Natural Language Processing by using Text Processing Algorithm

Sumanth V.
Assistant Professor,
Department of CSE,
R. R. Institute of Technology, Bengaluru,
Karnataka, India

Amrith Sharma
UG Student,
R. R. Institute of Technology,
Visvesvaraya Technological University,
Bangalore, India

ABSTRACT

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software. The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers As it is one of the oldest area of research in machine learning it is used in major fields such as machine translation speech recognition and text processing. Different text and speech processing algorithm are discussed in this review paper and their working is explained with examples. Results of various algorithms show the development done in this field over past decade or so. We have tried to differentiate between various algorithms and also its future scope of research. The Gap analysis between different algorithms is mentioned in the paper as well as the application of these various algorithms is also explained. Natural language processing has not attained perfection till date but continuous improvement done is the field can surely touch the perfection line. Different AI now use natural language processing algorithms to recognize and process the voice command given by user.

Keywords: NLP, LSTM, PBMT, NMT

INTRODUCTION:

Andrew Ng has long predicted that as speech recognition goes from 95% accurate to 99% accurate, it will become a primary way that we interact with computers. The idea is that this 4% accuracy gap is the difference between annoyingly unreliable and incredibly useful. Thanks to Deep Learning, we‟re finally cresting that peak.

Nowadays artificial intelligence is widely discussed buzzword and is in under rapid development. Basically artificial intelligence is a computer program that can do something smart like a human, it is actually machine mimicking human to perform task in his absence and sometimes in better as well as efficient way, broadly speaking.

Machine learning is subset of AI. The intelligence of machine is improved using machine learning as through learning algorithms and analysis of different types of data. Deep learning and neural networks are subset of machine learning. Deep learning algorithms do analysis of different data sets through algorithm again and again and improves the machine knowledge according to the output obtained.

Natural language processing is an integral area of computer science in which machine learning and computational linguistics are broadly used. This field is mainly concerned with making the human and computer interaction easy but efficient. Machine learns the syntax and meaning of human language, process it and gives the output to user. The area of NLP involves making computer systems to perform meaningful tasks with the natural and human understandable language.

The reason why natural language processing is so important in future is it helps us to build models and processes which take chunks of information as input and in form of voice or text or both and manipulate them as per the algorithm inside the computer.

Thus the input can be speech, text or image where output of an NLP system can be processed Speech as well as Written Text.

Different algorithms developed to increase the efficiency of processing the language in text form which we are going to discuss here are:

- Long short term memory
- Sequence 2 Sequence model
- Named Entity Recognition model
- User preference graph model
- Word Embedding model
- Feature based sentence extraction using fuzzy inference rules.
- Template based algorithm using automatic text summarization

Similarly language can be processed even if the input is in speech form. For that various algorithms are developed and the best of them all are:

- Word Recognition
- Acoustic Modeling
- Connectionist temporal classification
- Phase based machine translation
- Neural machine translation
- Google neural machine translation

In this review paper different algorithms and models are discussed and various improvements done in field of natural language processing. We provide you a basic idea about all the algorithms mentioned above, like on what basis they work on, their efficiency and different applications where these can be implemented for the betterment of the society. We worked under the Department of computer science, SVKM‟s NMIMS shirpur to complete the review paper.

ALGORITHMS, MODELS AND APPROACH TO PROBLEM:

Text processing algorithms:

LSTM:

LSTM stands for long short term memory. Recurrent neural network is the primary element of LSTM model. Recurrent Neural network is chunk of neural network which can remember values.

LSTM are special kind of recurrent neural network which can remember previous input over arbitrary time interval and predict the output. It is used for training machine through input sets. It is one of the learning model in machine learning which is broadly used in Natural Language Processing. Stored values are not modified as learning proceeds. LSTM model is unable to edit the input sets but it can learn from it by computing its frequency according to the event by processing it several times. First step in LSTM is what data input is flushed out of the network. It is decided by forget gate layer „0‟ represents „completely forget‟ and 1 represent „completely keep‟. Next step decides what to store in cell state which is decided by input gate layer. The next layer called a tanh layer creates a vector of new candidate values which is combined to input values to update the state. Some LSTM model have an extra state called peephole connection which lets the gate to check status of cell state and before dropping the data from network.

Seq 2 Seq model:

The tradition seq2seq model contains two recurrent neural network i.e. encoder network and decoder network [2].

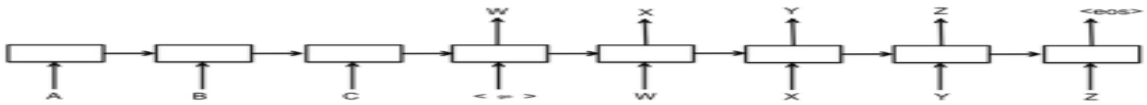

Fig 1: Recurrent neural network structure

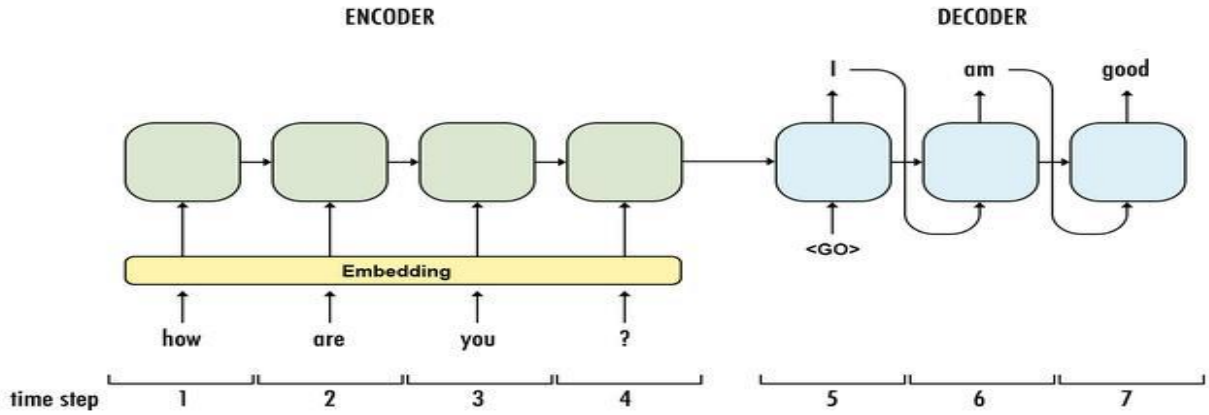

Fig 2: Encoder Decoder structure and working

Each box represents a RNN most commonly LSTM implemented RNN cell. In this model every input is encoded into fixed size vector which is later processing decoded using decoder. Vocabulary list is built and compiled using embedding so that the model can identify the correct grammar syntax. The vocabulary set is processed to check for the occurrences of the words and classify frequently used, rarely used and unique words in the vocabulary. The words are then replaced by with ids. Based on id‟s the reply suggestion is decoded and given as output. Following are some tags used in model while compiling the input.

EOS: End of sentence.
PAD: Filler.
GO: Start decoding.
UNK: Unknown; word not in vocabulary.
Following in the examples of Seq2seq model working:
Question: How are you?
Answer: I am fine.
This pair will be converted to:
Question: [PAD, PAD, PAD, PAD, PAD, PAD, “?”, “you”, “are”, “How”]
Answer: [GO, “I”, “am”, “fine”, “.”, EOS, PAD, PAD, PAD, PAD].

Named entity recognition Model:

As the name suggests named entity recognition is used to identify relevant names, classify name by the entity they belong to. NER model finds names places, people and other relevant entities in input data set which might be in text or speech form [3]. NER model works in two phases. First phase of NER model is to divide the text into segments or chunks to classify them. These chunks are classified in predefined categories such as name of person, organization, location etc. in form of tokens. The formatting is ignored like bolding and capitalization. Ex. $ mike^ (ENAMEX, name) who is a student of $ New York University^ (ENAMEX, org), $ New York university ^ (ENAMEX, location), scored $ 90% ^ (NUMEX, percent) in his seminar on the $ 19th of March ^ (TIMEX, date). In second phase of model. This model can be widely used in language and speech processing while user preference graph is to be created for smart reply of search suggestions.

User preference graph:

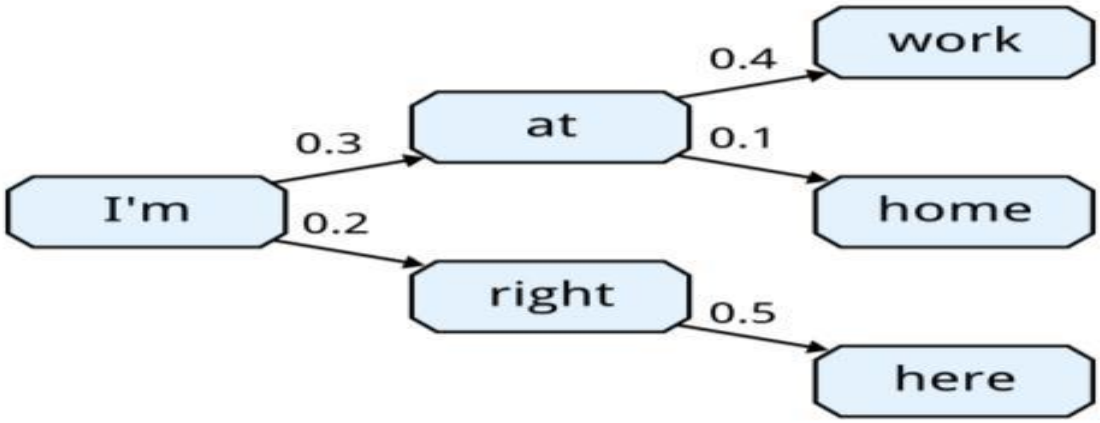

Fig 3: User preference graph example

User preference graph is used to create a set of user choices. When user repetitively choose specific tenses, adjectives, conjunctions and prepositions etc. a preference graph is created so that when user is using similar type of sentences then the model suggests the next words by calculating probability[4]. This words are mapped to each other hence a preference graph is created for particular user. On big scale implementation of this model, people with similar user preference graph are grouped together so that the suggestions can have wide scope and variety. This can be implemented in smart reply, smart suggestions in typing, auto reply systems etc.

Word Embedding:
Word embedding is derived from feature learning and language modeling in natural language processing where words and phrases are mapped onto vectors of real number of preference graph.

Phrase Based Machine Translation:

PBMT is one mode of Statistical Machine Translation. It uses predictive modeling to translate text [5]. These models are created with the help of or learned from bilingual large unstructured set of texts. With the help of these the most probable output is created.

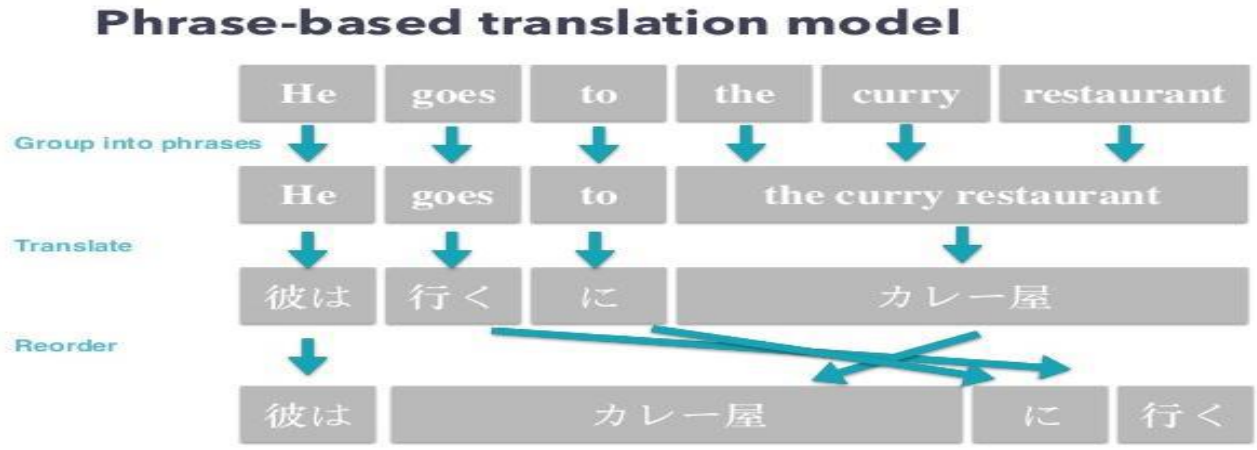

Fig 4: Figure based translation model

Algorithm for PBMT is as follows;
a) Breaking of Original Sentence into Chunks
b) Find all possible translations for each chunk: in this step with the help of the corpora we check how the humans translated the text in real world sentences.
c) Generate all possible sentences and find the most likely one: with the help of different combinations of translation in step b) we can generate more than 5000 combination of sentences.
d) Give the probable score by comparing it to training set. : Here the training set contains a large database of text from different books, articles, newspaper etc. By comparing each combination of the above step to the

training dataset we give it a probability score (likelihood score). After trying the different combinations and passing it through our training data set we will pick the one that has most likely chunk translation while also having a high likelihood score.

The disadvantage of PBMT [6] is that it is difficult to build and maintain. If there is a need to add a new language than bilingual corpora of that language should be present. For less famous language pair translation tradeoffs are made. That is if translation from Gujarati to Georgian it does not use a complex pipeline. Instead it may internally translate it to English and then translate it to Georgian. [7]

Neural Machine Translation (NMT):
Neural Machine Translation is newest method of Machine Translation. It creates much more accurate translations as compared to the Statistical Machine Translation. [5] Neural Machine Translation works by sending the input to different "layers" to be processed before output. NMT is able to use algorithm to learn linguistic rules on its own from Statistical Models. The NMT system [6] is based on attentional encoder-decoder and operates on sub word units. To improve the efficiency further, back-translations of the monolingual News corpus is used as additional training data. It is optimal for both direction translations. The strengths of NMT are that it can better handle verb order forms and avoid verb omission. It can handle English noun collection. Phrase Structure and articles are also well handled by NMT. [7] The limitations of NMT are ambiguous words into German. It also issues with forming verb continuous tenses. Dominant problem for NMT are prepositions.

Voice Processing Algorithms:

H. Acoustic Modeling:
contains the references of the individual sounds that make up a word. This individual sounds are then assigned a label. This label are known as phoneme. [8] A speech corpus and by using special training algorithm to create a statistical AQV presentations which represents each phoneme in a language.[9] These statistical representation are called as Hidden Markov Model (HMM). Each phoneme has its own HMM. Advantage of using Acoustic Modeling is that the users are motivated to articulate clearly, smartphones do high quality speech capture speech transferred to server error-free over IP. These models take lots and lots of training data to create and for many users they work just fine. However for many others they do not. This is because the data used to generate the model contains samples from tens of thousands of different speakers, so they are generic. Making specific models for individuals is not economical, neither is making models for accents with small populations. Acoustic models are a limitation of the technology.

A. Connectionist Temporal Classification (CTC):
Connectionist Temporal Classification is majorly used for training recurrent neural network (RNNs). One such example of recurrent neural network is LSTM Model. In speech recognition timing is major variable. The input is majorly similar to phenome but the timing may be varied. To overcome this problem where LSTM has issues with recognizing phonemes in speech audio. CTC work by output and scoring, thus being independent of the underlying network structure. CTC was first introduced in [9]. A CTC network has continuous output which is later fitted through training to model the probability of the label. The output are sequence of labels. If the sequence of labels differ only in length than they are considered equal. There are many combinations of equal labels. Thus making the scoring a non-trivial tasks. [10] Paper outlines a dynamic programming algorithm used to compute the sum of probabilities over all paths corresponding to a given labelling.

So the Algorithm is as follows;

a. Turning Sounds into Bits: This is called sampling. By Nyquist Theorem if we sample at least twice as fast as the highest frequency we can recover the original signal back. For speech recognition a sampling rate of 16,000 samples per second is optimal. After this there would be array of numbers with each number representing the sound wave‟s amplitude at 1/16000 a second intervals. We could directly give the sampled data to neural network but finding the pattern in such a large dump of data would be difficult and require lot of computations. Increasing the time complexity of the algorithm. So Pre Processing is done.

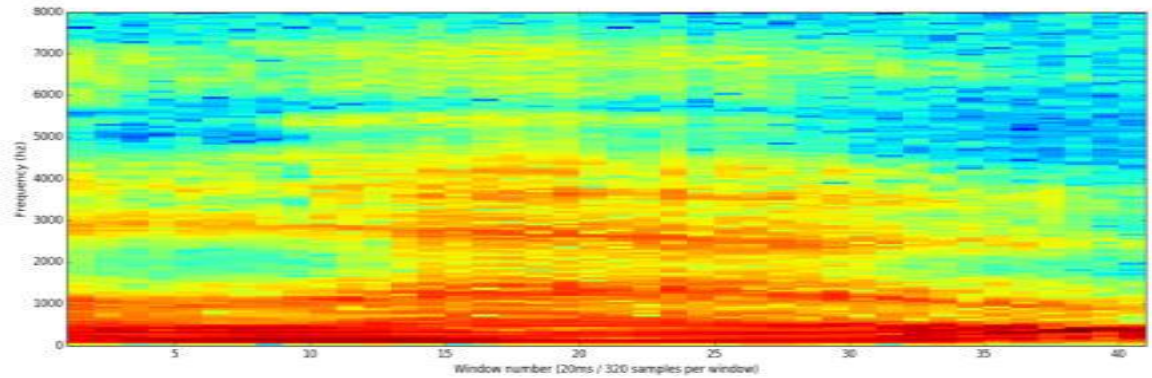

Fig 5: 20ms/320 samples per window

b. Preprocessing our Sampled Data: We reduce the time complexity of the algorithm by doing some preprocessing. This preprocessing includes grouping of our sampled audio into 20-millisecond long chunks. The below image shows first 20 milliseconds of audio of first 320 samples This short recording is also difficult to process since human speech comprises of complex sounds. There are low pitch sounds, mid-range speech sounds and even some high range speech sounds. To reduce the time complexity further we use Fourier Transform. Thus with the help of Fourier Transform the complex sound wave gets broken into simple sounds waves. We then add up how much energy is present in each one. A Spectrogram is created because for the neural network finding patterns in the spectrogram is far easier than finding the pattern in the raw sound files. Below is the representation of the above sampled data in spectrogram.

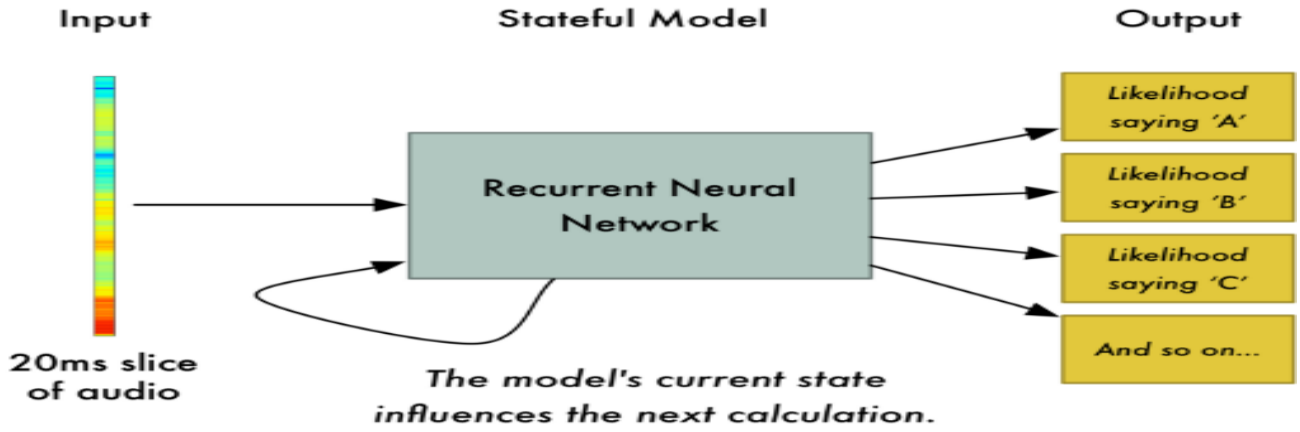

Fig 6: Working of Network for speech processing

d. Recognizing Characters from Short Sounds:

Since the input is straightforward to process. We can feed it to deep neural network. The input being the 20 millisecond audio chunks. For each input the neural network will try to figure out the phoneme. Since it is recurrent neural network the present output will also influences future predictions. For easy understanding we will consider that the input data is of person saying the word "HELLO". [11] So if the present neural network has recognized "HEL" so far, it"s very likely we will say "LO" rather than some random word such as "XYZ". Since CTC also deals with varying audio length. The output would be as HHHEE_LL_LLLOOO". HHHUU_LL_LLLOOO" AAAUU_LL_LLLOOO". We will then clean up the output by removing repeated characters. HHHEE_LL_LLLOOO becomes HE_L_LO HHHUU_LL_LLLOOO becomes HU_L_LO AAAUU_LL_LLLOOO becomes AU_L_LO Then we"ll remove any blanks: HE_L_LO becomes HELLO HU_L_LO becomes HULLO AU_L_LO becomes AULLO. Since all of the sounds similar to HELLO. So this pronunciation based predictions will be combined with likelihood scored based on large database of written text. Hence by this likelihood score of Hello will be greater than the other two. So the output would be shown correctly. [12] One disadvantage of this algorithm is that if the input audio file is of HULLO then the algorithm would not be able to recognize it correctly since the database of written text does not contain more number of HULLO. So the algorithm would malfunction when the reader says words which aren"t present in the database of written text.

Applications of NLP:

One of the application of Natural language processing which we are going to discuss here is summarization of text automatically with the help of software. We will also discuss two of the best algorithms which were designed to summarize the text and will also compare both of them so as to get to a conclusion.
But before we discuss about the algorithm it is better to know more about what automatic text summarization actually is.
Automatic text summarization is basically the task for a software to reduce a large amount of text into a meaningful short summary which allows the reader to understand what information the document contains in a short descriptive form so as it saves the efforts and time of the user.
There are mainly two general or fundamental ways to automatic summarization of text. Those are extraction and abstraction. In text summarization, Extractive methods work on choosing between a subset of words, phrases, or sentences present in the document in its original text to produce an extracted summary.[15]While in , abstractive methods the algorithm builds an internal semantic representation and then they use natural language generation techniques that is in the technique the machine acts as if it has a human brain and has the ability to generate a meaningful summary by understanding the text present in the document. This process creates a summary that is far closer to what a person might actually extract and present as a summary of text. This generated summary includes verbal innovations. Research by this date have focused primarily on extractive methods, which are pertinent for image collection summarization, text summarization and video summarization.

Feature based sentence extraction using fuzzy inference rules:
The stated algorithm is based on evaluating a sentence in the input data on basis of some rules which categorize the statement and assign those values as low, medium and high. This assignment of the values are done on the basis of rules which are total 8 in count and are also known as fuzzy rules or fuzzy logic. These rules are in an IF-THEN form. Like for an example the algorithm takes a statement F as an input and apply all the rules on it and assign values to it. Like IF(F1 is H) here it means the importance of the statement on the basis of first logic rule is „high‟ similarly all the rules are applied as (F2 is H), (F3 is M), (F4 is H), (F5 is M), (F6 if H), (F7 is H), (F8 is H)[13] and if the statement after being evaluated by the rules satisfies the criteria and is considered as important then it is added in the summary as per the same sequence as that was in the input data.
The algorithm consists of 4 stages which process the data and gives the final output as the processed summary. These stages are: first is Preprocessing then Feature extraction followed by Fuzzy logic scoring And Sentence selection and assembly
Each of these stages consists of sub processes and output of each stage is given as input to the next stage to process.
This algorithm used certain features to determine the importance of the sentence on which it is included into the summary. These features are [14]
1. Title feature
2. Term weight
3. Sentence length
4. Sentence position
5. Thematic word
6. And fuzzy logic

On the basis of these features the sentence is evaluated by the algorithm and the output is generated.

Template based algorithm for automatic text summarization:
As we saw that in feature based algorithm the sentences are evaluated on basis of some basic criteria or basic features and the sentences as arranged in the input data are added as same in the output summary while in case of template based extraction algorithm some modification is being done after extracting the text to arrange the information in a proper and grammatical way to arrange in the summary and to make it more look like a human work.
The template based algorithm for automatic text summarization is implemented in two phases [16]
A. Text pre-processing
B. And information extraction
A. Text pre-processing

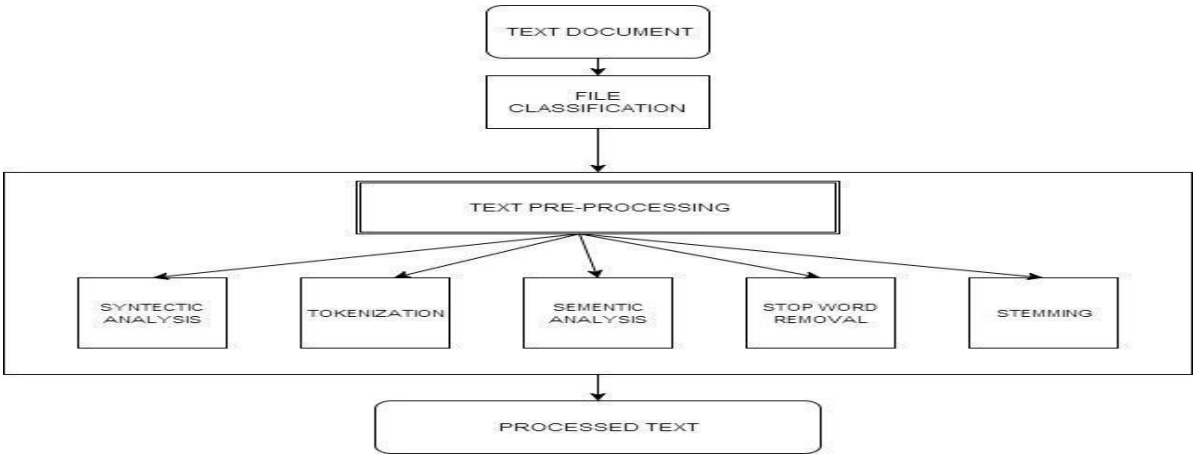

Fig 7: Text preprocessing model

This part of the implementation includes the module shown below:

1. Syntactic Analysis:

The job of syntactic analysis module is to decide the starting and ending of each sentence in the input document. Presently the algorithm take the full stop symbol as the ending of the sentence. And any string of characters up to the full stop symbol is taken as a one full sentence.

2. Tokenizer:

The job of tokenizer is to break the sentence given as the output from the syntactic analysis module into tokens. The broken pieces of a sentence can be words, numbers or punctuation marks.

3. Semantic:

Semantic analysis sub phase understands the role of every word which is in a sentence. Then assignment of a tag is done on every word named as noun, verb, adjective, and adverb and so on. This process of assigning and dividing the word into different classes is called Part-Of-Speech tagging or POS tagging

4. Stop Word Removal:

Some words are used more often in the natural language text but their value of importance in extracting meaning is very little in regards when we consider overall meaning of the sentence. Such words are stated as stop words and are removed

5. Stemming:

Stemming is the task of evaluating basic form of a certain word in the input text document. Same words are written in different tenses but in all having the same meaning thus to avoid that, stemming is done and these words having same meaning but different tenses are converted into basic simple tense.

B. Information extraction

This part of the algorithm includes the following modules:

1. Training the dialogue control
2. Knowledge based discovery
3. Dialogue management
4. Template based summarization

1. Training the dialogue control

During training, the system gain knowledge of important or index terms, named body like name of the persons, places, and temporal formats as per the norms of the rules. Intelligence and efficiency of the algorithm increases with every training set. That is every time we feed the data to the algorithm it stores the results of the process into a data storage and then uses them as reference or as experience while analysis the next input. The concepts learnt during training are stored in the knowledge base of the system.

2. Knowledge based discovery

Knowledge based discovery here means the process of extracting intelligent information and storing them in an unstructured text form. Thus decreasing the need to create multiple storage structures to store different terms of different category, thus reducing the search time and hence improving the overall performance of the algorithm.

3. Dialogue management

The dialogue management module is a process which offers human and computer interaction. With the help of this module the user can request the information he needs using normal natural language text. The dialogue control module which is built upon the training model as the experience data set, which accepts the user request, understands it, then refers to its knowledge base and then finally produces the answers which probably contains the sought information.

4. Template based summarization

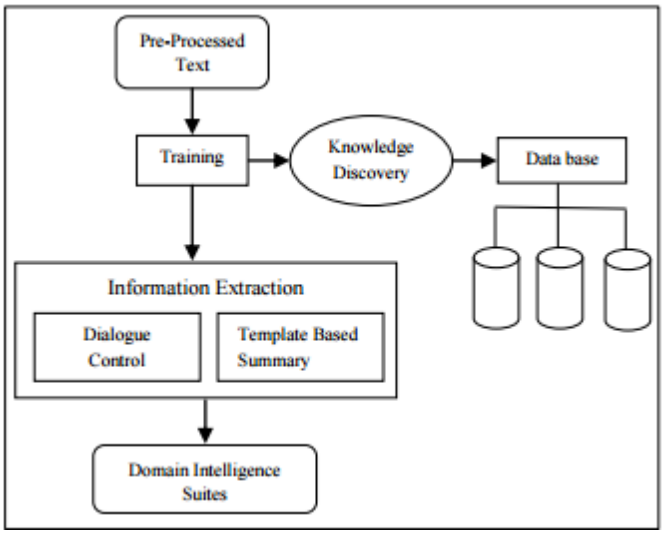

Fig 8: Information Extraction Module

It is the process of combining together all the meaningful text present in the input data or the document in a compact format.

The algorithm also allows user to prepare the template that has provisions to specify events, locations, named Entities etc. User also has the provision for specifying any number of and any type of POS patterns.

Gap Analysis:

In template based algorithm all the templates created on different documents by either same user or different user are being stored in the database for future references as it acts as training set for the software and hence increasing the accuracy of the system while in feature based algorithm there is no concept of database and thus no training set hence the gap is being fulfilled.[17]

CONCLUSION:

As above context indicate text processing algorithms are based on entity based classification and preference graphs. The text processing algorithms are used in smart reply and smart suggestions in various applications to reduce the user's workload and time giving appropriate and efficient output. Whereas in speech processing problem is nowhere near solved but it has improved a lot past decade. Neural and deep learning is used in text processing and speech processing to give a more efficient output. These recently improved algorithms have resulted in major breakthrough in this area. The accuracy level of output is close to perfect due to new improved algorithms which is now close to what humans would interpret. Various AI are developed based on text processing and speech processing algorithms to assess the user‟s requirement based on input classification. This improves results and user has more personalized result according to his needs. Combination of various text processing algorithms and speech processing algorithms are used for more refined output.

REFERENCES:

Adams Wei Yu, Hongrae Lee, Quoc V. Le “Learning to Skim Text”.
Alex Graves, Santiago Fernández et al. “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks” Pittsburgh, Pennsylvania, USA — June 25 - 29, 2006.
Brian Milch, Alexander Franz “Searching the Web by Voice” Taipei, Taiwan — August 24 - September 01, 2002.

Ciprian Chelba "Speech and Natural Language: Where Are We Now and Where Are We Headed?".
Dipanjan Das Andr´e F.T. Martins "A Survey on Automatic Text Summarization".
FangtaoLi§, YangGao† et al. "Deceptive Answer Prediction with User Preference Graph".
Grégoire Mesnil, Yann Dauphin et al. "Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding" IEEE Press Piscataway, NJ, USA Volume 23 Issue 3, March 2015.
Ian McGraw, Rohit Prabhavalkar et al. "Personalized Speech Recognition on Mobile Devices" Shanghai, China 19 May 2016.
Jan Chorowski, Navdeep Jaitly "Towards better decoding and language model integration in sequence to sequence models".
Luisa Bentivogli, Arianna Bisazza, and Mauro Cettolo "Neural versus Phrase-Based Machine Translation Quality: a Case Study".
Maja Popović "Comparing Language Related Issues for NMT and PBMT between German and English".
Matthew Henderson, Ramial-Rfou, Brian Strope etal "Efficient Natural Language Response Suggestion for Smart Reply".
Mr.S.A.Babar Prof.S.A.Thorat "Improving Text Summarization using Fuzzy Logic & Latent Semantic Analysis".
MR.S.A.Babar, MS.P.D.Patil "Fuzzy approach for document summarization".
Prashant G Desai, Saroja "A Study of Natural Language Processing Based Algorithms for Text Summarization" Devi Niranjan N Chiplunkar, Mahesh Kini M.
Prashant G.Desai, Sarojadevi,Niranjan N. Chiplunkar "A template based algorithm for automatic text summarization and dialogue management for text documents".
Yonghui Wu, Mike Schuster, Zhifeng Chen et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation".
