A Project Report

ON

Morphological Analysis & Translation from Telugu to Kannada

BY

Sharath K.P – 1PI13CS141

Shreyas G – 1PI13CS153

T.N.Suhas – 1PI13CS176

GUIDE
Dr. Dinkar Sitaram
Centre for Cloud Computing and Big Data,
PESIT-CSE
Bangalore

August 2015 – December 2015

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
PES INSTITUTE OF TECHNOLOGY
(an autonomous institute under VTU)
100 FEET RING ROAD, BANASHANKARI III STATE
BANGALORE-56008

PES Institute of Technology,
(An Autonomous Institute under VTU)
100 Feet Ring Road, BSK 3rd Stage, Bangalore-560085

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Special Topics: Mini Project work entitled **Morphological Analysis & Translation from Telugu to Kannada** is a bona-fide work carried out by **(Sharath KP – 1PI13CS141, Shreyas G – 1PI13CS153, T.N.Suhas – 1PI13CS176)** in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic semester August 2015 to December 2015.

Signature of the Guide	Signature of the HOD
Prof. Dinkar Sitaram	**Prof. Nitin .V. Pujari**

Sharath K.P	**– 1PI13CS141**
Shreyas G	**– 1PI13CS153**
T.N.Suhas	**– 1PI13CS176**

ABSTRACT

Information and language always go hand in hand. One cannot really realize the information without knowing the language in which it is written. Hence there is always scope for multilingualism. Morphological analysis of a language and thus its translation help us a great deal in being multilingual.

Our project deals with morphological analysis of Telugu language and thus translating it into Kannada. That is, given a Telugu input text, we generate corresponding text in Kannada language and visa-versa.

Realizing that there are very few such software that actually go in depth with the details of the Dravidian syntax, we took that as our project. So, we decided to develop software for the Dravidian languages (like Kannada, Telugu, etc)

This project has a lot of necessity especially for the immigrants who need assistance to learn the language of the land.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all the people who have been associated with this project directly or indirectly and have helped us with working on it.

We would like to extend our sincere thanks to **Prof. M. R. Doreswamy**, founder of PESIT, **Prof. D. Jawahar**, CEO of PESIT and **Dr. K.S. Sridhar**, Principal PESIT, for the encouragement, motivation and continuous support they have provided to us.

We would like to thank **Prof. Nitin V. Pujari,** Head of the Department CSE for his constant support and encouragement.

We would also like to thank **Dr.Dinkar Sitaram**, an esteemed Faculty of CSE Department and also our guide, for providing us an opportunity to work on a novel concept, and also for his guidance throughout the duration of this project. We would also want to thank him for his continuous inputs and ideas for this project. We would also like to thank **Mrs.Santwana Chimalamarri** who guided us during the course of our project.

We would also like to thank our parents and other family members for their continuous support to us, without which, we would not have been able to achieve whatever we have.

We would also like to thank all our friends who have directly or indirectly helped us throughout the duration of the project.

Table of Contents

1. INTRODUCTION

2. PROBLEM DEFINITION

3. LITERATURE SURVEY

4. PROJECT REQUIREMENT DEFINITION

5. SYSTEM REQUIREMENT SPECIFICATION

6. DESIGN

7. PSUEDO CODE

8. RESULTS

9. CONCLUTION

10. FUTURE ENHANCEMENTS

11. BIBLIOGRAPHY

1. INTRODUCTION

Language has always been a fascination to the mankind. Language has not just remained as a medium of communication but has gone a far way in defining an identity of a person.

Machine translation, abbreviated as MT is a subfield of computer-linguistic that facilitates the use of software to translate one language to another.

Translation is not at all a new idea. French philosopher René Descartes proposed the idea of universal language of different dialects sharing a same symbol way back in 1629.

The potential and progress of machine translation has always been debatable. The possibility of achieving fully automatic machine translation of high quality is still questionable. Yet the search and research in this field goes on and have yielded favorable results.

Translation can be described as a two step process of decoding or understanding the given language and then re-encoding the meaning into the target language. In our project we implement the computational morphology.

Morphology refers to the study of the way the words are built from smaller meaning-bearing words called units called morphemes. Inflection is the combination of a a word stem with a grammatical morpheme resulting in word of the same class as the original stem and filling some syntactic function like agreement.

The notable rise of social networking on the web in recent years has created a niche for the application of machine translation software – in utilities such as Facebook, or instant messaging clients such as Skype, GoogleTalk, MSN Messenger, etc. – allowing users speaking different languages to communicate with each other. Machine translation applications have also been released for most mobile devices. The application of this technology in medical settings where human translators are absent is another topic of research however difficulties arise due to the importance of accurate translations in medical diagnoses.

2. PROBLEM DEFINITION

There is a need to have software that can translate a given text of one language into another. Building such software would help the people who are interested in learning a new language and would also enhance translation of literary works. Given an input in notepad (.txt) format in one language the output should be a text file of another language (.txt) format.

Thus, build software that automatically reads out a given input text and morphologically analyze the given language.

Morphological analysis is a vital step for languages like Kannada and Telugu. Morphology brings down the parallel corpora requirement. Conversely, for a fixed amount of corpora the accuracy level goes up if the morphological analysis is applied.

We aim to set up lemma based mappings instead of words mappings.

3. LITERATURE SURVEY

At the beginning of our project, we carried out a search on the existing morphological analyzer tools available for Indian Regional Languages. We found that there are tools which efficiently handle the languages of the Indo-Aryan origin but very few for the Dravidian languages such as Kannada, Tamil, Telugu and Malayalam.

Also on more intensive search, and interaction with immigrants who were keen to learn the language of the state but could not due to their busy schedules, we concluded that the need of the hour was translator software for the Indian languages.

We went through "Verbs are all action lies: Experiences of shallow parsing of a morphologically rich languages" to have a better understanding on shallow parsing and Part Of Speech Tagging.

We also went through "A grammar of modern Telugu" by Krishnamurthi.B to have deep insight on the word structure of Telugu.

In GOOGLE we found many translator apps but many of them failed to analyze the intricacies of the Dravidian languages.

4. PROJECT REQUIREMENT DEFINITION

The usage of this project is mainly for the computer-literate people who are interested in learning a new language.

Any person with basic knowledge on how to compile a code must be able to operate it easily.

A notepad file with some text in it should be provided in the work space of the project and the file name must given as input to the code at the beginning.

A simple GUI and translation with a better blue score evaluated over corpus can be implemented (future enhancements) to make this project nearer to a common man.

5. SYSTEM REQUIREMENT SPECIFICATION

The basic system requirements for our project are:

- Any Linux distro (Ubuntu 14.04 used).
- SFST – Stuttgart finite state transducer. It is a very useful tool to perform morphological analysis of a given language. SFST is the elementary requisite for this rule based translation. This transducer enables us to check if a given word belongs to the specified language by parsing through the finite state Automaton.
- Python 3 is used to link SFST and facilitate two operations i.e. Generate and analyze on morphologically rich language.

6. DESIGN

Stuttgart finite state transducer is a tool which is used to implement computational morphologies. It assumes some basic knowledge about finite automaton and formal languages from the reader’s side.

A finite state transducer is an finite state automaton where each transition is labelled with a symbol rather than a single symbol. An FST is used to map strings to other strings. The mappings are reversible. The disjunction operator of two transducers is equivalent to union of two set of string pairs.

A computational morphology analyzes inflected words.

Kannada and Telugu words (nouns) are inflected based on number person and gender.

Consider an Fst based rule:

- Rule: $R1$ = ು<=> ೆ (#:<>ನ)

L(ು) <= (ೆ)R is similar to the regular expression[!.*L(ು). & !(ೆ):(ೆ)R.*] that ೆ should be mapped to (ು) if it appears in between L and R.

(ು) => (ೆ) is similar to the regular expression [!(!(.*L)(ು) : (ೆ).* | .* (ು) : (ೆ) !(R.*)] that ” ು " should be mapped to " ೆ " if preceded by a Vyanjana (It should be preceded by the character here any char (i.e. [.*L]) which is defined by the user)

The symbol “#” represents the boundary about which the word is split during the analyze mode.

All the characters should be defined before they are used. The Rule should be defined by the programmer before it is compiled else an error is raised.

Consider the analysis of the word "ಊರಿನಲ್ಲಿ". Here the root word is "ಊರು" which is a Fourth declension noun. According the Fst rule mentioned above the word "ಊರಿನಲ್ಲಿ" is split about "ನ" the character “ು“is mapped to the character “ಿ". The word is inflected as Saptami Vibhakti.

Therefore the output of the transducer on Analyze is:

ಊರು<ForDecNoun><Sapthami>

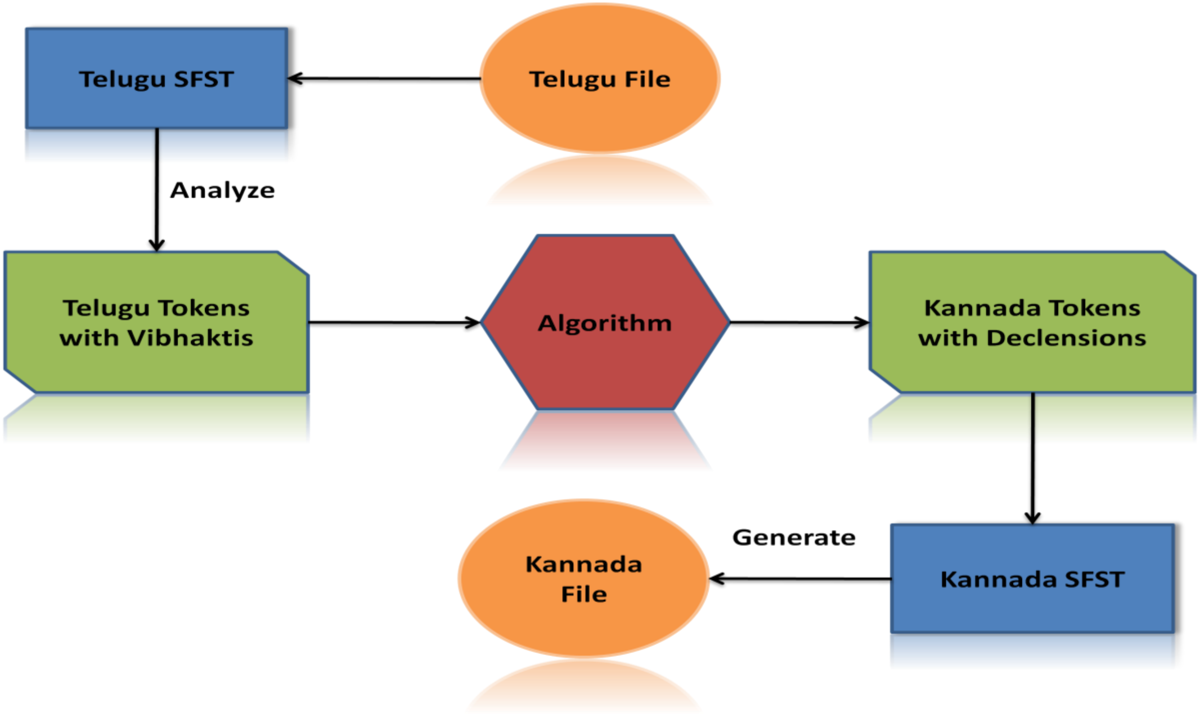

Figure 6.1: Control Flow of Translation

According to the Figure 6.1, this depicts the flow of translation on how a Telugu text is translated to Kannada Text. Using SFST we analyze the input text (Telugu word). The analyzed output is then used for generating the words which are handled in our python code. We then map the Telugu root word to Kannada root word by making use of the Dictionary that we have initialized.

7. PSEUDO CODE

The translation of a Telugu file to Kannada by morphological analysis follows the algorithm as mentioned below. The input is a .txt file encoded in UTF-8 format. During the flow of code different intermediate files are generated used by the SFST. The output from SFST is used for Translation into normal Kannada words.

```
Algorithm:
#Telugu - Kannada
begin
    intialize Telugu - Kannada Dictionary
       while next line in Input Telugu file
              split each line into seperate words as List L

              while word in L
                     pass word to Telugu SFST(Analyze)
                     write output in an Intermediate File 1
                     read next word from L
              end(while word)

              while next line in Intermediate File 1
                     if line != no result
                            convert result into Kannada format using
                            Dictionary and Mapping rules
                     else
                            pass line
                     write Kannada format in an Intermediate File 2
                     read next line from File 1
               end(while next line in File 1)
```

```
            while next line in Intermediate File 2
                  pass word to Kannada SFST(Generate)
                  write output of Generate to Output Kannada File
                  read next line from File2
           end(while next line in File 2)

           read next input line from Input Telugu file
    end(while next line in Telugu File)
end
```

Consider an example input: **ರವಿಯೊಕ್ಕ ಭಾರ್ಯ ಗೀತ**.

The Telugu to Kannada Dictionary is internalized as follows:

D = {” ರವಿ ” : “ ರವಿ ” , ” ಭಾರ್ಯ ” : ” ಹೆಂಡತಿ ” , ” ಗೀತ ” : ” ಗೀತ ” }

Splitting the sentence into words and forming a List L.

L = [“ರವಿಯೊಕ್ಕ “ , ” ಭಾರ್ಯ ” , ” ಗೀತ ”]

Passing each word separately into Telugu SFST Analyze and writing the output into a file we get:

ರವಿಯೊಕ್ಕ – ರವಿ <Noun><ShasthiS>

ಭಾರ್ಯ – ಭಾರ್ಯ <Noun><root>

ಗೀತ – ಗೀತ <Noun><root>

This Output is taken from the file and converted into Kannada format using the Dictionary and the Mapping Rules. The converted format is as follows:

ರವಿ<SenDecNoun><Shasthi>

ಹೆಂಡತಿ<ThdDecNoun><root>

ಗೀತ<FirstDecNoun><root>

These output lines and passed line by line to Generate of Kannada to get the final translated Kannada output.

Final Output: **ರವಿಯ ಹೆಂಡತಿ ಗೀತ.**

8. RESULTS

On giving inputs (i.e. Kannada and Telugu words) to SFST, we found that the transducer splits the given word giving its root word and its inflection for both Kannada and Telugu.

Cases of out-of-lexicon inputs are returned with no result.

Also input cases that are syntactically wrong in the given language are handled well.

The code also handles the cases when the input is a non member of the given language.

Hence, we were able to produce a word to word translation of the given sentence to both the language.

9. CONCLUSION

After good four months that we spent on the project analyzing the syntax and the grammar of Kannada and Telugu, we are very happy with the tool we have formulated.

Our tool is able to read simple Kannada sentences and phrases and translate word by word into Telugu.

We have achieved, to some extent, what we had hoped to achieve. We have also started taking initiative to add more words into the lexicon and expand the accessibility and utility of the tool.

Morphological analysis is a vital step for languages like Kannada and Telugu.

Morphology brings down the parallel corpora requirement. Conversely, for a fixed amount of corpora the accuracy level goes up if the morphological analysis is applied. The lemma based mappings was more efficient in translating two languages.

10. FUTURE ENHANCEMENTS

We are planning to take our tool one step further by expanding to a huge corpus and with a good blue score.

Also, we would like to deal with complex sentences and translate them efficiently.

We would also like to enhance our code which translates word-by-word to a tool which allows translation line by line. We would also wish to implement Veterbi's algorithm which plays an important role in machine translation.

11. BIBLIOGRAPHY

1. High-Performance, Language-Independent Morphological Segmentation by Sajib Dasgupta and Vincent.
2. Morpheme Segmentation for Kannada Standing on the Shoulder of Giants by Suma Bhat
3. A Paradigm-Based Finite State Morphological Analyzer by Pushpak Bhattacharyya, Harshada Gune, Mugdha Bapat
4. Novel approch for morphing telugu nouns forms using Finite state transducers by Sneha D.L, Dr. Bharadwaja Kumar Asst.Professor VIT University
5. SFST Manual by Helmut Schmid , Institute of Natural Language Processing , University of Stuttgart