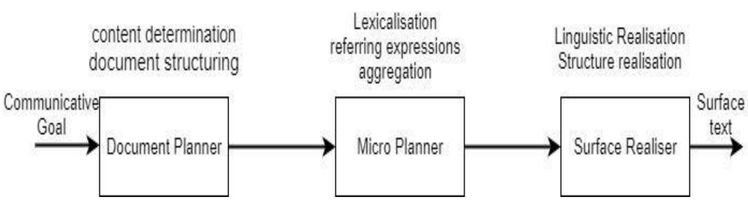

Fig. 9. Natural Language Generation System Architecture [22]

2) Speech Synthesis: Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech.A text to speech output is based on generating corresponding sound output when the text is inputted [23].Wide range of applications use text to speech technique in medicals, telecommunications fields, etc. Each spoken word is created from the phonetic combinationof a set of vowel and consonant speech sound units. Producing an artificial human speech is known as speech synthesis.

Working of text to speech is shown in Figure 10.

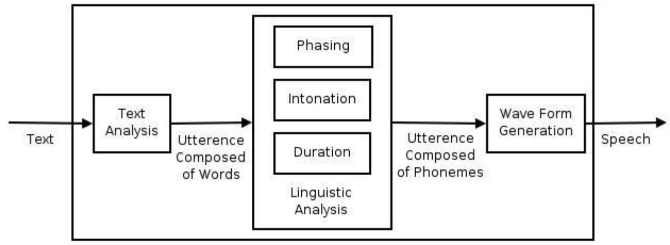

Fig. 10. Text to Speech Conversion Flow [24]

The Various speech synthesis methods that have been used for text to speech output for obtaining intelligible and natural output are Concatenative, Formant, Articulatory, Hidden Markov model (HMM).

1) Concatenative Synthesis: Concatenative synthesis is based on the concatenation (or stringing together) of seg-ments of recorded speech.In concatenative synthesis, the spoken sentence is broken down into words and words into syllables, demisyllables, phonemes, diaphones or triphones. Then concatenation and rearrangement of the above segments of recorded samples is done to create new words and sentences is known as concatenative synthesis [25].
2) Formant Synthesis: Also called as rules-based synthesis. The synthesized speech output is created using a model called acoustic model and also some additive parameters are used. Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech [23].
3) Articulatory Synthesis: Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there [23].
4) Hidden Markov Model(HMM): HMM-based synthesis is a synthesis method based on hidden Markov models, also called Statistical Parametric Synthesis. In this sys-tem, the frequency spectrum (vocal tract), fundamental frequency (voice source), and duration (prosody [26]) of speech are modelled simultaneously by HMMs [23], [25].

D. Related Study

Semantic Similarity between words Semantic relatedness or semantic similarity between words is observed as one of the basic hurdle for many Natural Language Processing (NLP) applications, such as sentence retrieval, word sense disambiguation, question answering, etc. Semantic relatedness or semantic similarity is a metric defined over a set of documents of terms, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their representation.

The research proposed in [27] gives a method to retrieve top-k similar words using a graph based approach. Graph Based model maps real dataset and query into the knowledge base WordNet. Then, start searching from the location of query in WordNet, and expand outwards by using query's hypernyms and hyponyms until all top-k similar words in the real dataset are not found. Refer Figure 11. This methodology works for different parts of speech too. Disadvantage of this

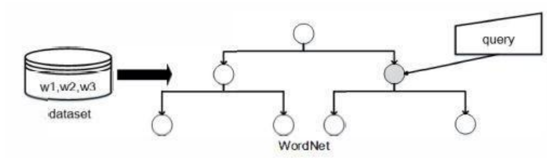

Fig. 11. Retrieve top-k similar words from real dataset by proposed model. [27]

approach is senses of word changes over time, these changes are not reflected in the training dataset which is Wordnet for Graph Based Approach.

This disadvantage is overcome in another approach for Semantic Similar word retrieval in which page counts and snippets are given as input to Support Vector Machine [28] and similarity value is calculated which comes between 0 and 1. If similarity value comes near to 0 then words are not similar but if the value is nearer to 1 then words are similar.

Page count is an estimate of number of pages that contain query words.Snippet is some text extracted by web search engine based on the query term given [29]. The system in [29] uses page counts and snippets retrieved by a search engine to measure semantic similarity between words. Various similarity scores are calculated from the page counts retrieved by the search engine for the queried conjunctive words. A lexical pattern extraction algorithm identifies the patterns from the snippets. Different patterns showing the same semantic relation are clustered using a lexical pattern clustering algorithm. At first stage, Support Vector Machine(SVM) was used.

But the major problem is that SVM cannot handle missing data which occur frequently in statistical data analysis.So the proposed system in [30] makes use of Latent Structural SVM (LS-SVM) which allows latent variables. The latent variables have no role in the output, but they are just intermediate representations. The below figure shows structural block diagram for LS-SVM.

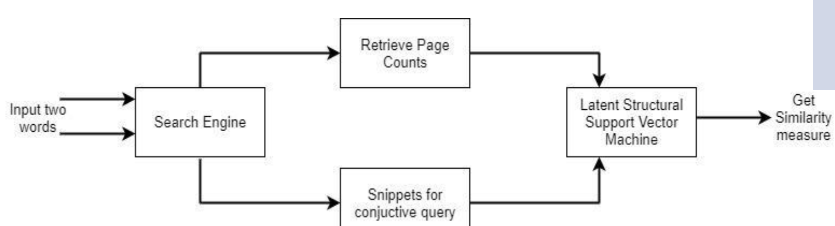

Fig. 12. LS-SVM Approach. [30]

Human Machine Dialog: The field of human-machine dialog has links with fields, such as Natural Language Pro-cessing (NLP), of which it is an essential application; Artificial Intelligence (AI), from which it arises and which completes the linguistic aspects with the reasoning and decision-making aspects; human-machine interfaces (HMIs), which it helps enrich by offering vocal interaction possibilities in addition to graphical and touch screen interactions; and, more recently, question-answering systems(QAS). Example:
User - I need a flight from New York to London, arriving at 10 pm ?
System - What day are you leaving?
User - Tomorrow.
System detects the missing information in your sentences.
Classification of Human Machine Dialog can be done as follows:

1) Finite state based systems
2) Frame based systems
3) Agent based systems [31]
4) Answering systems
5) Semi-dialogue systems
6) Full dialogue systems [32]

Text Mining: Text mining is the technique of converting unstructured data or free flowing data to structured data which can be used for analysis by machine learning experts. It is the process of finding or extracting non-trivial and interesting patterns or information from normal text documents. Text Mining is similar to data mining, except that data mining tools are designed to handle structured data from databases, whereas text mining can also work with unstructured or semi-structured data sets such as emails, text documents, etc [33].

The role of NLP in text mining is to deliver the system in the information extraction phase as an input. Information Extraction is the task of automatically extracting structured information from unstructured and semi-structured machine-readable documents. It involves defining the general form of the information that we are interested in as one or more templates, which are used to guide the extraction process.

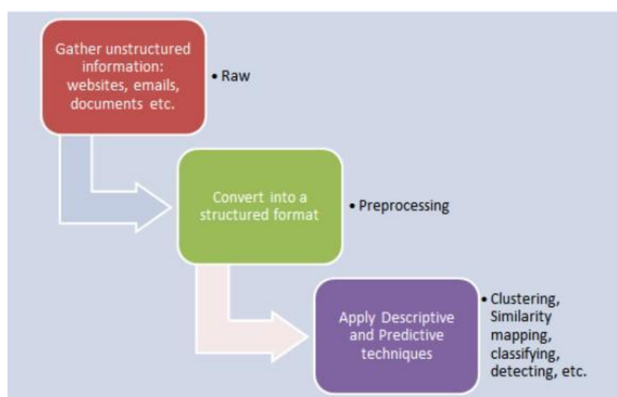

Fig. 13. Text Mining [34]

I. Performance of resultant systems based on different techniques for NLP

Receiving the Input	Speech Recognition	Y	Y	Y	Y	Y
	Naïve Bayes Classifier	Y	Y	Y	Y	Y
Processing of Input	Natural Language Understanding	Y	Y	Y	Y	Y
	Grammar Parsing	Y	Y	Y	Y	Y
	Entity Extraction	Y	Y	Y	Y	N
	Entity and Relationship Extraction	N	N	N	N	Y
Generation of Output	Natural Language Generation	Y	Y	Y	Y	Y
	Concatenative Speech Synthesis	Y	N	N	N	Y
	Formant Speech Synthesis	N	Y	N	N	N
	Articulatory Speech Synthesis	N	N	Y	N	N
	Hidden Markov Model	N	N	N	Y	N
Other techniques	Graph Based approach for Semantic similar word retrieval	Y	Y	Y	Y	N
	LS-SVM approach for Semantic similar word retrieval	N	N	N	N	Y
	Text Mining	y	Y	Y	Y	Y
Resultant System		Slow Inefficient Speech-output produced is very natural	Slow Inefficient Speech-output produced is unnatural	Slow Inefficient Producing speech-output is complex	Slow Inefficient Speech-output is not generated automatically	Fast Most Efficient Results Speech-output produced is very natural

III. INFERENCES

Inferences drawn from techniques studied in this paper are shown in above table. From all the reviewed techniques we can say that performance of Human Computer Interface system is based on its response time, efficiency, throughput, accuracy and many such parameters. The system which has performed best with efficient results, fast response and conveys natural(human like) output incorporates speech recognition, text retrieval, NLU, Entity and Relationship extraction, NLG, and concatenative speech synthesis into it. LS-SVM approach is efficient for semantic similar word retrieval because of its effectiveness and accuracy.

IV. CONCLUSION

In this paper, we discussed the topics relevant to the development of NLP systems. Based on the literature survey done, we can conclude that Automatic Speech Recognition technique is efficient for taking the input from the user. The required string is extracted using text retrieval and the retrieved text is then classified into respective categories using Naive Bayes Classifier. The input string then will go through parsing phase in Prolog. Prolog is used for writing parsing algorithms because it fits the mathematical model of computer linguistics. Next step in the process is NLU through which semantics of sentence is understood. Special expressions and entities are extracted using Entity Extraction and relationships between those entities are derived using Entity and Relationship Ex-traction. Here, the system goes from input processing phase to output generation. NLG is used for converting statistical data to Natural Language, and for Speech Synthesis concatenative synthesis is the most efficient method for converting text to speech. This paper made a clear and simple overview of working of natural language processing system for HCI.

V. FUTURE WORK

NLP techniques incorporate variety of methods to enable a machine to understand what is being said or written in human language not just single word in a comprehensive way. Some of the concepts which we will try to adopt in NLP system to enhance Human Computer Interaction are as follows:

1) Offline database: The current NLP system does not support offline interaction with the sytem. This limita-tion can be solved by introducing offline database. This database will record all the inputs when the system is in offline mode and will provide appropriate output when the system is back online. This can make the system more user-friendly and hence increasing the number of users interacting with the system.
2) Scale up and speed up: Increasing the data storage facility for the NLP systems and introducing machine learning algorithms to resolve queries with accuracy.
3) Context saving: Introducing the feature of context saving can make the NLP system more natural in interaction with the user.
4) Interfaces with traditional programming languages: There are many active packages such as swi-java and pylog which provides interface with logic programming language like prolog.

REFERENCES

[1] Ruhi Sarikaya, "Technology Behind Personal Digital Assistants", IEEE Signal Processing Magazine, January 2017.
[2] Machine Translation [Online] Available:https://www.ijser.org/paper/Example-Based-Machine-Translation-Using-Natural-Language-Processing.html
[3] ASR [Online] Available:http://support.docsoft.com/help/whitepaper-asr.pdf (2017, October 10)
[4] MFCC [Online] Available:https://en.wikipedia.org/wiki/Mel-frequency-cepstrum (2017, October 10)
[5] Speech Recognition [Online] (2017, September 23) Available:http://what-when-how.com/video-search-engines/speech-recognition-audio-processing-video-search-engines/
[6]Naive Bayes Classifier [Online] Available:https://www.analyticsvidhya.com/blog/2015/09/\\naive-bayes-explained/ (2017, August 23)
[7] Natural Language Processing [Online] Available:https://www.tutorialspoint.com/artificial-intelligence/artificial-intelligence-natural-language-processing.html (2017,August 23)
[8]Neha Sharma and Manoj Singh "Modifying Naïve Bayes Classifier for Multinomial Text Classification", \textit{IEEE International Conference On Recent Advances and Innovations in Engineering (ICRAIE-2016)}, December 23-25, 2016, Jaipur, India
[9] Naive Bayes Text Classifier [Online] Available:blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/ (2017,October 25)
[10] John Dowding, Jean Mark Gawron, Doug Applet, John Bear,Lynn Cherny, Robert Moore, Douglas Moran "GEMINI: A NATURAL LANGUAGE SYSTEM FOR SPOKEN-LANGUAGE UNDERSTANDING", SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025
[11] "Programming in Prolog Using the ISO standards" (5th edition); Springer Publication.
[12] "The Art of Prolog" (2nd edition); MIT Press Cambridge, Massachusetts, London, England.
[13]mind.ilstu.edu/curriculum/protothinker/natural_language_processing.php
[14] en.wikibooks.org/wiki/Linguistics/Syntax
[15] "Theory Of Computation", Oxford Higher Education; Vivek Kulkarni
[16] Entity Extraction [Online] Available:https://aylien.com/text-api/entity-extraction/ (2017, October 25)
[17] nltk.org. [Online] Available: http://www.nltk.org/book/ch07.htm (2017, October 25)
[18] Doug Downey, Oren Etzioni, and Stephen Soderland 2010. "Analysis of probabilistic model of redundancy in unsupervised information extraction". Artif. Intell., 174(11):726-848
[19] Alan Ritter, Sam Clark, Mausam and Oren Etzioni "Named Entity Recognition in Tweets: An Experimental Study" Computer Science and Engineering University of Washington Seattle, WA 98125, USA
[20] Mazhar Ul Haq, Hasnat Ahmed, Ali Mustafa Qamar "Dynamic Entity and Relationship Exraction from News Articles", 8-9 Oct. 2012
[21] James Curry, Weihang Zhu, Brian Craig, Lonnie Turpin, Jr. Majed Bokhari, Pavan Mhasavekar, "USING A NATURAL LANGUAGE GENERATION APPROACH TO DOCUMENT SIMULATION RESULTS", *Proceedings of the 2013 Winter Simulation Conference*, Beaumont, TX 77710, USA
[22] Natural Language Generation [Online] Available:https://github.com/simplenlg/simplenlg/wiki/Section-XV%E2%80%93-Appendix-A-%E2%80%93-NLG-and-SimpleNLG
[23] Sneha Lukose, Savitha S. Upadhya,"Text to Speech Synthesizer-Formant Synthesis", *International Conference on Nascent Technologies in the Engineering Field*, IEEE 2017 (2017 October 23)
[24] Text to Speech Flow [Online] Available: http://www.wezs.com/~danguy/monguy/TTS.html (2017 October 23)
[25] Speech Synthesis [Online] Available: https://en.wikipedia.org/wiki/Speech.synthesis (2017 October 23)
[26] Prosody [Online] Available:https://en.wikipedia.org/wiki/Prosody (2017 October 23)
[27] Yonggen Wang, Yanhui GU, Junsheng Zhou, Weiguang Qu, "A Graph-based Approach for Semantic Similar Word Retrieval", *2015 International Conference on Behavioral, Economic, and Socio-Cultural Computing*, BESC 2015, Oct 30-Nov 01, 2015, Nanjing, 24 China
[28] Support Vector Machine [Online] Available:http://docs.opencv.org/2.4/doc/tutorials/ml/\\introduction-to-svm/introduction-to-svm.html\#explanation (2017 September 25)
[29] Manasa.Ch, V.Ramana, "Measuring Semantic Similarity between Words Using Page Counts and Snippets", Manasa Ch et al , International Journal of Computer Science \& Communication Networks,Vol 2(4), 553-558
[30] S.Lavanya, S.S.Arya, "An Approach for Measuring Semantic Similarity between Words Using SVM and LS-SVM", *2012 International Conference on Computer Communication and Informatics* (ICCCI -2012), Jan. 10 – 12, 2012, Coimbatore, INDIA
[31] Suket Arora, Kamaljeet Batra, Sarabjit Singh, "Dialogue System: A Brief Review".
[32] Stavros Mallios and Nikolaos Bourbakis, "A Survey on Human Machine Dialogue Systems", Wright State University Dayton, OH, USA.
[33] Chandrasekhar Rangu, Shuvojit Chatterjee, Srinivasa Rao Valluru "Text Mining Approach for Product Quality Enhancement", *2017 IEEE 7th International Advance Computing Conference.*
[34] Text Mining [Online] Available:http://www.simafore.com/blog/bid/111839/3-ways-to-use-text-mining-with-RapidMiner-to-juice-up-your-job-search