International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Analysis of Paraphrase Detection using NLP Techniques

Mrunal Badade[1], Vaibhav Adsul [2], Jagruti Thombare [3], Akshata Deshpande[4], Prof. Mansi Kulkarni[5]

[1,2,3,4]Student, Computer Engineering, Pillai College of Engineering, Maharashtra, India
[5]Faculty, Computer Engineering, Pillai College of Engineering, Maharashtra, India

---

Abstract - *Major difficulties faced in natural language processing are ambiguity where the same text has several possible interpretations. The paraphrase is a way of conveying the same content without compromising the meaning. It is an alternate form in the same language stating the same semantic content with the help of reframing or rearranging the phrases of a sentence. Paraphrases can occur at the word level, phrase level or even sentence level. Paraphrasing is of two types: Paraphrase Generation and Paraphrase Detection. We propose an application to detect the semantic similarity between two texts of the same language to establish the similarity. The proposed solution of tackling similar content or text can be used as an application to detect plagiarism as well as conduct an evaluation for a machine translation system. Not only Paraphrase Detection can be used to tackle the uniqueness of a text and retain its meaning, but also provide a measure to assess the Machine Translations of a text. Since, current available applications fall short to verify the integrity of a text if it is paraphrased and fails to mark it as plagiarized. We will be using already established traditional algorithms to detect if the content is a duplication of an already existing work and on top of that, we will be using our application to measure if the content has been paraphrased in any way and on the basis of the performance we would evaluate our system's efficiency compared to the state of the art systems that already exists.*

Keywords - *Ambiguity, Paraphrase Generation, Paraphrase Detection, Semantic similarity, Plagiarism, Machine Translations, Integrity.*

1. INTRODUCTION

Paraphrase detection, which means analyzing sentences that are semantically identical. We propose to detect the semantic similarity between two texts of the same language to establish the similarity. To find related sentences written in natural language is complex for various applications, like text summarization, plagiarism detection, information retrieval and question answering system etc. Realizing this gravity, we examine in particular how to mark the challenges with detecting paraphrases using Natural Language Processing Techniques. We propose the multi-head attention mechanism helps the model learn the words relevant information in different presentation subspaces.

1.1 Fundamentals Natural

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. [1]

Challenges in NLP often refer to natural language understanding, speech recognition, and natural language generation.

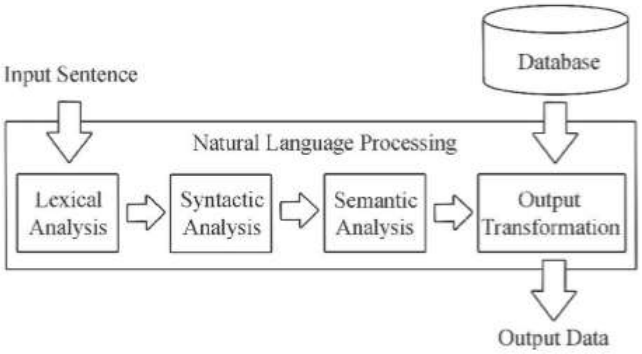

Fig-1: NLP Processing

Figure 1 shows the basic processing of an input or source sentence in natural language which is done by the machine.

1.2 Terminologies

- Tokenization -Tokenization is, generally, an early step in the NLP process, a step which splits longer strings of text into smaller pieces, or tokens. Larger chunks of text can be tokenized into sentences, sentences can be tokenized into words, etc. Further processing is usually achieved after a piece of text has been properly tokenized.
- Normalization - Before processing ahead, text must be normalized. Normalization usually refers to a sequence of related tasks involved to put all text on a level playing field: converting all text to the same case (upper or lower), eliminating punctuation, increasing contractions, converting numbers to their word identification, and so on. Normalization brings all words on the same footing, and allows processing to continue consistently.

- Stemming - Stemming is the process of discarding affixes from a word in order to gain a word stem.
- Corpus -In linguistics and NLP, corpus refers to a collection of texts. Such collections may be built on a single language of texts, or can span different languages -- there are various reasons for which multilingual corpora may be useful. Corpora may also subsist of the media texts. Corpora are frequently individually used for statistical linguistic analysis and hypothesis testing.
- Stop Words - Stop words are those words which are clean out of text, since these words share limited to, overall meanings, given that they are usually the most familiar words in a language. For example, "the," "and," and "a," while all needed words in an appropriate passage, don't usually contribute greatly to one's understanding of content.
- Parts-of-speech (POS) Tagging- POS tagging subsists of appointing a category tag to the tokenized parts of a sentence. The most famous POS tagging would be classified words as nouns, verbs, adjectives, etc.
- Bag of Words- Bag of words is a special representative model used to clarify the details of a selection of text. The bag of words model discards grammar and word order, but is attentive in the number of existence of words within the text. The entire illustration of the document selection is that of a bag of words.

1.3 Similarity Measures

There are various similarity measures which can be enforced to NLP. What are we measuring the similarity of? Mostly, strings.

- Levenshtein - The number of characters that must be deleted, inserted, or substituted in order to make a pair of strings equal.
- Jaccard - The limit of overlay between 2 sets; in the case of Natural language processing (NLP), generally, documents are sets of words.
- Smith Waterman –Smith Waterman is identical to Levenshtein, but with costs appointed to substitution, insertion, and deletion.

1.4 Syntactic Analysis

Also introduced as parsing, syntactic analysis is the function of analyzing strings as symbols, and providing them according to a well-established set of grammatical rules. This step must, out of necessity, come before any further analysis, which attempts to extract insight from the text--semantic, sentiment, etc. -- treating it as something beyond symbols.

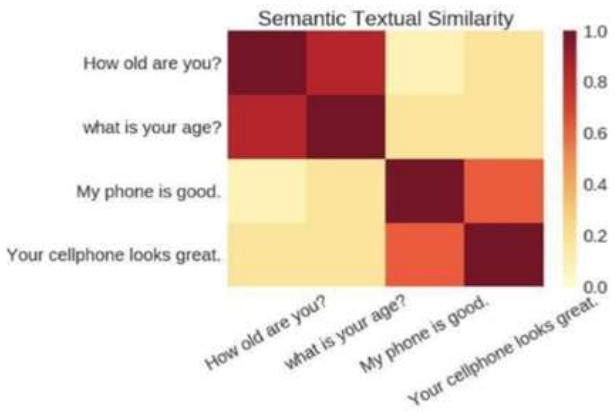

Fig-2: Semantic Textual Similarity

2. Proposed System Architecture

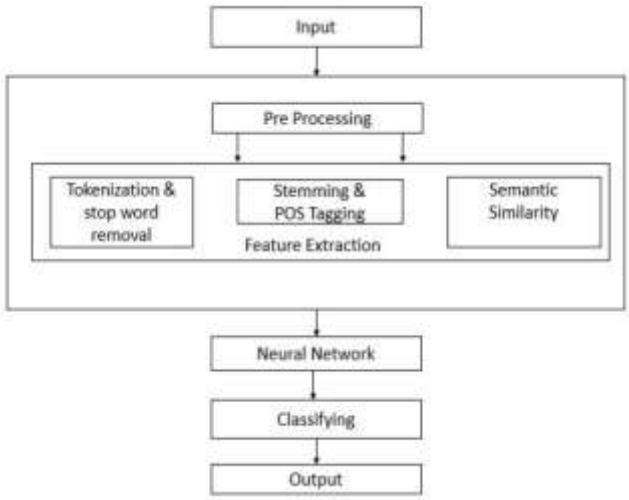

Fig-3: Proposed System Architecture

- Tokenization

Tokenization is the procedure of splitting up the given text into units called tokens.

Algorithm:

Input: A single sentence

The first color that is uppermost color in the flag which is the saffron color, stands for purity. The second color i.e. the middle color in the flag is the white color and it stands for peace. The third color that is the lowest color in the flag is the green color and it stands for fertility. The white color has an Ashoka Chakra of blue color on it. Ashoka Chakra contains twenty-four spokes which are equally divided. India has 29 states and 7 union territories.

Output: A list of tokens

'The', 'first', 'color', 'that', 'is', 'uppermost', 'color', 'in', 'the', 'flag', 'which', 'is', 'the', 'saffron', 'color', ',', 'stands', 'for', 'purity', '.' 'The', 'second', 'color', 'i.e', ',' 'the', 'middle', 'color', 'in', 'the', 'flag', 'is', 'the', 'white', 'color', 'and', 'it', 'stands', 'for', 'peace', '.' 'The', 'third', 'color', 'that', 'is', 'the', 'lowest', 'color', 'in', 'the', 'flag', 'is', 'the', 'green', 'color', 'and', 'it', 'stands', 'for', 'fertility', '.' 'The', 'white', 'color', 'has', 'an', 'Ashoka', 'Chakra', 'of', 'blue', 'color', 'on', 'it', '.' 'Ashoka', 'Chakra', 'contains', 'twenty-four', 'spokes', 'which', 'are', 'equally', 'divided', '.' 'India', 'has', '29', 'states', 'and', '7', 'union', 'territories', '.'

- Stopwords

Stop words are words which are cleaned out before or after preparing natural language data (text).
Algorithm:

1.Input
2.if words in sentence == stopwords list then goto step-4 3. else message ("No stopwords") then goto step-4
4.output
5.Exit Stopword

Input:

The first color that is uppermost color in the flag which is the saffron color, stands for purity. The second color i.e. the middle color in the flag is the white color and it stands for peace. The third color that is the lowest color in the flag is the green color and it stands for fertility. The white color has an Ashoka Chakra of blue color on it. Ashoka Chakra contains twenty-four spokes which are equally divided. India has 29 states and 7 union territories.

Output:

The first color **that is** uppermost color **in the** flag **which is the** saffron color, stands **for** purity. The second color i.e. **the** middle color **in the** flag **is the** white color **and it** stands **for** peace. The third color **that is the** lowest color **in the** flag **is the** green color **and it** stands **for** fertility. The white color **has an** Ashoka Chakra **of** blue color **on it**. Ashoka Chakra contains twenty-four spokes **which are** equally divided. India **has** 29 states **and** 7 union territories.

- Stemming

A stemming is a procedure in which the various forms of similar words are reduced to a frequent form.
Algorithm:
1.Input
2.if words suffix in sentence == suffix list then goto step-4 3. else message ("No stopwords") then goto step-4
4.output
5.Exit

Input:

Programmers program with programing languages. Studies studying cries cry

Output:

```
Programers  :  Program
program  :  program
with  :  with
programing  :  program
languages  :  languag
.  :  .
Studies  :  Studi
studying  :  studi
cries  :  cri
cry  :  cri
```

- POS Tagging

A POS Tagger is a sample of software that study text in some language and assign POS to each word, such as noun, verb, etc., despite usually computing operations use moreover fine-grained Part of speech tags like 'noun-plural'.
Input:

Rajasthan itself has a glorious history. It is famous for many brave kings, their deeds, and their art and architecture.

Output:

[('Rajasthan', 'NNP'),('itself', 'PRP'),
('has', 'VBZ'),('a', 'DT'),
('glorious', 'JJ'),('history', 'NN'),
('.', '.'), ('It', 'PRP'),
('is', 'VBZ'),('famous', 'JJ'),
('for', 'IN'), ('many', 'JJ'),
('brave', 'VBP'),('kings', 'NNS'),
(',', ','),('their', 'PRP$'),
('deeds', 'NNS'), (',', ','),
('and', 'CC'),('their', 'PRP$'),
('art', 'NN'),('and', 'CC'),
('architecture', 'NN'), ('.', '.')]

- Siamese Deep Neural Networks for semantic similarity (SNN)

A Siamese neural network (SNN) is an artificial neural network that uses equal pressure while working in tandem on two distinct input vectors to compute proportional output vectors. Generally, one of the output vectors is precomputed, thus constructing a baseline across which the other output vector is correlated.[3] This is identical to match fingerprints, but can be expressed more technically as a distance function for locality-sensitive hashing.

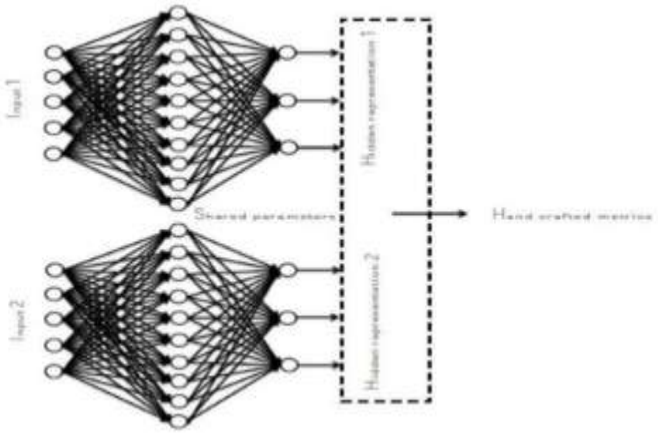

Fig-4: Siamese Deep Neural Network

SNN is a class of neural network architectures that consist of two or more duplicate subnetworks. Duplicate here means they have a similar configuration with a similar framework and density. Framework renewing is illustrated over both subnetworks. Siamese NNs are famous among functions that require finding the same or a relation between two proportionate things. Some examples are paraphrased scoring, where the inputs are two sentences and the output is a score of how similar they are; or signature verification, where they figure out whether two signatures are from the same person. Generally, in such tasks, two identical subnetworks are used to process the two inputs, and another module will take their outputs and produce the final output.

- Multi-head Attention Network

Attention takes two sentences, changes them into a matrix where the altercation of one sentence form the columns, and the altercations of another sentence form the rows, and then it makes correlations, finding related context. This is especially helpful in machine translation. You don't just have to use attention to correspond meaning between sentences in two distinct languages. You can also put the similar sentence simultaneously the columns and the rows, in order to figure out how few parts of that sentence refer to others. So, study allows you to view at the completion of a sentence, to make relations between any specific word and its related context. This is highly diverse from the small-memory, upstream-focused RNNs, and also quite specific from the proximity-focused convolutional networks.

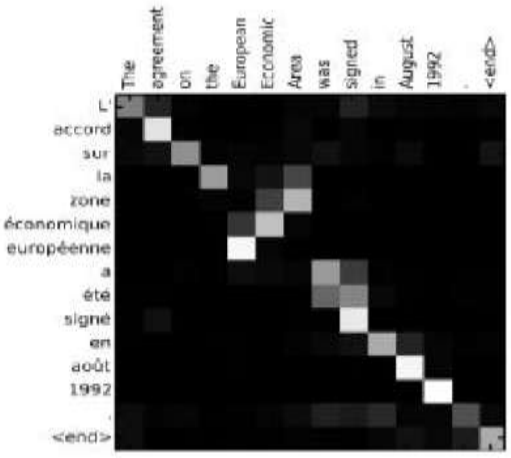

Fig-5: Multi-head Attention Network

3. Sample Dataset Used

Table -1: Sample Dataset Used for Experiment

Dataset	Source	Items	Type
SNLI	The Standford Natural Language Processing Group	570,152 Sentence pairs	Corpus
QQP	Quora Question Pairs	400,000 Sentence pairs	Corpus

- Corpus

Corpus is a collection of recorded utterances used as a basis for the descriptive analysis of a language. A corpus may contain texts in a single language (monolingual corpus) or text data in multiple languages (multilingual corpus).

Multilingual corpora that have uniquely formatted for side-by-side identification are known as aligned parallel corpora. There are two prime forms of parallel corpora which have texts in two languages. In a translation corpus, the texts in one language are translations of texts in the other language. In a comparable corpus, the texts are of the same kind and cover the same content, but they are not translations of each other.[6] To utilize a parallel text, some kind of text arrangement classifying identical text portions is essential for analysis. Machine translation methods for converting between two languages are usually prepared using parallel fragments composing a first and a second language corpus which is an element-for-element conversion of the first language corpus.

4. LITERATURE SURVEY

1. Paraphrase Detection in Hindi Language using Syntactic Features of Phrase by Rupal Bhargava, Anushka Baoni, Harshit Jain & Yashvardhan Sharma in 2016.
   Observations: A feature vector-based approach with 3 options (POS Tags, Word Stems and Soundex codes) is mentioned for paraphrase detection of Hindi Language. when extracting these 3 options, the python package "fuzzywuzzy" is employed to calculate the similarity scores. Levenshtein Distance wants to calculate the variations between string sequences.

2. Malayalam Paraphrase Detection by Sindhu.L & Sumam Mary Idicula in 2016.
   Observations: South Dravidian is employed as input language. The projected linguistics approach for distinguishing the paraphrases includes 3 phases – matching identical tokens, matching lemmas & matching with synonyms replaced. Similarity comparison is performed at the sentence level mistreatment the Jaccard, Containment, Overlap and cos similarity metrics.

3. Learning Paraphrase Identification with Structural Alignment by Chen Liang, Praveen Paritosh, Vinodh Rajendran, and Kenneth D. Forbus in 2016.
   Observations: Planned a brand-new alignment-based approach to find out linguistics similarity of texts. Used a hybrid illustration, attributed relative graphs, to inscribe native and structural data.

4. Convolutional Neural Network for Paraphrase Identification by Wenpeng Yin & Hinrich Schutze in 2015.
   Observations: Conferred the deep learning design "Bi-CNN-MI" for paraphrase identification (PI). Bi-CNN-MI: "Bi-CNN" stands for double CNNs, "MI" for multi granular interaction options. supported the insight that PI needs examination 2 sentences on multiple levels of coarseness, we have a tendency to learn multi granular sentence representations exploitation convolutional neural network (CNN) and model interaction options at every level.

5. Paraphrase Detection Based on Identical Phrase and Similar Word Matching by Hoang-Quoc Nguyen-Son, Yusuke Miyao, & Isao Echizen in 2015.
   Observations: Developed a similarity matching (SimMat) metric for police work paraphrases that's supported matching identical phrases and similar words and quantifying the minor words. It's calculated by matching identical phrases and similar words. Phrase-by-phrase matching is completed employing a "Heuristic algorithm". Word matching is completed using the "Kuhn-Munkres algorithm".

6. Paraphrase Detection Using Recursive Autoencoder by Eric Huang in 2011.
   Observations: The autoencoder learning algorithmic rule is an associate degree approach to mechanically extract options from inputs in an associate degree unattended manner. Applied autoencoders recursively to the input. algorithmic Autoencoder "RAE" used. For the combination options, SVMs are used as the classifier.

7. A Semantic Similarity Approach to Paraphrase Detection by Samuel Fernando & Mark Stevenson in 2009.
   Observations: Planned technique makes use of WordNet-based lexical similarity measures applied otherwise from previous approaches. The system was evaluated on the Microsoft analysis Paraphrase Corpus and located to shell antecedently reportable approaches.

8. A Metric for Paraphrase Detection by Joao Cordeiro, Gael Dias, & Pavel Brazdil in 2007.
   Observations: Planned a brand-new metric, the Sumo-Metric, for finding paraphrases. "Sumo-Metric" outperforms all progressive metrics over all corpora in conducted experiments by the authors. conjointly performed a comparative study between already existing metrics and new tailored ones and planned a brand-new benchmark of paraphrase take a look at corpora.

9. Methods for Detecting Paraphrase Plagiarism by Victor U Thompson & Chris Bowerman in 2018.
   Observations: To sight lexical substitutions, a mixture of question growth and word try similarity measure mistreatment WordNet and therefore the word2vec word embedding model is employed. question growth is employed to come up with synonyms for every question word in a very suspect sentence. The word2vec model transforms the synonyms into word vectors and compares them with word vectors of the supply sentence mistreatment cos similarity.

10. A Survey on Paraphrase Detection Techniques for Indian Regional Languages by Shruti Srivastava & Sharvari Govilkar in 2017.
    Observations: Techniques like Vector based mostly "Bi-CNN-MI" approach & MT approach "SimMat Metric, grappling Metric" yield the foremost effective results for paraphrase detection in West Germanic.

11. The Study and Review of Paraphrase Detection Techniques in Machine Learning by Darshana S Bhole, Sandip S. Patil in 2017.
    Observations: Stemmer algorithm used for removing the suffixes. The 2 sentences paraphrased or not is calculated by token count, token matching and synonyms token matching done by the classifier.

12. Detecting Paraphrases in Tamil Language Sentences by Dr.S.V.Kogilavani1, Dr.R.Thangarajan, Dr.C.S.Kanimozhiselvi, Dr.S.Malliga in 2017.
    Observations: The sentences area unit classified mistreatment 2 supervised machine learning algorithms like SVM and goop Entropy utilize sixteen completely different syntactical and linguistics options to best represent the similarity between sentences.

13. Detecting Paraphrases in Indian Languages Using Multinomial Logistic Regression Model by Kamal Sarkar in 2016.
    Observations: The foremost usually used corpora for paraphrase detection is the MSRP corpus (Microsoft Research Paraphrase Corpus). "Word2Vec" model (Python) accustomed to sight linguistics. Similarity between word vectors for the words.

14. Overview of Shared Task on Detecting Paraphrases in Indian Languages (DPIL) by M. Anand Kumar, Shivkaran Singh, Kavirajan B & Soman K P in 2016.
    Observations: Four Indian languages (Hindi, Punjabi, Tamil and Malayalam) thought of. No annotated corpora or automatic linguistics

interpretation systems like MSRP offered for Indian languages thus Created benchmark knowledge for paraphrases. Vocabulary size for Hindi & Punjabi languages is a smaller amount than Tamil and Malayalam. Tamil and Malayalam are extremely agglutinative in nature. Tamil & Malayalam language accuracy is low as compared to the accuracy obtained by Hindi & Punjabi language.

15. A Novel Approach to Paraphrase Hindi Sentences using Natural Language Processing by Nandini Sethi, Prateek Agrawal, Vishu Madaan & Sanjay Kumar Singh in 2016. Observations: Input taken as "Hindi" sentence. Segmentation, Parsing & linguistics Analysis performed step by step. Applied Reframing Rules with the assistance of "Synonym Replacement" & "Antonym Replacement" Combined the results to make a new paragraph. Technique used was the "N-Gram" technique.

5. CONCLUSION

Given a set of two sentences, finding semantic similarity between those sentences is a difficult task as it requires analyzing a language which the system does not understand. (Human natural language). Paraphrase Detection has multiple real-world applications that solve real life problems as discussed earlier. We propose a system to detect paraphrasing of sentences in natural language using SNN in TensorFlow. By training the system in its initial phase via deep learning architecture such as CNN, RNN & MAN, pre-processing the sentence to extract features & derive results through a classifier in testing phase.

ACKNOWLEDGEMENT

We have taken efforts in building this paper. However, it would not have been possible without the kind of support and encouragement from many individuals and college. We would first like to extend our sincere thanks to all of them.
A hearty thanks to our project supervisor Prof. Manasi Kulkarni for the constant supervision and guidance in all difficulties faced during the project and report making.

We are thankful to H.O.D of the Computer Department Dr. Sharvari Govilkar for providing the necessary information and resources.

We are highly indebted to the Principal Dr. Sandeep Joshi for the constant motivation and encouragement which helped a lot in completion of this report.

We would like to express our gratitude towards our parents for their kind cooperation. We would also like to thank the authors whose books we have referred to in making the project and report.

REFERENCES

[1] Natural language processing. https://en.wikipedia.org/wiki/Natural_language_processing

[2] Victor U Thompson & Chris Bowerman, (2018) “Methods for Detecting Paraphrase Plagiarism”.

[3] Domain generalization in sketch-based image retrieval systems. http://reports.ias.ac.in/report/20604/domain-generalization-in-sketch-based-image-retrieval-sbir-systems

[4] Shruti Srivastava & Sharvari Govilkar, (2017) “A Survey on Paraphrase Detection Techniques for Indian Regional Languages”.

[5] Darshana S Bhole, Sandip S. Patil, (2017) “The Study and Review of Paraphrase Detection Techniques in Machine Learning”.

[6] Text corpus. https://en.wikipedia.org/wiki/Text_corpus

[7] Dr.S.V.Kogilavani1, Dr.R.Thangarajan, Dr.C.S.Kanimozhiselvi, Dr.S.Malliga, (2017) “Detecting Paraphrases in Tamil Language Sentences”.

[8] Kamal Sarkar, (2016) “Detecting Paraphrases in Indian Languages Using Multinomial Logistic Regression Model”.

[9] Nandini Sethi, Prateek Agrawal, Vishu Madaan & Sanjay Kumar Singh, (2016) “A Novel Approach to Paraphrase Hindi Sentences using Natural Language Processing”.

[10] M. Anand Kumar, Shivkaran Singh, Kavirajan B & Soman K P, (2016) “Overview of Shared Task on Detecting Paraphrases in Indian Languages (DPIL)”.

BIOGRAPHIES

Mrunal Badade, Student of Pillai College of Engineering, completing last year of my degree in Computer Engineering department.

Vaibhav Adsul, Student of Pillai College of Engineering, completing last year of my degree in Computer Engineering department.

Jagruti Thombare, Student of Pillai College of Engineering, completing last year of my degree in Computer Engineering department.

Akshata Deshpande, Student of Pillai College of Engineering, completing last year of my degree in Computer Engineering department.

Prof. Mansi Kulkarni, Faculty of Pillai College of Engineering. Assistant professor in department of Information Technology.