Hindi to Punjabi Machine Translation System

Vishal Goyal
Lecturer
Department of Computer Science
Punjabi University Patiala
Vishal.pup@gmail.com

Gurpreet Singh Lehal
Professor
Department of Computer Science
Punjabi University Patiala
gslehal@gmail.com

1. System Description

This proposed system is developed on Windows Platform. The system is divided into broadly three stages: Preprocessing, Translation Engine, and Post Processing stage. The words from source language are chosen, their equivalents in target language are found out from the lexicon and are replaced to get target language. The source text is passed through various pre processing phase and out put is also passed through a post processing phase. Following is the description of various steps of this architecture.

2. System Architecture

The system architecture as shown in figure 1, has following stages through which the source text is passed:

2.2.1 Pre Processing

2.2.1.1 Text Normalization

It works on spelling standardization issues, thereby resulting in multiple spelling variants for the same word. During this phase of Pre Processing phase , rules specific to Hindi which can handle such variations have been used for making the input text normalized for better accuracy. For example we found widely used spelling variations for the Hindi word अंग्रेजी (*aṅgrējī)* as shown below:
अँग्रेजी, अंगरेजी, अन्ग्रेजी, अँगरेजी, अंग्रेजी, अंग्रेज़ी

2.2.1.2 Replacing Collocations: It means finding and replacing those combinations of words in Hindi that cannot be translated word to word . This activity helps a lot in increasing the accuracy of the system. For example, the collocation उत्तर प्रदेश (*uttar pradēsh)*, if translated word to word, will be translated as ਜਵਾਬ ਰਾਜ (*javāb rāj),* But it must be translated as ਉੱਤਰ ਪ੍ਰਦੇਸ਼ (*uttar pradēsh)*.

2.2.1.3 Replacing Proper Nouns: It means finding and replacing those combination of words in the input text that are acting as names of person, bank, river, ocean, days of week, months of year, university, cooperative society etc. For example: कमल गोयल (*kamal gōyal*) is a proper noun.

2.2.2 Tokenizer

The tokenizer takes the text generated by previous text as input. This module, using space, a punctuation mark, as delimiter, extracts tokens (word) from the text and gives it to Translation engine for analysis. This process is repeated for the whole text.

2.2.3 Translation Engine

The translation engine is responsible for translation of each token obtained from the previous step. It uses various lexical resources for finding the match of a given token in target language. Following is the description of how a token is passed through various modules.

2.2.3.1 Analyzing the word for Translation /Transliteration

The token obtained in the previous stage is passed through various stages.

2.2.3.1.1 Identifying Titles: The token is checked whether it is a title like प्रो(prō), श्रीमती(shrīmtī) etc. If the current token is found to be a title, then the token next to it, should be transliterated instead of translation.

2.2.3.1.2 Identifying Surnames:
The token is checked whether it is a surname like अग्रवाल (agrvāl), ओबेरॉय (ōbērāy) etc. If the current token is found to be a surname, then the token previous to it, should be transliterated instead of translation.

2.2.3.1.3 Lexicon Lookup:
If the token does not satisfy above two steps, then it is looked into the lexicon for a match for direct word to word translation.

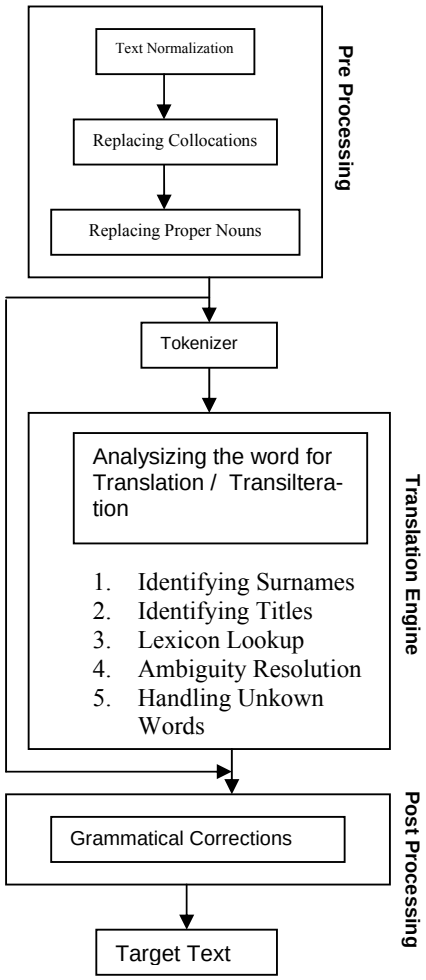

Figure 1 : Overview of Hindi-Punjabi Machine Translation System

2.2.3.1.4 Resolving Ambiguity:
If the token is not present in the lexicon for direct translation, it is looked into the database of ambiguous words. If this token is found to be ambiguous, then disambiguity is resolved with the help of n-gram language modeling. The system uses bigram and trigram databases, which contains one and two words respectively in the vicinity of an ambiguous word and corresponding meaning for that particular context.

2.2.3.1.5 Unknown Words:
If all the above modules fail to analyze the token, it is considered to be foreign/unknown word. Such words first pass through the morphological analysis phase based on the rules for inflections in Hindi words. Morphological generator generates the transliterated word using the inflectional rules and then checks the generated word in the Punjabi unigrams database for its genuinity. If this new generated word is found in the Punjabi unigrams, it is considered for translation otherwise the token is sent to transliteration module for transliteration.
Transliteration Module is the major module in the system that uses various rules specifically designed from the translation point of view.

2.2.4. Post Processing
After converting all the source text to target text, there are some of the grammatical errors that need to be corrected. For this purpose, we have formulated the rules for correcting the grammatical errors. Such rules have been implemented using Regular expressions and Pattern matching. This Post Processing phase is responsible for correcting grammatical errors in the generated output.

Presently, the system is available online for use at website http://h2p.learnpunjabi.org.

3 Results

3.1Subjective Test Analysis
The accuracy of the system has been evaluated as 94% on the basis of intelligibility test and 90.84% on the basis of accuracy test. The accuracy score is comparable with other similar systems.

3.2 Error Analysis
Word Error rate, which is the percentage of erroneous words from all words, is found out to be 2.10%. It is comparably lower than that of the general systems like Salt, Incyta, Internostrum, where it ranges from 3.0 to 4.9 (Tomas J. et.al., 2003). The Sentence Error rate is found out to be 20.15%.

Conclusion:
This is one of most accurate Machine Translation system developed so far for Indian languages. Simialr system has been developed by CDAC Noida and is available at http://sampark.iiit.ac.in which shows very less accuracy than the one discussed above.