Pattern Recognition System for Translating the English Sentence into Hindi

M.M.S. Rauthan
Department of Computer Science
H.N.B. Garhwal University
Srinagar Garhwal, India

Pritam Singh Negi
Department of Computer Science
H.N.B. Garhwal University
Srinagar Garhwal, India

H.S. Dhami
Director, Information Communication Technology
Kumaun University, Nainital

ABSTRACT

In the present work a model is proposed which deals with specifying the pattern for translating the English sentences into Hindi. Here a Vector Space based translation model has been proposed that transforms a Vector Space by graphical representation of text that addresses the issues of manual, automatic and adaptive strategies by incorporating the selection preferences for word argument positions. Vector Space Model (VSM) represents documents and queries usually as Vectors, Matrices or Tuples. The similarity of the Query Vector and Document Vector is represented as a scalar value. This model constructs a sentence graph for a given sentence and applies structural parsing on this sentence. The quality of a system is measured by considering its usefulness for typical users of the system. The recent development of related techniques stimulates new modeling and estimation methods that are beyond the scope of the traditional approaches.

Keywords: Pattern Recognition, Vector Space Model, Mathematical Model.

1. INTRODUCTION

The goal of proposed model is used to construct a translation model in such a way that the phonetic structure of words should be preserved as closely as possible. It presents transformation of a vector space by graphical representation of text. Graph Theory is applied here so that the dimensions of the vector space tally with the number of Parts of Speech and take linguistic sense in the right manner.

The Translation Model proposed in this system works on Vector Space. In this model properties of a corpus are defined in the terms of vector space. The eye-catching concept in this paper is the study of Natural Language with the help of Knowledge Graph Theory. Sub-graphs can be represented by a vertex as well. Here edges and arcs are seen as single elements, not as pair of vertices. Graph theory has been tried to tackle the cumbersome process of n-dimensional vector space so that the dimensions of vector space tally with the number of Parts of Speech and take linguistic sense in the right manner. The proposed translation model overcomes the limitations of other translation models such as SMT, VSM etc. by representing a sentence graphically then forming a vector Space for it. The developed technique has the advantage of overcoming the harassing position in machine translation when a sentence is translated word by word without providing any emphasis on Language Semantics. The proposed Translation Model resolved various translation problems that were prevalent in previous systems. To name a few:

- Abundant Homophony
- No counterparts of several Parts of Speech in another language
- Erroneous translation of Rigid Word Combinations.
-

2. LOGIC

For specifying the pattern of translating the sentence we can categorize the whole program into three phases. In first phase we perform the following steps:

- For a given sentence in source language L1, a sentence graph is to be mapped on a knowledge graph.
- It comes under Structural Parsing and two graphs for each word can be constructed.
- Syntactic Word Graph refers the ways the words can be combined.
- Semantic Word Graph refers to the actual meaning of the word.

In second phase:

- We map this knowledge graph on a knowledge graph in another source language L2 with the same meaning.
- Word by Word substitution is applied maintaining the original structure and thus we get the required sentence graph.

In third phase we can utter this graph by the rules of language L2 for conversion in the sentence form.

Sentence in L1 $\xrightarrow{\textbf{Structural parsing}}$ Sentence Graph in L1.

Sentence Graph in L1 $\xrightarrow{\textbf{Transformation}}$ Sentence graph in L2.

Sentence graph in L2 $\xrightarrow{\textbf{Uttering}}$ Sentence in L2.

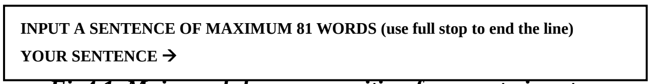
3. ALGORITHM FOR PATTERN RECOGNITION

This algorithm is created for calculating the pattern of the sentence when translating the English sentence into Hindi by comprising the pattern of similar type of group of sentences.

```
1. Start
2. Input a sentence of maximum 81 words (use full stop to end the line)
3. Store the sentence in to the memory.
4. Break the sentence into its constituent words.
5. Call SUBLINK module and pass each constituent word into it.
    5.1. Search for collocation by searching all the possible combinations of the words.
    5.2. If collocation found then
        5.2.1.  Remove collocation
        5.2.2. Return
    5.3 else
```

```
            5.3.1. Check consecutive verbs
            5.3.2. if found
                  5.3.2.1. Substitute a single term instead of a multiple
                           terms
                  5.3.2.2. Return
            5.3.3 Return
        5.4 Return
 6. Call TERM SEARCH Module
   6.1. Check for presence of each term in the database.
   6.2. If term found
        6.2.1  Call POS tagging module
        6.2.2 Store the term and its related part of speech
        6.2.3 Return
    6.3 Else
        6.3.1 Search the term against a verb database
        6.3.2 If match found
             6.3.2.1 Tag the term as verb and store the result
             6.3.2.2. Return
         6.3.3 Else
             6.3.3.1 Tag the term as Noun and store the result
             6.3.3.2 Return
         6.4 Call L1 GRAPH module to display the sentence
              graph.
          6.5  Return
  7. Call Transformation Module
     7.1 Calculate vector space and display it.
         7.2 Transformation sequence in L2 is available
             7.2.1 Display vector space for transformed sentence in
                   L2.
             7.2.2 Display proposed word sequence in L2.
             7.2.3 Return.
          7.3 Else
              7.3.1 Display Transformation sequence not available.
              7.3.2 Ask user to manually enter the transformation
                    sequence.
              7.3.3. Store the sequence in text file.
              7.3.4 Return
          7.4 Return
  8. End
```

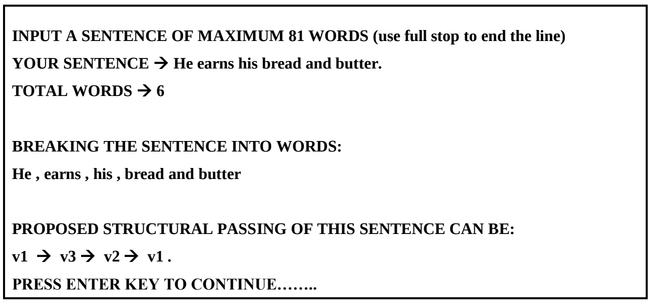
4. SNAPSHOTS AND DESCRIPTION OF DIFFERENT MODULE PRESENT IN THE PROPOSED MODEL



```
INPUT A SENTENCE OF MAXIMUM 81 WORDS (use full stop to end the line)
YOUR SENTENCE →
```

Fig4.1 Main module screen waiting for user to input a English sentence

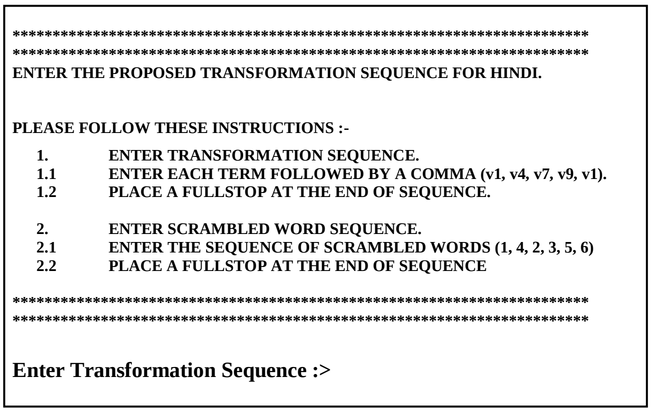
In this user input a sentence in proper way. This computes the number of words (separated by spaces) in the given sentence and calls other sub-modules subsequently. It passes the entire sentence to SUB-LINK sub-module which latter checks for collocation and removes if it is present.



```
INPUT A SENTENCE OF MAXIMUM 81 WORDS (use full stop to end the line)
YOUR SENTENCE → He earns his bread and butter.
TOTAL WORDS → 6

BREAKING THE SENTENCE INTO WORDS:
He , earns , his , bread and butter

PROPOSED STRUCTURAL PASSING OF THIS SENTENCE CAN BE:
v1 → v3 → v2 → v1 .
PRESS ENTER KEY TO CONTINUE……..
```

Fig 4.2 Detection of collocation and structural parsing of sentence

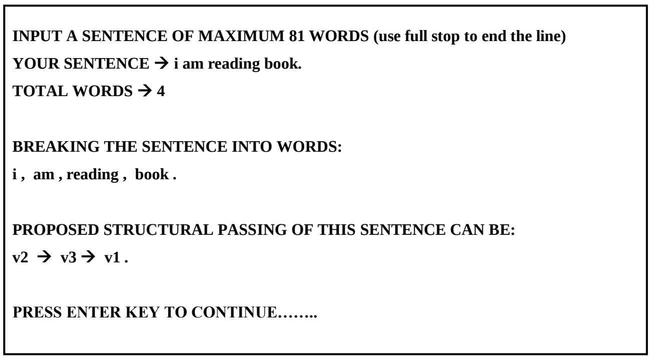
It is called by the main module after a user has entered a sentence ending with a full stop. This module accepts the entire sentence as argument and breaks it into consecutive terms. Then it makes a combination of each term with every term to check for possible combination of words by matching these combinations against a database a collocation database. If any phrases or word combinations are found then they are adjusted by concatenating them together. This module also concatenates multiple verbs as they get transformed into a single verb in L2 language also.



```
*************************************************************************
*************************************************************************
ENTER THE PROPOSED TRANSFORMATION SEQUENCE FOR HINDI.


PLEASE FOLLOW THESE INSTRUCTIONS :-
   1.        ENTER TRANSFORMATION SEQUENCE.
   1.1       ENTER EACH TERM FOLLOWED BY A COMMA (v1, v4, v7, v9, v1).
   1.2       PLACE A FULLSTOP AT THE END OF SEQUENCE.

   2.        ENTER SCRAMBLED WORD SEQUENCE.
   2.1       ENTER THE SEQUENCE OF SCRAMBLED WORDS (1, 4, 2, 3, 5, 6)
   2.2       PLACE A FULLSTOP AT THE END OF SEQUENCE

*************************************************************************
*************************************************************************


Enter Transformation Sequence :>
```

Fig 4.3 Waiting for user response to enter the new sequence if there is no predefined transformation

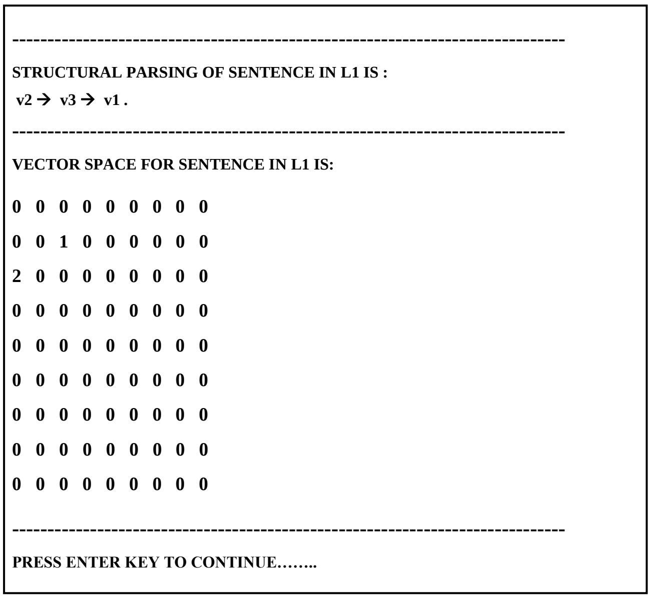
This module searches each term for its presence in the text database. This module is called by SUB-LINK module after collocation and verb concatenation is properly done. It searches whether each term entered is available in the main database or not. If a match is found then its related part of speech tagging is done by calling POS TAGGING module. If no match is found then the term is searched against a Verb database. If a match is found then the term is assigned as Verb else it is assigned to be a Noun. This module is called by TERM SEARCH module only if the term is found either in main database or in verb database. This module tries to allocate a suitable Part Of Speech to each term and stores the result.



```
INPUT A SENTENCE OF MAXIMUM 81 WORDS (use full stop to end the line)
YOUR SENTENCE → i am reading book.
TOTAL WORDS → 4

BREAKING THE SENTENCE INTO WORDS:
i ,  am , reading ,  book .

PROPOSED STRUCTURAL PASSING OF THIS SENTENCE CAN BE:
v2  →  v3 →  v1 .

PRESS ENTER KEY TO CONTINUE……..
```

Fig 4.3 No collocation but verb concatenation present and its structural parsing after verb concatenation

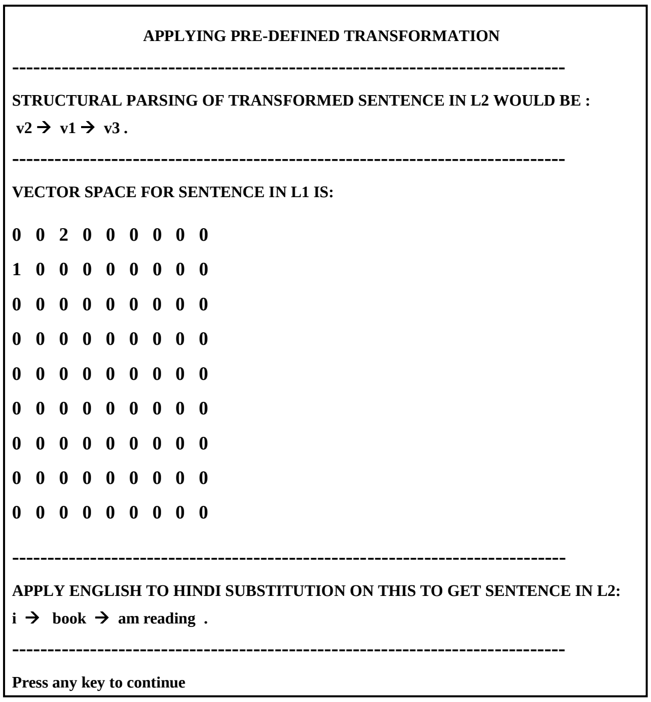
This module is called by POS TAGGING module after it has assigned all terms a suitable part of speech. This module presents a graphical representation of the sentence that the user has entered previously. This graphical representation shows each term with its related part of speech tagging that has been assigned to each term by the program.



```
-------------------------------------------------------------------------------
STRUCTURAL PARSING OF SENTENCE IN L1 IS :
 v2 → v3 → v1 .
-------------------------------------------------------------------------------
VECTOR SPACE FOR SENTENCE IN L1 IS:

0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0
2  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0

-------------------------------------------------------------------------------
PRESS ENTER KEY TO CONTINUE……..
```

Fig 4.4 Structural parsing and vector space representation of a sentence if its transformation sequence is available

This module checks whether a transformation sequence for the entered sentence is available or not. It is called by Main Module and if a sequence is found then the corresponding Sentence Graph and its Vector Space in L1 language as well as its Transformed Sentence graph and vector space is also displayed on user's screen.



```
                    APPLYING PRE-DEFINED TRANSFORMATION
-------------------------------------------------------------------------------
STRUCTURAL PARSING OF TRANSFORMED SENTENCE IN L2 WOULD BE :
 v2 → v1 → v3 .
-------------------------------------------------------------------------------
VECTOR SPACE FOR SENTENCE IN L1 IS:

0  0  2  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0

-------------------------------------------------------------------------------
APPLY ENGLISH TO HINDI SUBSTITUTION ON THIS TO GET SENTENCE IN L2:
i →  book →  am reading  .
-------------------------------------------------------------------------------
Press any key to continue
```

Fig 4.5 Applying predefined transformations to a sentence in L1

It displays the possible word sequence on which the user should apply term to term substitution to get the desired transformed sentence in L2 language.

5. FLOW CHART OF THE MODEL

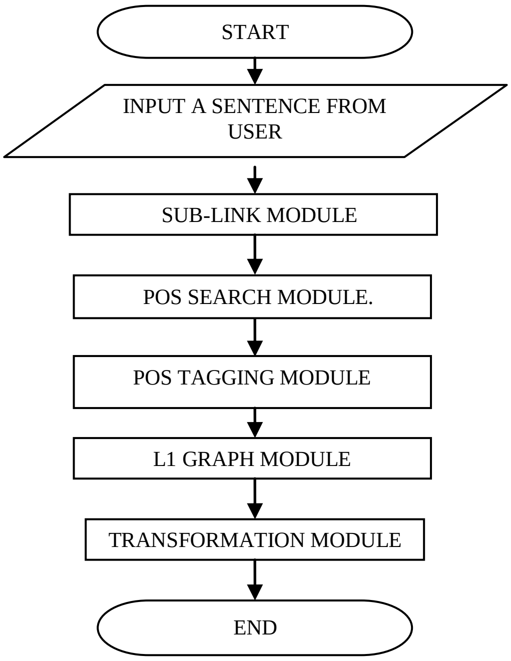


Fig. 1 Flow Chart of main module

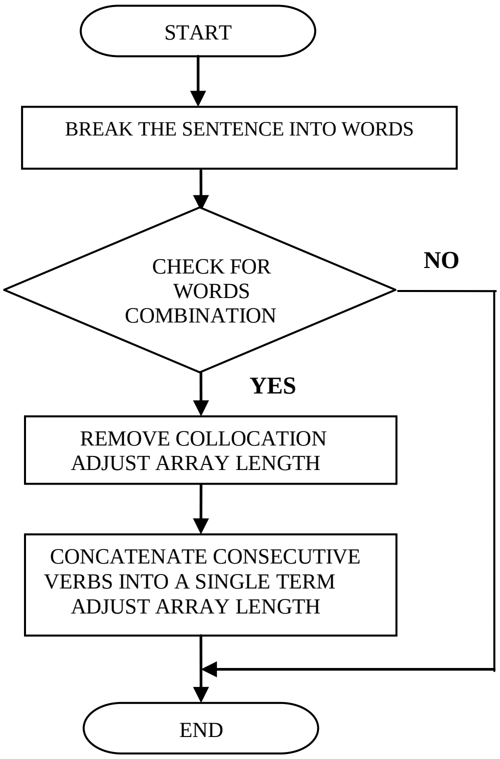

START
BREAK THE SENTENCE INTO WORDS
CHECK FOR WORDS COMBINATION
NO
YES
REMOVE COLLOCATION ADJUST ARRAY LENGTH
CONCATENATE CONSECUTIVE VERBS INTO A SINGLE TERM ADJUST ARRAY LENGTH
END

Fig. 2 Flow Chart of sub link module

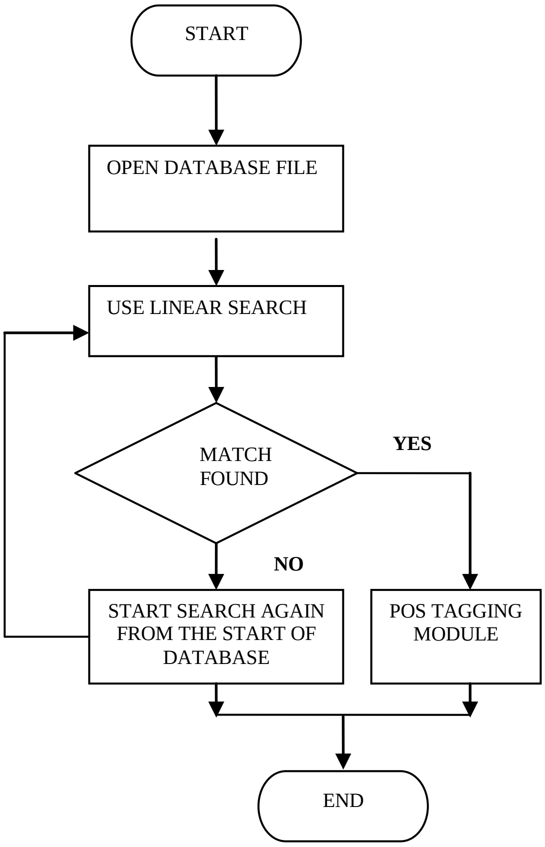


Fig. 3 Flow Chart of term search module

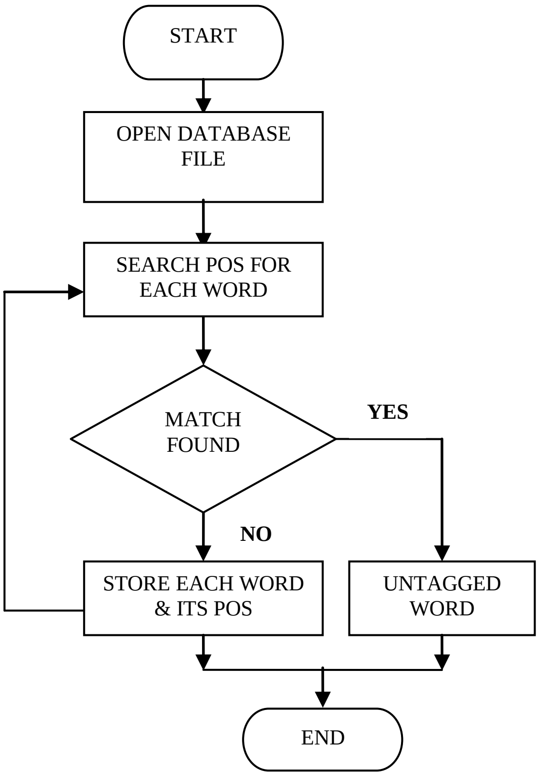


Fig. 4 Flow Chart of POS tagging module

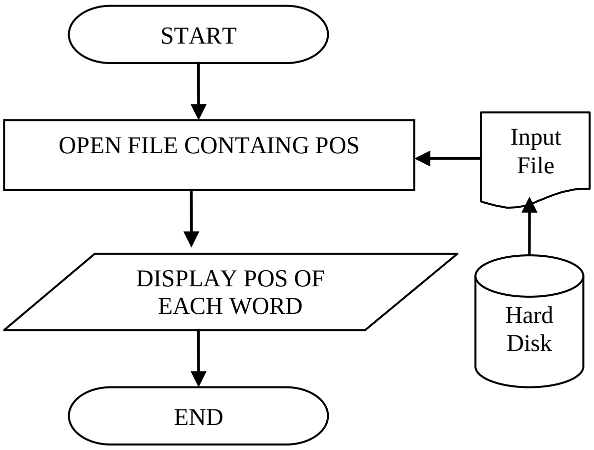


Fig. 5.5 Flow Chart of L1 Graph Module

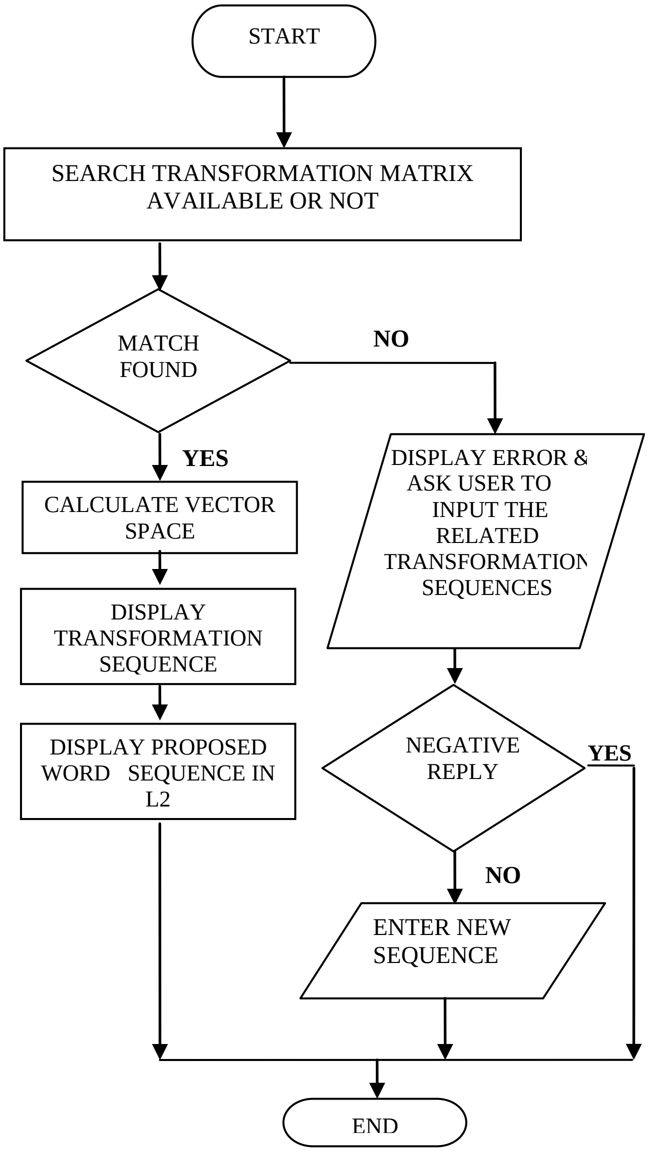

START
SEARCH TRANSFORMATION MATRIX AVAILABLE OR NOT
MATCH FOUND
NO
YES
CALCULATE VECTOR SPACE
DISPLAY TRANSFORMATION SEQUENCE
DISPLAY PROPOSED WORD SEQUENCE IN L2
DISPLAY ERROR & ASK USER TO INPUT THE RELATED TRANSFORMATION SEQUENCES
NEGATIVE REPLY
YES
NO
ENTER NEW SEQUENCE
END

Fig. 5.6 Flow Chart of Transformation Module

6. ADVANTAGES OF PROPOSED TRANSLATION MODEL

The Translation Model presented here has a numerous appealing features that the previous translation models lacked. It is based on the fact that each sentence in every language follows a particular rule of word sequence that is inherent to the language. In English language affirmative sentences follow the rule of: SUBJECT + VERB + OBJECT. Whereas in Hindi language the sentences follow the rule of: SUBJECT + OBJECT + VERB. When we translate an English sentence into a Hindi sentence, then not only words are transformed but the sequence of words is also changed.

Problem of abundant homophony

This arises due to fact that some words have different meanings in different contexts. The proposed Translation Model provides a geometrical representation of sentences in graphical form based on the Parts of Speech and therefore the translation will not take place according to the word sequence but in fact it will take according to the Parts of Speech sequence.

Problem of some words in source language having no counterpart present in the target language

Existing models were unable to cope up with the problem when some words plays a semantic role in source language but it is either absent or plays no role in the target language. The proposed Translation Model solves this problem by taking an equal number of vertices in both graphs L1 and L2. It implies that the parts of speech shall be common in source and target languages. Such words will be connected with other words by some edges in graph L1 but may not be in graph L2 or vice versa. Such type of words will have no translation at all.

Problem of collocation

Collocation is one problem where one faces the rigid words combinations whose meaning cannot be derived exactly by its word by word meaning. Existing systems cannot handle this situation but the proposed Translation Model overcomes this hurdle by adding a SUB-LINK process. This process searches for such phrases or combinations and then provides simple words using Thesaurus or Dictionary. The graph is constructed again in L1 after SUB-LINK has been applied to the sentence.

7. CONCLUSION & LIMITATIONS

The model is aimed at the application of Graph theory to translate English sentences into Hindi language. The proposed Translation model overcomes some of the shortcomings of the previous translation models. We can apply predefined transformations to vector space of L1 to get the corresponding L2. Moreover this proposed model performs at and above the state of art for modeling the contextual adequacy of paraphrases also. It presents serious shortcoming when the number of words in a sentence exceed 81. This is so because we have used a 9 X 9 matrix only for parts of speech. Although a little correction is provided by breaking the sentence into parts yet some work has to be done to overcome this shortcoming. It is presently uses linear search method to search a term out of approximate three lac words, which dampens its performance. Better search algorithms have to be applied to get faster results. The text files used by this model also lacks various words. Moreover some terms has abrupt Part of speech mentioned there. A need of better database is required for proper and adequate results.

8. REFERENCES

[1] Aho Alfred V. and Ullman Jeffrey D. (1972), The theory of parsing, translation, and compiling, ACM Classic Books Series.

[2] Alexandrescu A. and Kirchhoff. K., (2007), Data-Driven Graph Construction for Semi-Supervised Graph-Based Learning in NLP. In HLT.

[3] Alexandrescu Andrei and Kirchoff katrin, (2009), Graph-based learning for statistical machine translation, Proceedings of Human

[4] Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics.

[5] Ali Ola Mohammad, GadAlla Mahmoud, and Abdelwahab Mohammad Said, (2009), Improving Machine Translation using Hybrid Dictionary-Graph Based Word Sense Disambiguation with

[6] Semantic and Statistical Methods, International Journal of Computer and Electrical Engineering, Vol. 1, No. 5 December, 2009, pp 618-623.

[7] Amtrupy Jan Willers, (1999), Incremental Speech Translation, Springer.

[8] Becker Jörg and Kuropka Dominik, (2003), Topic based vector space model; Proceeding of 6th International Conference on Business Information Systems, Poznan, Poland, pages 7-12.

[9] Erk Katrin and Sebastian Pad´o (2006). A Structured Vector Space Model for Word Meaning in Context; Proceeding of conference on empirical method in natural language (EMNLP-08).

[10] Franz Josef Och, Christoph Tillman, and Hermann Ney, (1999), Improved Alignment Models for Statistical Machine Translation, Proceedings of EMNLP, pp 20–28.

[11] G. Salton. (1971), The smart retrieval system. Experiments in Automatic Document Processing, 1971.

[12] Gudába Milan, Horal Stanislav, Izakovic Ladislav, Kalinová Michaela and Snášel Václav. (2007). Geometrical approach for modeling semantics in linguistics; Proceeding of SYRCODIS Vol-256.

[13] Hinrich Sch¨utze. (1993), Word space. In Proceedings of the 1993 Conference on Advances in Neural Information Processing Systems, NIPS'93, pages 895–902, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[14] Hoede, C., Syntax and semantics, (2004), A comparison of the structuralistic language theory of Ebeling with knowledge graph theory, Memorandum No. 1710, Faculty of Mathematical Sciences, University of twente, Enschede, The Netherlands, ISSN 0169- 2690.

[15] Huntsville Alabama, (2010), Domain Model Translation Using Graph Transformations, 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'03).

[16] Dan Melamed, (2004), Statistical Machine Translation by Parsing, Proceedings of ACL.

[17] Jonathan Graehl and Kevin Knight, (2004), Training Tree Transducers, Proceedings of HLT-NAACL.

[18] Kenji Imamura, Hideo Okuma, Eiichiro Sumita, (2005), Practical

[19] Approach to Syntax-based Statistical Machine Translation, Proceedings of MTSUMMIT X.

[20] Kenji Yamada and Kevin Knight, (2001), A Syntax-based Statistical Translation Model, Proceedings of ACL, 2001.

[21] Lin Dekang, (2004), A path-based transfer model for machine translation, Proceedings of the 20th international conference on Computational Linguistics, Association for Computational Linguistics.

[22] Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, (1990), A Statistical Approach to Machine Translation, Computational Linguistics, 16(2), pp 79–85.

[23] Philip Koehn, Franz Josef Och, and Daniel Marcu, (2003), Statistical Phrase-based Translation, Proceedings of HLT-NAACL, 2003.

[24] Sahlgren Magnus, (2006), The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high dimensional vector spaces. PhD Dissertation, Department of Linguistics, Stockholm University.

[25] Zhang L., (2002), Knowledge Graph Theory and Structural Parsing, PhD thesis, University of Twente, Enschede, The Netherlands, ISBN 90-3651835-0.