Abstract—This paper presents an overview of the English to Hindi Machine Translation System using Rule-based translation system. Rule-based Machine Translation system translates a language into another language using linguistic information like dictionary and grammars and by following the semantic and syntactic rules of each language. Communication between people from different linguistic backgrounds still poses as a major problem. Hindi is one of the most popular and used languages in India but English even though being a universal language is not known by many Indians. The aim of this translation system is to provide a platform to do the translation which will enhance the well-learned society of Indians without the barrier of language.

Index Terms—Translation; Rule-Based Machine Translation; Statistical Machine Translation

I. INTRODUCTION

India is a country having 18 languages and 10 scripts, one of each territorial region. Though English is a universal and most used language worldwide, people generally prefer using their native language. This makes it difficult for Indians, especially the remote countrymen to read, write and understand English[2]. Cross-language communication between two countries plays a very crucial role to build a favorable environment for various benefits. In today’s age, the need to break this cross-language barrier can be easily done by various machine translation systems available on the Internet.

A lot of countries have made numerous efforts to develop and improve various machine translation systems. There have been enormous approaches taken to design and build a reliable translation system by a significant number of people during the last few years. The goal is to build a system that can be utilized by various applications depending on the requirements like a simple textual translation or a speech translational system[2]. At the initial development phase, many systems assume that the input sentence will be grammatically correct and will not require any checking. Hindi being the most understood and spoken language in India, the need of translating from English to Hindi language and vice versa is very much needed in all parts of the country. There are many systems build to produce this translation but some have not matured enough to handle all the ambiguity and semantics of the language.[2] Translation is not just doing a word to word conversions from the source language to destination language but handling the grammatical semantics and the syntax is the most crucial.

There are numerous systems available freely, each having their own strengths and weaknesses using various approaches like Rule-Based Machine Translation, Statistical Machine translation etc. but they all have a room of improvement and can be challenged for their limitations. This paper uses the Rule-Based Machine Translation approach which is built using dictionaries and other corpses as the data source along with various linguistic rules. The other two most widely used techniques are Example-Based Machine translation (EBMT) and the Statistical Machine translations (SMT). These two systems pull the rules automatically from the database instead of doing the initial analysis and generating the rules themselves. These systems are thus called data-driven approaches. Apart from these systems, there is Hybrid Machine Translation system that is a combination of multiple approaches.[2] The advantage of such systems is that they can incorporate all the positive features of each approach and can reduce the negative impacts of the same. Such a system can successfully deliver better quality output but the only drawback of such an approach is that it is very difficult to implement.

In Rule-based Machine translation system, various Semantic, Syntactic and Morphological pieces of information of both the languages are used.[3] There are different approaches like Direct, Transfer based, Knowledge-Based, Corpus-Based etc. In this system, some rules are automated and some need to be continuously updated by humans based on different conditions. The major focus should be on the analysis of the two languages to build these rules. Set of accurate and well-modeled rules will lead to a very accurate translation system.

Further, a little more about the Background is covered in section 2. Section 3 and 4 cover the Methodology and Results respectively. Future Work and Conclusion will be covered in detail in section 5.

II. BACKGROUND

During the last several years, there have been many translation techniques that have been developed. This section covers all the widely used approaches for Machine Translation in detail. It can be broadly classified into traditional and modern technology.[2]

A. Rule Based Machine Translation

Rule-Based Machine Translation system follows a classical approach. As mentioned above, this system is based on the detailed linguistic information of both the source and destination languages.[3] The source language is parsed and converted to the target language using various rules created from data sources like dictionaries and grammar of both the
languages. There are different approaches under this system like the Direct, Transfer based etc.

1) Direct Machine Translation: As the name suggests in this approach the translation of every word happens directly from the dictionary. Generally, this method is directed in one way and considers one word to translate at a time. To use this approach typically the system only needs the dictionary or such related data source. Typically, it works best for the languages that are very similar based on the grammar or the semantics that it uses. For example, translation between two Indian languages like Punjabi and Hindi that are very similar to each other and have almost the same grammatical rules can use this approach for translation. This system is very simple to build and less costly for similar languages as mentioned. Although, it is not very adaptive as it has no knowledge about the relationship between the words in the sentence. For the languages that are not at all similar to each other like English and Hindi, this approach is not recommended as the whole meaning of the sentence can be lost.

2) Transfer Based Translation: There are translation rules in this approach that are used to perform the translation. The input to this technique is a sentence and if the sentence matches one of the rules present in the database, it translates it into the destination language based on that rule. This approach also uses dictionaries as the data source. This approach is better than the previous approach since it removes a lot of ambiguity and incorrectness that would occur in the previous approach but the problem still prevails if the rule to the corresponding input sentence is not present then the output will be generated same as Direct Machine Translation.

B. Corpus Based Machine Translation

Corpus-Based Translation is the most used technique in today’s translation system. Statistical analysis of the corpus of the source and target languages is used in this method. The analysis is done using a large data set. There are two different methods of this system:

1) Statistical Machine Translation: This approach assumes that each sentence can be successfully translated from source to destination language. There are three different stages of this approach and the system computes every possible translation to the destination language from the source input sentence. The re-ordering of the sentences can be done automatically using the SMT which is very helpful. Advantages of this approach are that it is not confined to a set of languages and it is less costly compared to rule-based where the rules have to be updated manually and also depend on the languages. However, the disadvantage of this approach is that if the analysis of the language does not happen properly at the start due to less data resource then the results are not as expected.

2) Example Based Machine Translation: This approach depends on the corpus having translation examples between the source and destination languages. Retrieval module and an Adaptation module are the two modules used under the EBMT. The first stage is the Retrieval stage which looks up for a translated sentence from the corpus. The next stage uses the retrieved sentence and makes some adjustments to generated the final correct translation. Typically, the system finds the closest sentence to the given input sentence in the corpus based on some measures like word, semantic or syntactic similarity and retrieves its corresponding translation. Then, the retrieved sentence is checked for grammar and compared with the nouns of the input and generated sentences. Such adaptations like grammar correction, replacement of nouns etc are done in the Adaptation Phase. Example-Based Machine Translation might not work if the structure of the translated sentence is very different than the structure of the input sentence.

C. Hybrid Machine Translation System

Combination of various approaches into a single system forms a hybrid machine translation system. The goal is to develop a machine consisting of all the positive feature of different approaches and generate a system having a high level of accuracy. There can be many different combinations of hybrid system that can be generated. Following are the two most widely used or popular translation system here:

1) Statistical Rule Generation: As the name suggests it is a combination of the Statistical and Rule-Based approach. Similar to the statistical machine translation system, this technique also uses the data generated after the statistical analysis of the various lexical and syntactic rules. Once these rules and the data are produced the input is sent to the system and rules are applied similarly to the rule-based system. This technique is better in a way since it does not require creation of rules manually. This method is very successful in domain-specific fields since it relies largely on the similarity between the input sentence and the training corpus.

2) Multi-Engine Machine Translation System: Outputs of various Machine Translation engines are combined in this system. The output generated by this technique is the combination of output generated by various sub-systems used. This system has been very successful in various speech and text translation systems.

A large amount of work is already done in this field. Many systems have already been built using the above-mentioned approaches as shown in Table 1.

III. METHODOLOGY

A. Overview

This paper proposes a Rule-Based Machine Translation system. Thus, it depends on rules derived from the data sets to generate appropriate translation in the target language.
The overall system has been developed in a series of steps:
1) Conversion of the Audio Input to Text
2) Segmentation
3) Tagging
4) Translation
5) Rearrangement
6) Conversion of the translated text back to Audio Output.

In figure 1, the Flowchart of the approach taken to build this project is shown. All the steps are explained briefly in the following sections.

B. Data Preprocessing

Before developing the system, data was collected from various data sources. The datasets used to build this system are various Bilingual Dictionaries, Text Corpora, and Audio data sets. This step includes cleaning of the input text to make it favorable to be used by the system by applying multiple operations.[2] Generally, the proceedings involve checking the structure of the input sentence, checking for spelling error and removing them, replacing the titles stored in the dataset etc.

C. Algorithm

The algorithm followed to build the Machine is explained in great detail in this section.

1) Conversions of Audio input to Text:: The open source Google Translate API available in python is used to covert the input audio file into the text format which can be used to perform the translation. The generated text then goes through the data preprocessing phase where it gets cleaned to be used by the system successfully. The generated input text is then passed to the phase.

2) Segmentation:: During the segmentation phase, the input text is parsed and token of words are generated. A tokenizer is used to generate segments of the input text. It basically splits the data which in this case is a large text data with multiple sentences into series of words on the basis for some delimiter. Usually, space is used to split the sentence into an array of words. These words are then processed to convert every word into its root word. There are many stemming rules developed to make the conversions. Some of the rules are:

   Rule 1: Remove suffixes like ed, ly etc. Rule 2: Remove s if it is present at the end of the word Rule 3: Remove prefixes like In, dis etc.

   To explain the process with an example let us take a very simple input sentence say: Roma is secretly working on her project

   To begin with, the input sentence is broken into a series of words and each word is replaced by its corresponding root word. For the example taken here, The words secretly and working have their corresponding root words which are secret and work respectively. The remaining words are already in their root form. After this stage, the generated output text is Roma is secret work on her project.

   The rules mentioned above are just a sample set of all the rules used in the stemming phase. This step is vital since the words need to be in their root form to get correctly tagged.

   After this step, the generated sentence is passed to one of the most crucial steps which is Tagging.

   3) PoS Tagging:: During this phase, each segment is assigned a tag which gives it a grammatical information. To assign accurate tags to each word, there are several rules developed after thorough analysis of the language. There are tags already present in the English dictionary but the rules make the machine learn the structure and layout of the language in a better way. It is very important to have high accuracy in this phase since tags are nothing but property assigned to each word which determines the results of the further steps.

   Once the segments are assigned tags based on the rules generated the accuracy can be computed using the formula:
   
   Accuracy = Number of accurately tagged words / Number of inaccurately tagged words [1]

   The desired percentage of accuracy is more than 95

   Some of the rules generated to give the tags to the segments are as follows:

   • Rule 1: Nouns generally comes at the start if the sentence. If the sentence starts with a word not present in the dictionary and is at the start of the sentence, it is most likely to be a noun.
   • Rule 2: After analyzing the pattern of the language, it is discovered that if a sentence starts with the word there or here, the noun or the person in the context always follows a verb.
   • Rule 3: Another way to identify a Noun is by the words surrounded by it. For example, A, The, Numbers, This, That etc.
   • Rule 4: There are various words that do not affect the sentence in any way they are prepositions.
   • Rule 5: If there are more than one Subjects present in the sentence then the verb should be in the plural form.
   • Rule 6: If there are two people talking to or about one person, then the verb should be in a singular form.
In table 2, the tags used during this process are as mentioned.

To compute the accuracy of tagging, 5 different text corpora were tagged and the accuracy of each was computed. In table 3, the performance of the tagging is depicted.

Similarly, tagging of the Hindi words is performed in the next phase. Various grammar rules along with Hindi text corpora were used to generate these rules. An example of the rule created is that the feminine name or the verb associated with it usually ends with the अ. Similarly, the verb usually ends with आ.

These rules will be used to after the translation is performed to do the rearrangement and also perform morphological analysis to further train the machine.

4) Translation: Every word we get after the tagging phase is searched in the dictionary and the most appropriate translated word from English to Hindi is found to generate the initial translated sentence.[1] The process of translation is pretty straightforward. Every word is searched in the dictionary and if exists its corresponding translated word is extracted. Otherwise, it is assumed that the word is a proper noun or a name which can be confirmed by the tags assigned in which case the transliteration dictionary that does letter by letter translation is performed to obtain an accurate translation. To make this process even more efficient, there can be a way to save most common nouns or proper nouns to directly fetch the translation every time that word from the dataset appears.

In the rule-based approach, analysis of the structure of both the language to generate a correct translated text is very important.[1] In the next phase, the rearrangement if the words based on the analysis is explained in detail.

5) Rearrangement: The most crucial module in this phase is the analysis of the structure of both the source and target languages for different kinds of sentences. Different kinds of sentences mean Simple, Compound and Complex Sentences. The structure and the semantics of the sentences differ based on their property. Thus, the rules will differ slightly for the simple sentences from that of the complex sentences. By doing the analysis of the languages, patterns were generated for each type of sentence. These patterns will be then used for rearrangement of the translated sentence to generate correctly structured output as the final result.

The very first analysis that was drawn was the grammatical structure of the sentences in English and Hindi. The usual structure in the English language, if you take for example a simple sentence She goes to school is [subject, verb, object] whereas the same sentence in Hindi will be वो स्कूल जाती है which is [subject, object, verb]. From this analysis, a rule was generated which is like if the structure of the given statement in English is of the form [subject, verb, object] then the Hindi translated sentence should be [subject, object, verb]. The rules generated based on the analysis of these languages will be applied to the tagged sentences. Using the rule mentioned above, the tags given to the input and output sentences are compared and if they satisfy the rule, it is assumed that the translated text is in the correct structure. By such analysis, we can see how different are the two language structures from each other. These difference increase a bit more for complex and compound sentences.

An example of a compound sentence is Where did you go yesterday. This is of the structure [interrogation, place, subject, verb, time]. The translated sentence in Hindi would be कहा गया था which is [time, subject, interpolation, place, verb]. Based on this analysis we generated the rules. Now if the input sentence follows this structure sentence, the output sentence should follow the desired structure in the rule.

Once the structure of the translated sentence is correct, some more rules to employ the morphological changes made during the segmentation phase are applied again to the target language statement and matched with the grammar. Morphological rules are also developed based on the knowledge of the language.[3] For example, the Noun or the verb are affected by the gender, number i.e. singular or plural etc. Lets say input sentence is boys play cricket. Segmentation phase segments the sentence as boy + play + cricket. Next phase is tagging and translation and we finally get the translated sentence as ladka cricket khelta hai which is incorrect since it means A boy plays cricket. Thus, morphological analysis compared such properties of the sentence and gives the desired output.[3] To enhance the system and handle such challenges, along with the tags various other flags were assigned to each word and the rules were created to check if the flags of the input and output sentence match. If they do then the result is 100 percent accurate, else there is still some scope for improvement.

Similar to the accuracy computation in the tagging phase, accuracy of translation and rearrangement was computed for every type of sentence i.e. simple, compound and complex using those data sets.[2] After this phase, the translated text output is converted back to the audio file using the Google translate API available in Python.

IV. RESULTS AND DISCUSSIONS

At every phase, results were computed to understand the accuracy of the system at every level. For each level, if the accuracy is less than the expected value then some more rules

<table>
<thead>
<tr>
<th>Tags</th>
<th>Property</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Proper Noun</td>
<td>The Name of a person, place, company etc.</td>
</tr>
<tr>
<td>NOUN</td>
<td>Noun</td>
<td>Can be person, things, animal or place.</td>
</tr>
<tr>
<td>VERB</td>
<td>Verb</td>
<td>Words that denote some Action</td>
</tr>
<tr>
<td>PRON</td>
<td>Pronoun</td>
<td>Words used instead of a noun</td>
</tr>
<tr>
<td>ADJ</td>
<td>Adjective</td>
<td>Describing property of the Noun.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Total Words</th>
<th>Accurately Tagged</th>
<th>Inaccurately Tagged</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140</td>
<td>120</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>535</td>
<td>500</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>520</td>
<td>480</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>730</td>
<td>670</td>
<td>60</td>
</tr>
</tbody>
</table>

TABLE II

TAGS

TABLE III

ACCURACY OF TAGGING
were added to improve the system and let the system generate better results.

A. Segmentation

The text input is segmented into tokens and rules were applied to transfer each token into its root word.

Let’s take an example to understand the results at every stage: The input sentence: grandma said that you need to find three keys that can unlock the treasure which can be found deep in the cave.

After segmentation phase: [ grandma + say + that + you + need + to + find + three + key + that + can + unlock + the + treasure + which + can + be + find + in + the + deep + cave ]

This segmented text is sent to the tagging module to assign tags to every word.

B. Tagging

Following is the result obtained after tags are assigned to each word for the input sentence:

[ grandma<Subject> + say<verb> + that + you<subject> + need<verb> + to + find<verb> + three<number> + key<object> + that + can + unlock<verb> + the + treasure<object> + which + can + be + find + in + the + deep<adjective> + cave<object> ]

There were many improvements made to this module while developing the rules. The accuracy of the overall phase was improved by trying various corpora one at a time and handling all different possible scenarios. For every corpus, the accuracy reduced a bit compared to the previous corpus and thus additional rules were generated to improve the accuracy.

C. Translation and Rearrangement

The output of the tagging phase is then passed to the translation and rearrangement phase and the final translated sentence is as shown in figure 2. The figure is console output of the system.

This was the most crucial part of the system. In the beginning, the output of the sentences did not make sense. The same procedure of learning from different datasets and adding rules to the system was adopted to improve the accuracy of the system. In this step, the machine learned about the structure and the meaning of both the languages into account. Once the translated sentences obeyed the rules and the structure of the language, the gender and number parameters were not as desired. Some more update to the rules and code was done to handle them. All the improvements were made while developing the system and not at the beginning.

The system also handles jargons, idioms, and proverbs by making use of already available datasets to translate such sentences. Figure 3 depicts the output where a simple jargon is handled successfully.

D. Accuracy charts

Apart from the results, at every stage, a chart to learn about the performance of the each step were created.

There are two charts included in this section one depicting the performance of Tagging and another depicting the performance of Translation and rearrangement.

Figure 4 shows the performance of tagging for two corpora. From the first one, we can see that the ration of correctly tagged words to the incorrectly tagged words is 9:1 which is very good. While for the second one, the ratio drops to about 4:1. The second corpus was a large one compared to the first one. Thus, in the first set for every 9 correctly tagged words, there was only 1 incorrectly tagged word but in the second set for every 4 correctly tagged words, there was an incorrectly tagged word which is considerable drop.

Figure shows the performance of Translation and Rearrangement.
rangement. This is a percentage comparison of accuracy between a compound and simple sentence for a small and a large corpus. The first one is the accuracy of translation and rearrangement for a small corpus and it is clear that the accuracy is remarkable. However, for the second corpus which is the large corpus, the accuracy is reversed. The accuracy of the compound sentences is more than the simple sentences since the corpus contained a lot of compound sentences. The machine can be trained with more corpora to generate better results and have a high performance.

V. Future Work and Conclusion

In this paper, a Rule-Based Machine Translation system to translate from English to the Hindi language is successfully implemented. The Google translate API available in the python library was integrated to perform the Audio to Text translation.

As mentioned above, a Hybrid system that could incorporate features of various methods will be better than just a rule-based system. This system can be enhanced and improved by adding few features of the Statistical Machine translation system like adding some statistical analysis to the system can increase the accuracy of the system.

The system also successfully handles jargons, idioms but there is a scope to handle sarcasm and more stop words to make the system more useful and friendly.

Another addition to the system can be generating a UI or an application for the user to make it more user-friendly and widely used.

ACKNOWLEDGMENT

The author would like to thank Dr. Carol Romanowski, Associate Professor at Rochester Institute of Technology, for her guidance during the project. I would also like to thank my colloquium instructor Dr. Reynold Bailey for his enlightening feedbacks after every milestone.

REFERENCES


