

Text Comparison Based on Semantic Similarity

Sonali Mhatre
*Department of Information technology
Bharati Vidyapeeth College of
Engineering
Navi Mumbai, India
sonalinmhatre@gmail.com*

Aditi Hire
*Department of Information Technology
Bharati Vidyapeeth college of
Engineering
Navi Mumbai,India
adithire2@gmail.com*

Shilpa Satre
*Department of Information Technology
Bharati Vidyapeeth College of
Engineering
Navi Mumbai,India
shilpa.m.shelar@gmail.com*

Aniket Itankar
*Department of Information Technology
Bharati Vidyapeeth College of
Engineering
Navi Mumbai,India
aniketitankar01@gmail.com*

Mansi Hajare
*Department of information technology
Bharati Vidyapeeth College of
Engineering
Navi Mumbai,India
mansihajare3@gmail.com*

Shruti Patil
*Department of Information Technology
Bharati Vidyapeeth College of
Engineering
Navi Mumbai,India
shrutipatil1303@gmail.com*

Abstract—Main objective of the research is to develop a method for computing text similarity incorporating both statistical techniques and semantics. The method uses algorithms, libraries, and semantic models to analyze the word usage patterns and frequency of the texts, and can model human common sense knowledge through the use of multiple algorithms. One strength of the method is its adaptability to different domains, because of the incorporation of similar semantics. Users can choose from a range of methods within the implementation, giving them flexibility in their text analysis. The research is about developing a promising method for text similarity analysis that combines the strengths of statistical and semantic approaches. The stated technique is demonstrated to get the mapping of Course Objective sentences and Programme Outcome sentences based on similar semantics as required by National Board of Accreditation.

Keywords—Text Similarity , Corpus , Similar Semantic , Natural Language Processing (NLP) , Cosine Similarity , TF-IDF Vectorizer , Count Vectorizer , Sen2Vec , word2vec.

I. INTRODUCTION

The field of natural language processing (NLP) has made remarkable progress recently, particularly in the area of text similarity. Text similarity is the measure of how comparable two or more pieces of text are in terms of their content and structure. Text similarity has numerous practical applications, including detecting plagiarism, clustering documents, and optimizing search engines. This project will create a text similarity algorithm based on similar semantics. Similar semantics refers to the analysis of collections of text data to extract patterns and relationships between words and phrases. Gathering of corpus of text data is done, then it is processed, and extracted relevant statistics that will be used to design the algorithm. Statistical techniques are employed to evaluate the frequency and usage patterns of words and phrases in the texts to determine their similarity. The accuracy and effectiveness is assessed of the algorithm using a variety of text samples. The outcome of this project has the potential to advance the development of more precise and efficient text similarity algorithms, which can have significant practical applications across various domains.

II. REVIEW OF LITERATURE

In [4], the proposed methodology utilizes a custom word embedding model to identify groups of similar maintenance jobs by leveraging unstructured textual information. This

model combines two sources of information: maintenance records and industrial taxonomy. By effectively learning the word distribution using information from both semantic and taxonomic sources, a weighting parameter governs the learned representation. The proposed methodology outperformed other models in terms of identifying clusters.

A method for identifying and characterising lexical semantic change based on Gaussian word embedding (w2g) is presented in Paper [3]. (LSC). Three stages make up the LSC analysis with w2g approach. A kernel modelled after the n-gram-word2vec implementation is used in the first step. In the subsequent stage, embedding from various time periods are subjected to the vector alignment process. Semantic change identification is carried out during the third and last step using thresholding and similarity assessments.

In [2] researchers introduce Write Better, a corpus-based writing assistant designed to help users improve their writing by providing word use examples from real contexts. It can be integrated into popular writing tools such as Microsoft Word, Google Docs, and Overleaf. The aim of corpus-based assistants is to assist users in generating ideas, choosing words, and completing ideas, ultimately improving the composition of a text. The article introduces Write Better, a corpus-based writing assistant designed to help users improve their writing by providing word use examples from real contexts. It can be integrated into popular writing tools such as Microsoft Word, Google Docs, and Overleaf. The aim of corpus-based assistants is to assist users in generating ideas, choosing words, and completing ideas, ultimately improving the composition of a text. Effective communication when presenting Write Better to users is critical. Since users tend to scan web pages rather than reading them in full, presenting the benefits and use guidelines of Write Better in a clear and concise manner is essential. While the tool has the potential to be a useful aid for writers, it is important to ensure that it is utilized effectively.

The paper [1] describes the development of a domain-specific word embedding model for representing food items, using a textual corpus of recipe cooking preparations. Unlike previous approaches that relied solely on ingredient lists, the authors trained their model on recipe steps to capture the context and food descriptions associated with each ingredient. They then combined the word embedding model

with a fuzzy-based document distance to identify the most similar food elements for an adaptation task. The results of their study suggest that this unsupervised method is effective in determining the most appropriate ingredients for each recipe style. Overall, the authors' approach has potential applications in recipe recommendation systems and personalized nutrition advice.

This article [5] addresses research that attempts to create a framework for creating a parallel corpus for sign languages that can be used for creating and analysing machine translation models for translating from natural language to sign language and vice versa. The study suggests a method for creating a parallel corpus of words for various sign language dialects as well as a corpus of source and translated phrases at the sentence level. The system involves a wide range of stakeholders in creating and validating the repository in a quality-controlled manner by utilising crowdsourcing and editorial management. The research has successfully applied the suggested framework and created a parallel corpus at the word level that includes gestures for about 700 words in Pakistan Sign Language as well as a corpus at the sentence level that includes more than 8000 sentences in various tenses. This work has significant potential to improve the accessibility of sign languages for the deaf and mute community by enabling the development of machine translation models and avatar-based videos for different dialects of sign languages.

III. PROBLEM DEFINITION

National Board of Accreditation mandates the mapping of Course Objectives with Programme Outcomes. Based on human cognition every institute/department has their own set of mapping, which at times becomes difficult to validate. This research tries to automate the above mentioned process using Natural Language Processing. Text similarity algorithms have been proven handy in solving these automation problem Current text similarity methods, such as edit distance, city block distance, Euclidean distance have limitations in capturing the semantic and contextual meaning of the texts being compared. Traditional methods do not take into account the usage patterns and relationships between words and phrases in large collections of text data, which can lead to inaccurate similarity scores. To address this issue, similar semantics-based approaches have been proposed, but there is a need for a more robust algorithm that can accurately measure the similarity between pieces of text. This model aims to use text similarity algorithm based on similar semantics that can accurately measure the similarity between pieces of text by utilizing advanced semantics techniques to extract relevant patterns and relationships between words and phrases in large collections of text data. It will have significant practical applications in various domains, such as plagiarism detection, document clustering, and search engine optimization.

IV. METHODOLOGY

Whenever user wants to find out the similarity of one sentence with respect to multiple sentences, so two .csv files will be provided as input. Both .csv files will contain number of sentences and one sentence from first csv file is compared to all the sentences of second .csv file and similarity is obtained in the form of matrix. Also as an output .csv file of similarity matrix is obtained.

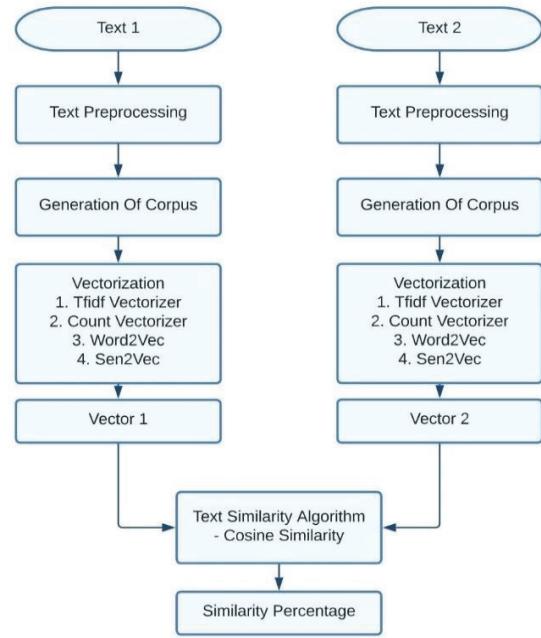


Fig. 1. Working Process

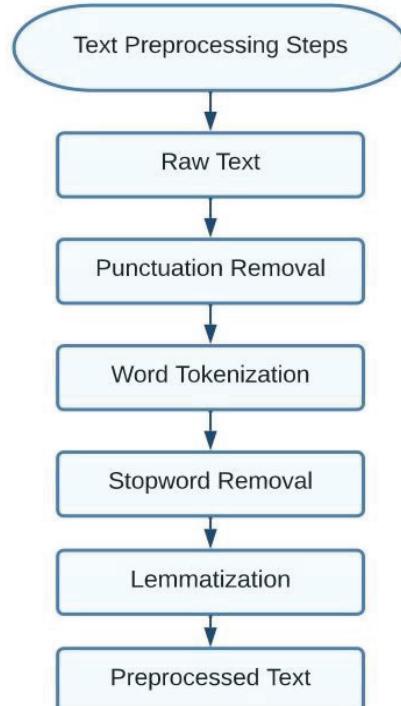


Fig. 2. Text Pre-processing Module

One sentence is selected from each of the CSV files, and the test pre-processing steps were applied to both sentences. A corpus is generated from the pre-processed sentences, and vectorization is performed on the corpus to obtain vectors for both sentences. A similarity algorithm is applied to these vectors, and the resulting similarity percentage is obtained, refer below diagram [Fig. 1, Fig. 2] to understand the above process.

V. TECHNIQUES USED

A. TF-IDF Vectorizer

TF-IDF is a widely used technique for converting textual information into a numerical representation that can be leveraged for machine learning models. Specifically, it is an algorithm that determines the significance of each word in a given document or a collection of documents. This is done by computing a statistical value that measures how important a word is to a document or corpus.

The TF component of TF-IDF represents the frequency of a word in a document relative to the total number of words in the document. This provides an indication of how often the word appears within that particular document. The IDF component of TF-IDF, on the other hand, indicates the rarity of the word in the corpus by calculating the logarithm of the total number of documents in the corpus divided by the number of documents containing the word. This component provides an indication of how unique or uncommon a word is within the corpus as a whole. By combining these two components, TF-IDF is able to provide a measure of the importance of a word to a document or corpus, making it a useful tool for various text-based applications.

B. Count Vectorizer

Count Vectorizer is a useful NLP tool provided by the scikit-learn library in Python that enables the transformation of text data into a numerical representation. The technique used by Count Vectorizer involves converting a collection of text documents into a matrix consisting of word counts. In this matrix, each row represents a single document, while each column corresponds to a unique word in the entire collection. The value in each cell of the matrix is the count of how many times the respective word appears in the corresponding document. Common applications of Count Vectorizer include sentiment analysis, text categorization, and topic modeling, among other NLP tasks.

C. Word2Vec

The word2vec algorithm is a technique used to understand and learn word associations from a large corpus of text data. This algorithm employs a neural network model, which is trained on the corpus to capture the relationships between words. Once trained, the model can recognize synonyms or suggest additional words to complete a partial sentence.

In the word2vec model, each distinct word in the corpus is represented as a vector, which is essentially a list of numbers. These vectors are carefully chosen to capture both the semantic and syntactic properties of words. By using these vectors, a mathematical function such as cosine similarity can be applied to determine the degree of semantic similarity between words.

By using the word2vec algorithm, we can gain insight into the meaning of words and how they relate to each other in a given context. This technique has become a popular tool for natural language processing and can be used in various applications such as text classification, sentiment analysis, and language translation.

D. Sent2Vec

Sent2Vec is a versatile NLP model used to create sentence embeddings for various NLP applications, such as sentiment analysis, text classification, and information

retrieval. The model operates by constructing sentence embeddings through the combination of word embeddings, where each word is represented as a dense vector in a high-dimensional space.

To apply Sent2Vec in Gensim, certain modifications would need to be made to the existing Word2Vec and Fast Text models to accommodate sentence-level input. This would involve adjusting the training objective to produce sentence embeddings rather than word embeddings, as well as revising the input data to include sentences instead of individual words. Specifically, the training algorithm would need to calculate the sentence embedding by averaging the corresponding word embeddings for each sentence.

E. Cosine Similarity

Cosine similarity is a widely used metric for determining the similarity between two pieces of text or documents. This approach involves representing each document as a vector in a high-dimensional space. In this space, each dimension corresponds to a specific term or word in the document.

The cosine similarity score between two documents is computed by evaluating the cosine of the angle between their corresponding vectors. This score ranges between 0 and 1, with 1 indicating that the documents are identical, 0 indicating that they are orthogonal (i.e., completely dissimilar). By using cosine similarity, it is possible to compare large sets of documents and determine which ones are most similar to each other. This can be useful for various natural language processing tasks, such as information retrieval, text classification, and clustering. The cosine similarity formula is expressed as:

$$\text{cosine similarity}(x, y) = (x \cdot y) / (\|x\| * \|y\|)$$

Where \cdot denotes the dot product of two vectors, and $\|x\|$ and $\|y\|$ denote the norms of the vectors. Intuitively, the cosine similarity is a measure of the cosine of the angle between two vectors, which ranges from 0 (perpendicular to each other) to 1 (indicating identical directions).

When comparing two documents or bits of text, cosine similarity can be used to compare how similar they are by describing each document as a vector of word frequencies or embeddings. The cosine similarity between the two vectors indicates how similar the two texts' contents are expected to be.

VI. IMPLEMENTATION

Text data is a crucial source of information that can be used to gain insights and make informed decisions. However, text data often requires pre-processing in order to prepare it for further analysis and modeling. This pre-processing pipeline typically involves removing punctuations and stop words, tokenizing the text, and building a corpus of similar words. These steps help to reduce the dimensionality of the data and make it more manageable for analysis.

One common technique for representing text data is to convert each word in the corpus into a vector. This allows the text to be processed at a more granular level and enables the identification of patterns and relationships between words. There are several algorithms that can be used to generate word vectors, such as the TF-IDF Vectorizer, Count Vectorizer, and Sent2Vec. The choice of algorithm depends on the specific task and data.

A well-liked metric for determining how similar two vectors are in natural language processing is cosine similarity. It calculates the cosine of the angle formed by the projection of two vectors onto a multidimensional space. Two text documents can be represented as vectors in a high-dimensional space, where each dimension corresponds to a word in the lexicon, when used in the context of NLP. The two vectors' cosine similarity measures how similar the two documents are based on the words they include.

The range of cosine similarity is 0 to 1, with 0 denoting no resemblance and 1 denoting total similarity. When two vectors have a cosine similarity score of 1, it signifies that they are orthogonal to one another and exchange no information, but when they have a score of 0, they are not. Information retrieval, grouping, and text categorization are just a few of the NLP activities that cosine similarity can be utilized for.

In text classification, cosine similarity can be used to classify new documents based on their similarity to existing documents in a corpus. In clustering, cosine similarity can be used to group similar documents together based on their similarity scores. In information retrieval, cosine similarity can be used to retrieve relevant documents based on their similarity to a query document.

In summary, the pre-processing pipeline for text data involves several steps such as removing punctuations and stop words, tokenizing the text, and building a corpus of similar words. Word vectors can then be generated using algorithms such as TF-IDF Vectorizer, Count Vectorizer, and Sent2Vec. Cosine similarity is a popular metric used in NLP to measure the similarity between two vectors, which can be used for tasks such as text classification, clustering, and information retrieval.

Sample text used to the test the outcome by each of the vectorization methods is as given below:

Course outcome statements of Blockchain Lab.:

CO1: Describe the basic concept of Blockchain and Distributed Ledger Technology

CO2: Interpret the knowledge of the Bitcoin Network, nodes, keys, wallets and transactions

Programme Outcome statements

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change.

Fig. 3 and Fig. 4 represent graphs showing similarity percentage when compared each of two course outcomes statements of the subject, Blockchain Lab, with all twelve program outcome statements. Each course outcome statement is compared with twelve program outcome statements to generate twelve similarity values using four different vectorization methods as indicated by different colors.

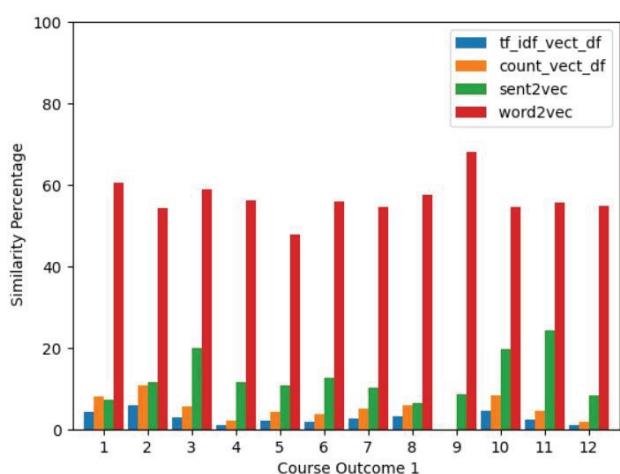


Fig. 3. Similarity percentage by different methods of CO1 with all PO statements

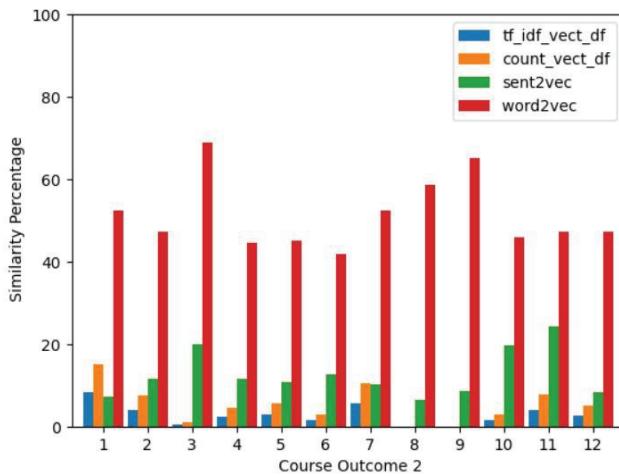


Fig. 4. Similarity percentage by different methods of CO2 with all PO statements

VII. APPLICATION

The work described here involves using various vectorization methods (TF-IDF vectorizer, count vectorizer, Word2Vec, and Sen2Vec) to convert Course Objective sentences and Programme Outcome sentences into vectors. These vectors are then used as input for a text similarity algorithm, specifically cosine similarity, which computes the similarity score between each Course Objective sentence and each Programme Outcome sentence.

The output of this process is a matrix where each row represents a Course Objective sentence, and each column represents a Programme Outcome sentence. The values in the matrix represent the similarity score between each Course Objective sentence and each Programme Outcome sentence.

By examining the values in the matrix, you can determine which Course Objective sentences are most similar to which Programme Outcome sentences. This information can be used to identify potential areas where the course objectives are being met or not being met by the programme outcomes.

Overall, this project aims to use natural language processing techniques to analyse the relationship between Course Objective sentences and Programme Outcome sentences and identify areas where the two may be aligned or misaligned.

VIII. CONCLUSION

In this work the methods are studied that can comprehend the meaning of words and sentences, leading to more nuanced analysis of text similarity. The model developed has the potential to make a significant contribution to fields such as information retrieval, document clustering, and plagiarism detection. It is an innovative solution to the challenge of measuring text similarity, and is confident that it will have practical applications in a range of fields. Four different vectorization methods are used, which give different similarity values. This approach considers both surface-level and underlying semantic relationships between words, providing a more sophisticated analysis of text. The importance of advanced natural language processing techniques for understanding the meaning of text and accurately measuring similarity is highlighted.

REFERENCES

- [1] A. Morales-Garzón, J. Gómez-Romero and M. J. Martín-Bautista, "A Word Embedding-Based Method for Unsupervised Adaptation of Cooking Recipes," in IEEE Access, vol. 9, pp. 27389-27404, 2021, doi: 10.1109/ACCESS.2021.3058559.
- [2] A. Bellino and D. Bascuñán, "Design and Evaluation of Write Better: A Corpus-Based Writing Assistant," in IEEE Access, vol. 8, pp. 70216-70233, 2020, doi: 10.1109/ACCESS.2020.2982639.
- [3] A. Yüksel, B. Ügurlu and A. Koç, "Semantic Change Detection With Gaussian Word Embeddings," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 3349-3361, 2021, doi: 10.1109/TASLP.2021.3120645.
- [4] A. S. Bhardwaj, A. Deep, D. Veeramani and S. Zhou, "A Custom Word Embedding Model for Clustering of Maintenance Records," in IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 816-826, Feb. 2022, doi: 10.1109/TII.2021.3079521.
- [5] U. Farooq, M. S. Mohd Rahim, N. S. Khan, S. Rasheed and A. Abid, "A Crowdsourcing-Based Framework for the Development and Validation of Machine Readable Parallel Corpus for Sign Languages," in IEEE Access, vol. 9, pp. 91788-91806, 2021, doi: 10.1109/ACCESS.2021.3091433.
- [6] L. Van Zijl and A. Combrink, "The South African sign language machine translation project: Issues on non-manual sign generation", Proc. Annu. Res. Conf. South Afr. Inst. Comput. Scientists Inf. Technologists IT Res. Developing Countries, pp. 127-134, 2006.
- [7] T. Hanke, L. König, S. Wagner and S. Matthes, "DGS corpus & dictasign: The hamburg studio setup" in 4th Workshop Represent. Process. Sign Languages: Corpora Sign Lang. Technol. (CSLT), Valletta, Malta, pp. 106-110, 2010.
- [8] C. Rohrdantz, A. Hautli, T. Mayer, M. Butt, D. A. Keim and F. Plank, "Towards tracking semantic change by visual analytics", Proc. 49th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol., pp. 305-310, 2011.
- [9] S. Mitra et al., "An automatic approach to identify word sense changes in text media across timescales", Natural Lang. Eng., vol. 21, no. 5, pp. 773-798, 2015.
- [10] A. Boulton and C. Landure, "Using corpora in language teaching learning and use", Recherche Et Pratiques Pédagogiques En Langues De Spécialité. Cahiers de l'Apliut, vol. 35, no. 2, pp. 1-13, Jun. 2016.
- [11] T.-H. Yen, J.-C. Wu, J. Chang, J. Boisson and J. Chang, "WriteAhead: Mining grammar patterns in corpora for assisted writing", Proc. ACL-IJCNLP Syst. Demonstrations, pp. 139-144, Jul. 2015.
- [12] M. Charles, "Getting the corpus habit: EAP students' long-term use of personal corpora", English for Specific Purposes, vol. 35, pp. 30-40, Jul. 2014.
- [13] Y. Chen, "A statistical machine learning approach to generating graphstructures from food recipes," Ph.D. dissertation, Graduate School ArtsSci., Brandeis Univ., Waltham, MA, USA, 20
- [14] Nguyen, T. B. D., Phung, T. N., & Vu, T. T. (2018). A Rule-Based Method for Text Shortening in Vietnamese Sign Language Translation. In Information Systems Design and Intelligent Applications (pp. 655-662). Springer, Singapore.
- [15] Karbasi, M., Zabidi, A., Yassin, I. M., Waqas, A., & Bhatti, Z. (2017). Malaysian sign language dataset for automatic sign language recognition system. Journal of Fundamental and Applied Sciences, 9(4S), 459-474.