JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A NLP Filter Module for Translation of English Sign Language into Text Sentences

Udaya Raj Dhungana
Assistant Professor
School of Engineering
Pokhara University, Pokhara, Nepal
Email: udaya@pu.edu.np

Abstract: This research work developed a filtration module of the NLP component for "Translation of English Sign Language into Text". This filtration module receives the n-gram gestures from the Image Processing (IP) component, generates all possible combination of each word in each gesture, checks the semantic correctness of each combination of the words from which it can build all possible semantically correct sentences and finally chooses the most appropriate combination. This module uses the information generated from the knowledgebase to check the semantic correctness. The results obtained from experiments shows the average accuracy of 87.6%. This module works for any natural language and therefore it is language independent.

Index Terms **– NLP, Image Processing, English Sign Language, Gesture, Knowledgebase.**

I. INTRODUCTION

This research work has developed a filtration module to integrate in the NLP component of a system called "Translation of English Sign Language into Text". The whole system can be divided into two main components. The first component contains all the processing activities regarding to the image processing and recognition and it is called the Image Processing (IP) component. The second component contains the Natural Language Processing tasks and is called the Natural Language Processing (NLP) component.

The main aim of the whole system is to translate English sign language into English written text. The first component of the system- the IP component receives the video images from the camera, analyzes recorded sign language in the images, chooses the best image among many recorded images, generates hashes from the images and compares them with training images in a database. The IP component gets the three words with the highest probability for a gesture image. Each word generated this way is attached with its probability and the produced gestures are submitted to the NLP module for further processing.

The NLP component receives the probable words attached with a probability for a gesture from the IP module as its input. This module then generates all the combinations of words in n-gram gestures. Some combinations may not constitute a semantically correct sentence while other combination can do. Therefore, this module first chooses a single combination that is most appropriate to constituent a semantically correct sentence. This is performed using the knowledge generated in the knowledgebase. The selected most appropriate combination is given to the Translation module- the next module in the NLP component.

The Translation module then rearranges the words so that they appear in the same order in the semantically correct sentence. Then generated combination of words is transferred to the Generation module. This Generation module prepares the final sentence in English language inserting the required articles and prepositions in the appropriate place in the combination of words. Thus, the whole system receives the sign language recorded from camera, processes the images and finally using NLP tasks translates the sign language into the written text in English language.

II. THE NLP COMPONENT

The NLP component consists of three modules: - a) Filter module, b) Translation module and c) Generation module. The filter module directly receives the output of the IP component as its input. This module forms all possible combinations of words from each gesture and finally it outputs the most appropriate combination of words from the different n-gram gestures. The most appropriate combination of words contains the constituents of a real world sentence. The Figure 1 shows the different constituent modules in the NLP components.

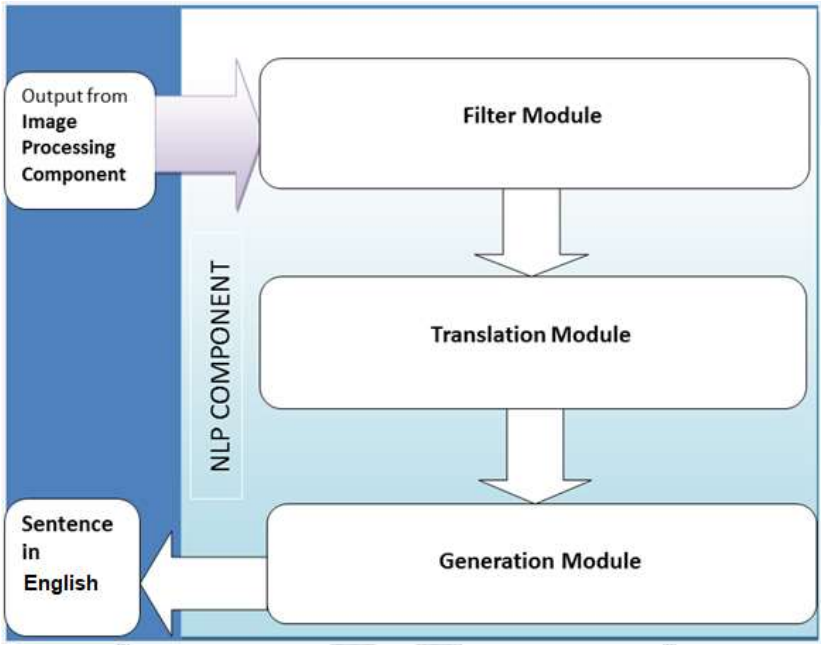

Figure 1. The constituent modules in NLP component

The Translation module takes the output from Filter module and rearranges the words in right position that occur in the English sentence. Finally, the Generation module fills the appropriate article and prepositions where they are necessary to generate the good sentence in the English language.

III. The Filter Module

The Filter module is the first module in the NLP component that directly receives the output of the Image Processing (IP) component. The actual output of the IP component is a continuous flow of words without any indication for the beginning of a new sentence. The big challenge in the NLP module is here to find at which point the new sentence begins in the continuous flow of the words that are coming from the IP component. This task to find the beginning or ending point of a sentence in a continuous stream of words is a complex task in NLP and can be a good research topic in NLP. Therefore, we have made an assumption that there is an indication in the beginning of each sentence.

1. The Output from Image Processing (IP) Component

Each unit in the output of the IP component represents one gesture. The single unit contains always three probable words with attached probability. The probability is attached in the IP component. It is calculated through the Hemming distance between an input hash of an image and a single image hash for a word stored in the database.

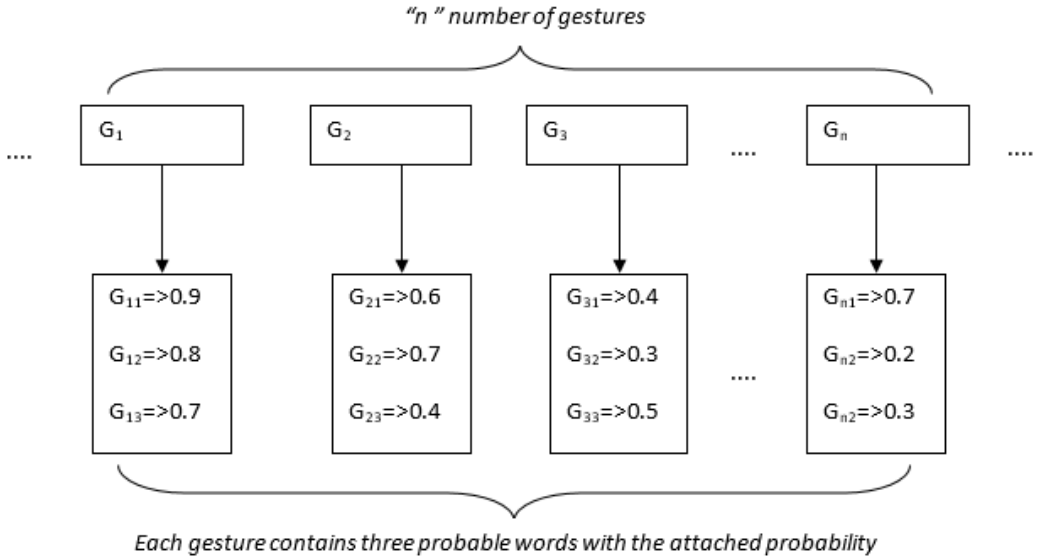

Figure 2. Output stream of n-gram from the IP component

The probability describes how probably the word for the input gesture is. How many such units belong to one sentence is not known since there is no indication in the end or beginning of a sentence. The Figure 2 shows the output stream of n-gram gestures from the IP component. It is actually a continuous stream of gestures that are captured in the IP module. Each gesture contains three probable words with the probability attached showing how much the word is probable to represent that gesture.

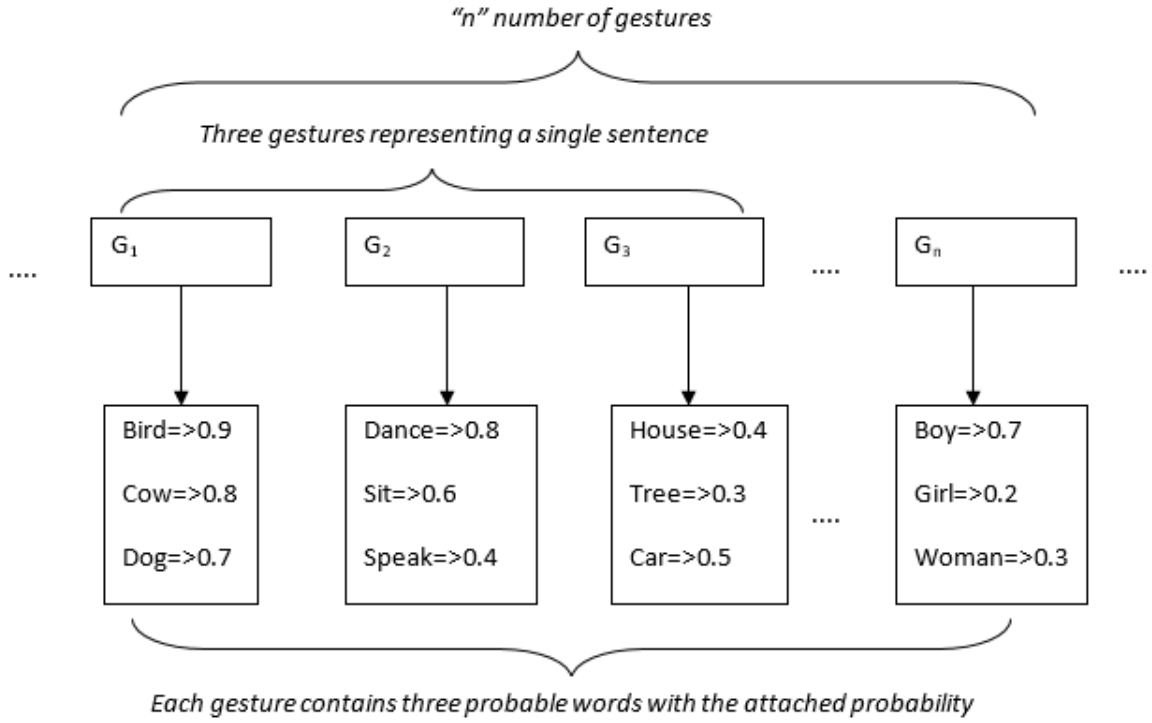

Figure 3. An example for the output of the IP component

There is no any information about which gestures belong to a particular sentence. The IP component captures the video while a person is using the sign language, identifies gestures of the person using the sign language, processes the image containing the gesture and finally attached the three words per gesture with calculated probability how much probably the word represent the gesture. The information about the gesture is then submitted to the NLP module for NLP processing. This is a continuous process while the sign language speaker uses the sign language to provide the information to the audience using the gestures. The Figure 3 shows an example to make clear how the output of the IP component looks like.

As an entirely, the overall task of the NLP component is to receive the n-gram gestures (each gram containing the three words with probability attached for a particular gesture) that represent all gestures within a sentence, build the possible combinations of words using all three words in each gesture from the n-gram gestures, use the NLP theories to choose the most appropriate combination of words from the gestures and finally present the sign language translation in the most accurate sentence in English language.

The Filter module is responsible to receive the input from the IP component, generate all the possible combinations of words from each gesture that constitutes a word in a sentence and finally remove all the others combinations selecting the most appropriate one.

IV. The Related Works

Khan et al. in [1] developed an efficient sign language translator device using convolutional neural network and customized ROI segmentation in 2019 for conversion of Bangla sign language to text. They trained 5 sign gestures using custom image dataset to implement in Raspberry Pi. They noticed better outcomes while using ROI selection approach with compared to conventional approaches. Hernandez-Rebollar et al. in [2] used a new instrumented approach to translate American Sign Language into sound and text. They broke down gestures of American Sign Language into poses and movements which are recognized by software modules. For 42 postures, orientations, 11 locations and 7 movements, they found recognition rates of modules up to 100% using linear classification. They used American Sign Language Dictionary with 30 signs to test sign recognizer and found 98% accuracy.

In 2001, Viola and Jones developed a translator algorithm to detect human face in real time. Later many researchers used this algorithm to detect other objects like car's number place, eyes, and mouth and traffic signs. In addition, it is used to detect hand signs successfully. Truong et al. in 2016 [3] proposed a system which detect static hand signs of alphabets in American Sign Language by adopting AdaBoost and Haar-like classifiers. In 2015, Elmahgiubi et al. developed a Data Acquisition and Control system [4]. This sys that translates the sign language into text. It is capable of capturing the gestures of the hands and convert them into readable text. It can understand 20 out of 26 letters with a recognition accuracy of 96%.

Duarte in 2019 explored sign language translation from spoken language to sign language and vice versa [5]. They developed How2Sign which is a public American Sign Language dataset. This data set can be used to advance sign language translation. Kunjumon and Megalingam in 2019 developed a hand gesture recognition system to translate Indian sign language into speech and text in two languages English and Malayalam to display in android phone [6].

Lozynska et al. in 2020 developed a Tourist Sign Translation System for Ukrainian Sign Language using rule-based method [7]. It consists of Offline Phrasebook and Individual Translator for both online and offline use. Jiang and Zhu in 2019 proposed a new Chinese sign language identification approach. The approach adopted wavelet entropy for feature reduction and classification was employed using support vector machine with overall accuracy of 85.69 ± 0.59%.

Almasoud and Al-Khalifa in 2011 proposed a semantic machine translation system for translating Arabic text to Arabic sign language in the jurisprudence of prayer domain by applying ArSL translation rules as well as using domain ontology [9]. Similarly,

Luqman and Mahmoud in 2019 proposed a rule-based machine translation system to translate Arabic text into ArSL [10]. They evaluated their system using a parallel corpus in the health domain consisting 600 sentences and found the accuracy of 80%.

V. THE PROBLEM STATEMENT

The first module of the NLP component is the Filter module. This module receives the output from the IP component that forms the input to the NLP component. The input is received in the form as shown in the Figure 2 and Figure 3 which is already described in the previous section. After receiving the input from the IP component until delivering the input to the Translation module, this module has to take care of the following responsibilities:

a) Receive the output from the IP component
b) Separate the words and probabilities for each gesture
c) Keep track of these words and probabilities attached for the future use
d) Build all combination of words from n-gram gestures that can be the constituents of a single sentence
e) Filter the most appropriate combination that may form a sentence in a real world
f) Output the final result to the next component called the Translation module for the further NLP processing.
g)

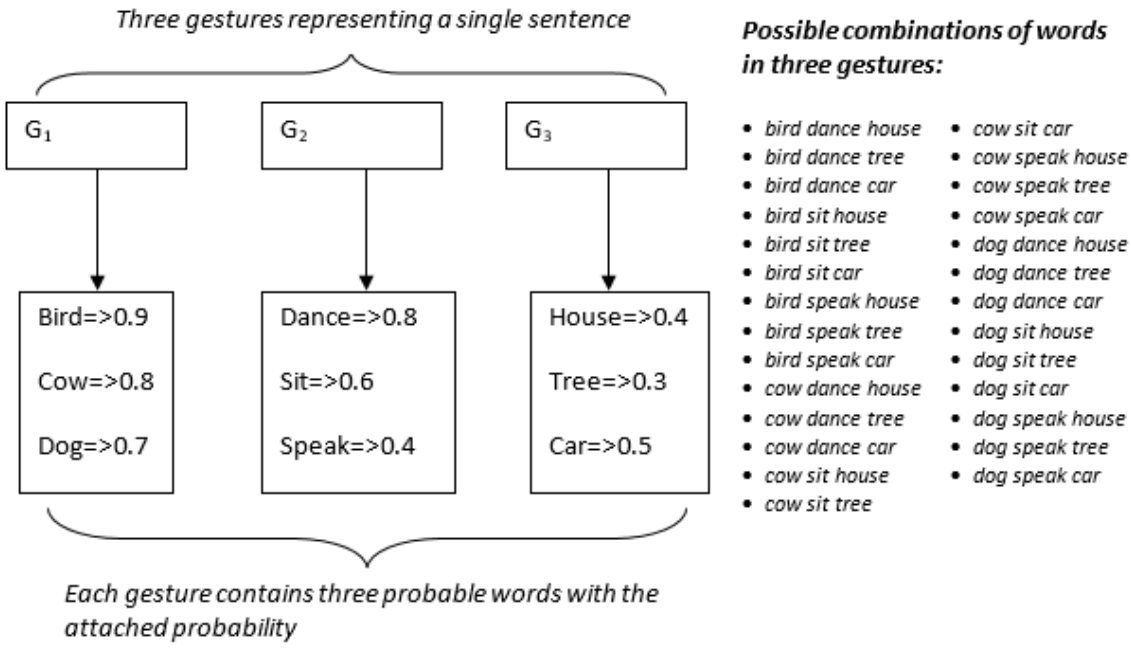

Figure 4. Three gestures and possible combinations

There are the major tasks that need to be performed by this module. The first task is to generate all the possible combinations that can be formed using the three words in each gesture with n-gestures that constituent a single sentence in the sign language. The remaining two tasks are related to the filtration of the generated combination. For example, let us consider three gestures G1, G2 and G3 constituent a sentence parts. Each gesture has three probable words each attached with a probability value. Fig. 4 shows these three gestures and the possible combinations of each word in each gesture. The first filtration is performed just to remove the combination of words that do not contain appropriate sentence constituents.

The first filtration process produces few numbers of combinations which can be used to produce appropriate sentence. The second filtration uses the probability and selects only one most appropriate combination among the many combinations produced by the first filtration process. Finally, a single combination is provided to the next module called the Translation module for the further processing.

VI. THE SOLUTION APPROACH

The solution approach uses the concept of knowledgebase to check semantically correct sentence. The following subsections explains the solution approach adapted in this research:

1. Concept of Knowledgebase

The main task of this module is the filtration of one appropriate combination from the all generated combinations as shown in Fig. 4. The main problem here is to determine how to identify a generated combination of words constituent a correct sentence with a meaning. To solve this problem, we developed the concept of a knowledge base to check whether the combination of the words may lead to produce a correct sentence or not.

Suppose a combination contains words “bird”, “sing” and “tree”. Now the words in the combination are taken both way individually and together and searched in the Wikipedia. The text returned by the Wikipedia in both cases is taken as the knowledge to check semantically the correctness of the sentence. Each sentence in the knowledge base is separated and forms an array of sentences. One element of that array is a sentence. Then all the words in the combination are checked to find whether all these words can be found together in a single sentence or not. If all the words in a combination are occurred in a single sentence, we believe that the combination containing those words will produce a correct sentence semantically.

We measure this factor to see how much a combination is semantically correct to form a sentence using probability. Suppose a combination contains five words and suppose if all the five words appear in a single sentence of the generate knowledge base, then we attached the probability 1.0 for that combination. If only three words out of five appear in a sentence, then we attached a probability 0.6 to this combination. In this way we calculate the probability for a combination to determine the correctness of a sentence using a generated knowledge base. The attached probability defines how probably the generated combination may produce a semantically correct sentence.

2. Threshold Probability Value

When we calculate and attach a probability for a combination, then the another important and tricky task is to find out the threshold probability value that will check the combinations to pass the filtration. In this point we have assumed that a combination with probability 0.6 and above will form a semantically correct sentence and therefore combinations with probability 0.6 and above are selected and passed to the next stage filtration.

3. Second Stage Filtration

The filtration is performed in two stages. In the first stage filtration, we used the knowledgebase to determine how probably a generated combination can form a semantically correct sentence and using the defined threshold probability value, the combinations with probability 0.6 and above are passed to the second stage filtration.

In the second stage filtration, we make the use of the attached probability for each word in each gesture that is provided by the IP component. We safely keep track of each provided probability from the IP component. For each combination that passes the first stage filtration, we calculate the summation of the probability of each word in the combination and the sum is attached to each combination.

Finally, the combination with the highest summation value is selected as a winner combination and this winner combination is provided to the next module the Translation module in the NLP component for the further processing.

VII. THE ARCHITECTURE OF FILTER MODULE

The solution generated in the previous section is implemented using java. The various components that implement the solution are shown in the Fig. 5. This forms the overall architecture of the Filter module. The output from the IP module is received by the input receiver. Then the input is provided to the Word and Probability Splitter block which splits the words and its attached probability. The Fig. 6 shows the actual input format that our module receives.

The Fig. 6 shows the 4-gram input. The words in each gram (one gesture) are separated by a space. Each unit in a gesture contains a words and its attached probability separated by colon. Each gesture is placed in a new line. If n-gram input contains six gestures, then there will be six lines each containing a single gesture. The input could be any n-gram gesture.

The detached probabilities from the words are sent to the Probability Tracker block which keeps the track of each probability to use in the Second Stage Filtration while the detached words from each gesture are sent to the Possible Combination Generator module to generate all the possible combinations of words from each gesture. All the generated combinations are then submitted to the First Phase Filter.

As described in the solution section, the First Phase Filter uses the information from the knowledgebase, calculate a probability for each combination using Probability Generator block to show how probable the combination is to constitute a semantically correct sentence and finally all the combinations with the probability threshold 0.6 or above are let to pass the first stage filtration. All the remaining combinations are blocked thinking that these combinations with probability less that 0.6 could not constitute a semantically correct sentence in real world.

The passed combinations from the first stage filtration are feed into the Second Phase Filter for second stage filtrations. The Second Phase Filter uses the probability from the Probability Tracker block for each word in the received combinations and for each combination it finds the summation of probabilities by adding probabilities of each word in the combination. Finally, the combination which has the highest summation value win the race as a most probable semantically correct combination to produce a sentence and it is the final output of our Filter module. This winner and single combination is sent to the next module called the Translation module.

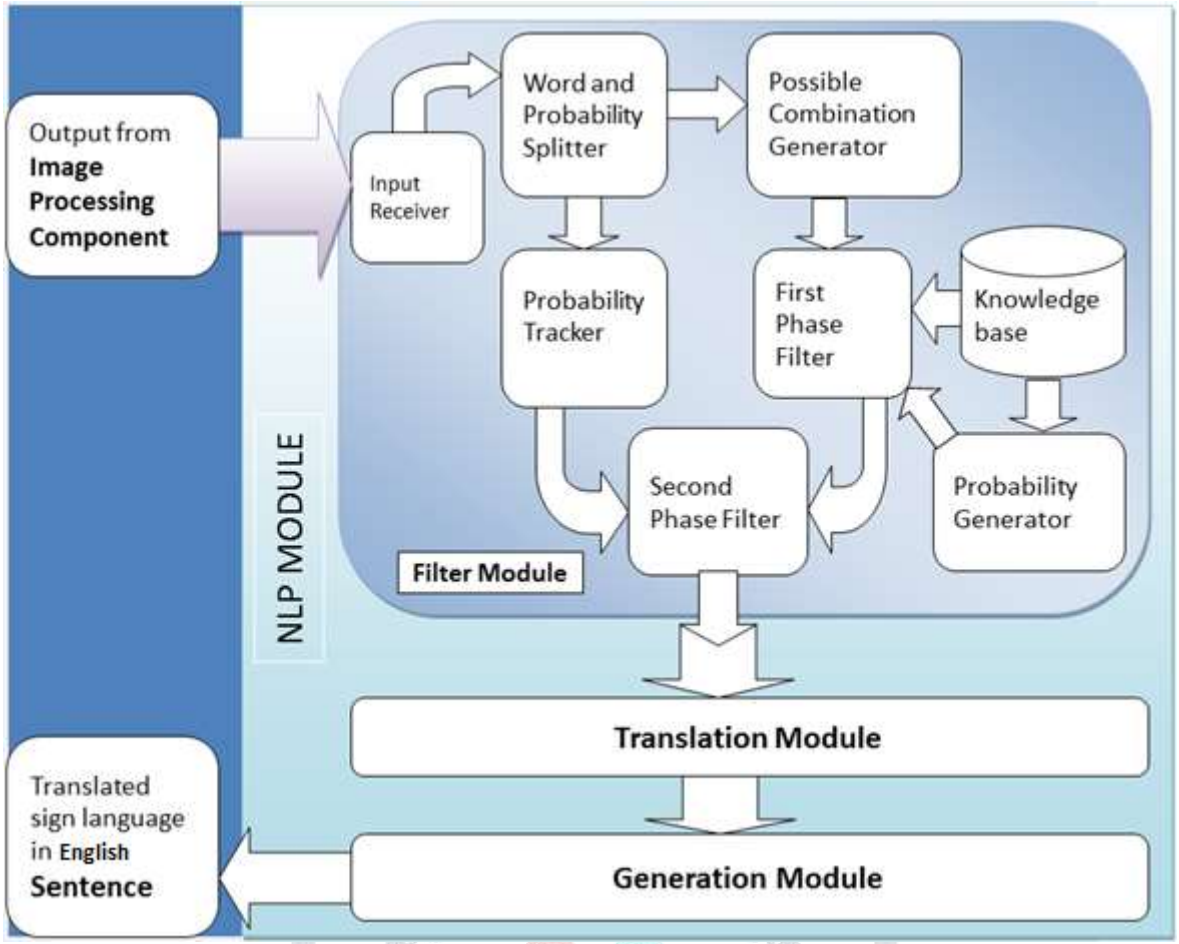

Figure 5. Architecture of Filter Module

```
Airplane: 0.43 bird: 0.54 car: 0.53
   Sing: 0.65 sit: 0.78 run: 0.37
 Forest: 0.77 city: 0.58 tree: 0.44
Yesterday: 0.65 two: 0.45 many: 0.33
```

Figure 6. The Input format that containing 4-gram input

VIII. THE KNOWLEDGEBASE

The knowledgebase is the most important part of this module since it is used to check the semantic correctness of each generated combination. We strongly believe that if all the words in a combination are found with in a same real world sentence stored in the knowledgebase, the combination of the words can constitute a semantically correct sentence. To generate the knowledge in the knowledgebase is quite tricky. The concept here is that we first look for the words in the combination, pick the each word and search it in the Wikipedia, the information obtained for the word is retrieved and store in the knowledgebase. This process of finding the information of word and retrieving the information related to the word is repeated of all words in each combination. Plus, we also find the information for each combination by taking the all words in a combination as a sentence constituent also. At this time the information retrieved may contain some sentence which may contain all the words in a combination. The actual concept is this. However, in this project we have not implemented this concept in the same way as it is stated. We have built the knowledgebase manually so that it contains only the information about the words which are provided for test the system.

IX. THE EXPERIMENTS AND RESULT ANALYSIS

An experiment is designed to take 3-gram inputs which are actually the output of the Image Processing Part. Each gram can contain any number of possible estimated word with a probability count. Using these inputs, all possible combinations are generated and from these only the combinations which can give correct sentential forms are filtered using the information provided in knowledgebase. Finally, a single most probable combination is selected for output of this Filter module and sent to the Translation Module for the further processing.

$$Accuracy = \frac{Number\ of\ correct\ combinations\ of\ words}{50} X\ 100\% \qquad \textbf{Eq. (1)}$$

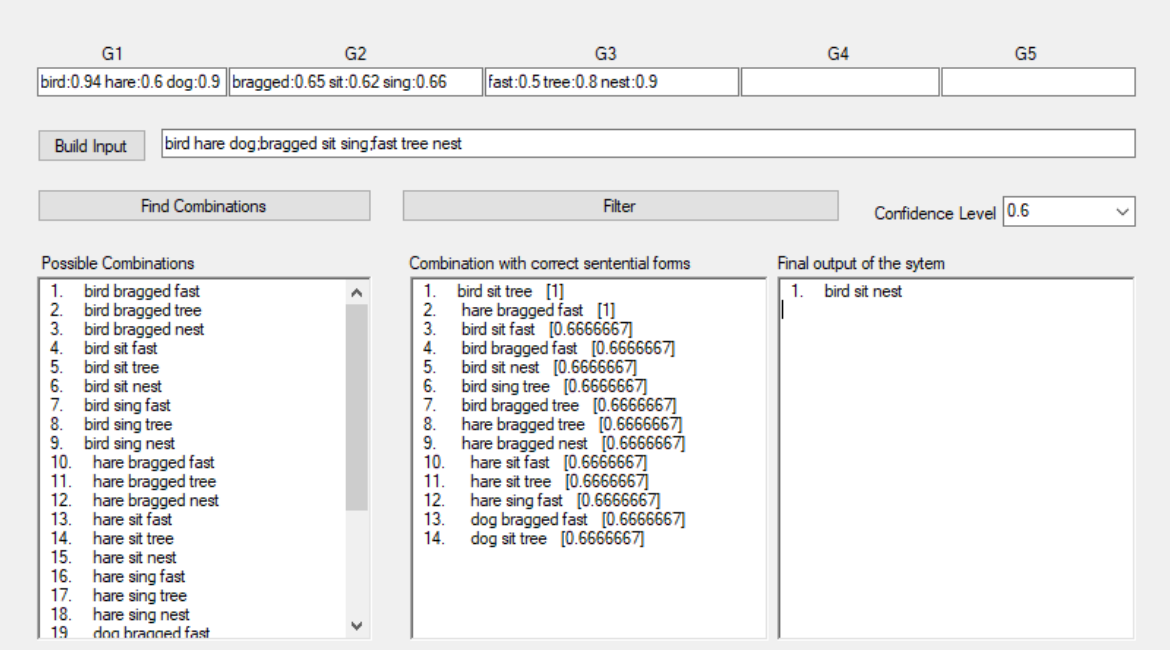

Figure 7. The output for 3-gram input

The developed system is tested with 10 different cases in the knowledgebase. In each case, 50 different inputs in each n-gram with randomly generated probabilities are provided. The experiment is repeated for 10 different cases, each case containing 50 different input for 3-grams. Altogether, 500 runs of experiments are executed. The Figure 7 shows an instance of a run of experiments. It shows the interface for the experiments. The results obtained from the system are checked to find whether the output of the system contains the n-gram words which can be used to generate semantically correct sentence. For each case, accuracy is calculated using Eq (1).

The Table 1: Accuracies obtained in experiments

Case Number	No of Semantically Correct Output	Observed Accuracy
1	45	90
2	49	98
3	38	76
4	41	82
5	44	88
6	42	84
7	50	100
8	39	78
9	48	96
10	42	84

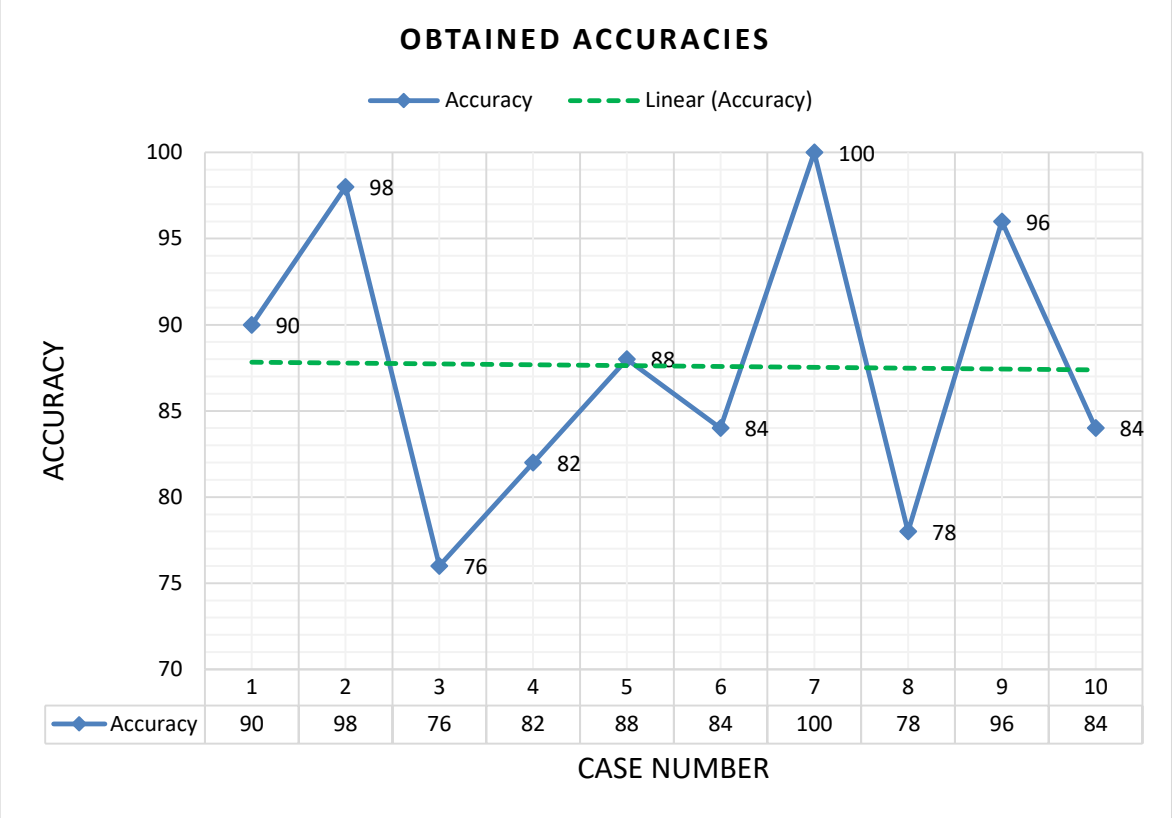

Figure 8. Accuracies obtained in 10 different cases, each case containing 50 different inputs in each n-gram with randomly generated probabilities

The Table 1 shows the results of experiments with 10 cases, each containing 50 different inputs. The results of the experiments shows accuracy of 100% in case number 7. In case number 3, the lowest accuracy (76%) is observed. The average accuracy is found to be 87.6%. The Figure 8 shows the accuracies obtained in 10 different cases, using line chart diagram.

X. Conclusions

We have successfully developed and implemented a Filter module in the NLP component. This module is capable to generate the all possible combinations of words in n-gram gestures, each gesture containing exactly three words, to check a combination to determine whether the combination may produce a semantically correct sentence or not and finally choose a most appropriate combination which is most appropriate to constitute a semantically correct sentence using the knowledge from the knowledgebase. Our developed module is language independent. This module can be used to check the semantically correct sentence in any natural language with no or little modification. The only change need to do is that the knowledgebase must contain the knowledge in the particular natural language in which this module is being used. Mainly the accuracy of our module depends on how correctly the probability is calculated in the IP component and sent to our NLP module plus how exactly the information in the knowledgebase is retrieved to build the knowledge. In this project, we have generated the knowledgebase manually. Therefore there is no ways to check the accuracy of the knowledgebase. Therefore the accuracy of our module entirely depends on how accurately the probability for each word is calculated in the previous IP component.

XI. Limitations and Recommendations

The main limitation of our NLP module is at the point of receiving the input from the IP component. The IP component provides the finite gestures continuously as the sign language speaker goes on using the sign language to express the information. The sign language speaker don't use any sign to indicate the end of a sentence termination just like we people speak our language. We go and go speaking the information but we do not speak the termination of the spoken sentence as it is done only in the written language. Therefore, we don't have any indication in the sentence termination in the input we are provided with. Also we did not have developed a module that can take the continuous gesture stream and can separate gestures that belongs to a single sentence. Therefore, we have assumed that we get the input that contains only the gestures that belong to a sentence only. This limitation to find out the termination of a sentence in spoken language is our future research task. Another limitation is that we have generated the knowledge in the knowledgebase manually. In future, we will develop a module that can automatically retrieve the required information from the Wikipedia as stated in the concept of our knowledgebase.

References

[1] S. A. Khan, A. D. Joy, S. M. Asaduzzaman and M. Hossain, "An Efficient Sign Language Translator Device Using Convolutional Neural Network and Customized ROI Segmentation," *2019 2nd International Conference on Communication Engineering and Technology (ICCET)*, 2019, pp. 152-156, doi: 10.1109/ICCET.2019.8726895.

[2] J. L. Hernandez-Rebollar, N. Kyriakopoulos and R. W. Lindeman, "A new instrumented approach for translating American Sign Language into sound and text," *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, 2004, pp. 547-552, doi: 10.1109/AFGR.2004.1301590.

[3] V. N. T. Truong, C. Yang and Q. Tran, "A translator for American sign language to text and speech," *2016 IEEE 5th Global Conference on Consumer Electronics*, 2016, pp. 1-2, doi: 10.1109/GCCE.2016.7800427.

[4] M. Elmahgiubi, M. Ennajar, N. Drawil and M. S. Elbuni, "Sign language translator and gesture recognition," *2015 Global Summit on Computer & Information Technology (GSCIT)*, 2015, pp. 1-6, doi: 10.1109/GSCIT.2015.7353332.

[5] Amanda Cardoso Duarte. 2019. Cross-modal Neural Sign Language Translation. In Proceedings of the 27th ACM International Conference on Multimedia (MM '19). Association for Computing Machinery, New York, NY, USA, 1650–1654. DOI:https://doi.org/10.1145/3343031.3352587

[6] J. Kunjumon and R. K. Megalingam, "Hand Gesture Recognition System For Translating Indian Sign Language Into Text And Speech," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2019, pp. 14-18, doi: 10.1109/ICSSIT46314.2019.8987762.

[7] Lozynska O., Savchuk V., Pasichnyk V. (2020) Individual Sign Translator Component of Tourist Information System. In: Shakhovska N., Medykovskyy M.O. (eds) Advances in Intelligent Systems and Computing IV. CSIT 2019. Advances in Intelligent Systems and Computing, vol 1080. Springer, Cham. https://doi.org/10.1007/978-3-030-33695-0_40

[8] Jiang X., Zhu Z. (2019) Chinese Sign Language Identification via Wavelet Entropy and Support Vector Machine. In: Li J., Wang S., Qin S., Li X., Wang S. (eds) Advanced Data Mining and Applications. ADMA 2019. Lecture Notes in Computer Science, vol 11888. Springer, Cham. https://doi.org/10.1007/978-3-030-35231-8_53

[9] Ameera M. Almasoud and Hend S. Al-Khalifa. 2011. A proposed semantic machine translation system for translating Arabic text to Arabic sign language. In Proceedings of the Second Kuwait Conference on e-Services and e-Systems (KCESS '11). Association for Computing Machinery, New York, NY, USA, Article 23, 1–6. DOI:https://doi.org/10.1145/2107556.2107579

[10] Luqman, H., Mahmoud, S.A. Automatic translation of Arabic text-to-Arabic sign language. Univ Access Inf Soc 18, 939–951 (2019). https://doi.org/10.1007/s10209-018-0622-8

BIOGRAPHIES OF AUTHOR

Dr. Udaya Raj Dhungana, grew in a beautiful city Pokhara, Nepal, is the inventor of PolyWordNet- a lexical database that organizes the senses of polysemy words based on their related words. He achieved Doctor of Philosophy (PhD) in Computer Engineering from Institute of Engineering, Tribhuvan University, Nepal under the Young PhD Fellowship granted by University Grant Commission, Nepal in 2021. He also revceived Erasmus Mundus Action 2 Scholarship under IDEAS project for his PhD research at Darmstadt University of Applied Sciences, Germany from Sept, 2014 to Jun, 2015. During his research stay at Darmstadt, one of his research paper is awarded as a best research paper in IEEE conference CICSyN 2015, Riga, Latvia. He obtained Master of Engineering (ME) in Computer Engineering from Kathmandu University, Nepal in 2011 and Bachelor of Computer Engineering from Pokhara University, Nepal in 2005.

He is an Assistant Professor at School of Engineering, Pokhara University as since 2013. He served as an ICT director at Pokhara University from 2017 to 2019. He is also working at the Darmstadt University of Applied Sciences as a guest faculty since 2018. In addition, he is the coordinator of Erasmus+ scholarship project between Darmstadt University of Applied Sciences and Pokhara University since 2019. His research interest includes the Word Sense Disambiguation, Lexical Database, Knowledge Representation, Expert System and Automatic Question Answering.