

Functional Specifications of **Transliteration**

1. Introduction

Transliteration module helps to convert utf files written in one Indian language to another. The input files can be in SSF (Shakti Standard Format) or simple UTF/text files. This makes use of Conversion modules of the corresponding Indian language in which mappings are written for their UTF characters. In SSF file; TKN, LEX, VIB, NAME and HEAD fields will be converted when they start with a '@' symbol. In case of text files, the whole words will be converted.

2. Input-Output Specifications

2.1 Input file is SSF

Input:

```
((          CAT_      <fs af=TKN_,, , , ,V_,'name='@N_ '>
@TKN_1      CAT_1      <fs af=@TKN_1, , , , ,@V_1,'
head='@H_1'>
))
```

Output:

```
((          CAT_      <fs af=TKN_,, , , ,V_,'name='@N_2'>
@TKN_3      CAT_1      <fs af=@TKN_3, , , , ,@V_3,'
head='@H_3'>
))
```

2.2 Input file is Text Format

Input:

W_1 W_2 W_3 W_4

Output:

W_5 W_6 W_7 W_8

Examples:

Here source language is Hindi and target language is Telugu

1. Input file is SSF

INPUT:

```

ADDR_  TKN_  CAT_          OTHR

<Sentence id="1">
1      ((      NP      <fs af='@लडका,n,m,pl,,d,, ' head="लड़के">
1.1    @उस    DEM    <fs af='@उस,pn,any,sg,3,o,, '>
1.2    लड़के  NN      <fs af='@लडका,n,m,pl,,d,@ने, ' name="@लड़के">
1.3    @ने    PSP     <fs af='@ने,psp,,,,, '>
      ))
2      ((      NP      <fs af='@रादा,n,f,sg,3,,, ' head="@रादा">
2.1    @रादा  NN      <fs af='@रादा,n,f,sg,3,,, ' name="रादा">
2.2    का     PSP     <fs af='का,n,m,sg,,d,, '>
      ))
3      ((      NP      <fs af='केला,n,m,sg,,d,, ' head="केला">
3.1    केला   NN      <fs af='केला,n,m,sg,,d,, ' name="केला">
      ))
4      ((      VGF     <fs af='@काया,n,f,sg,,d,, ' head="@काया">
4.1    @काया   VM     <fs af='@काया,n,f,sg,,d,, ' name="@काया">
      ))
5      ((      NP      <fs af='ता,n,m,sg,,d,, ' head="ता">
5.1    ता     NN      <fs af='ता,n,m,sg,,d,, ' name="ता">
      ))
</Sentence>

```

OUTPUT:

```

<Sentence id="1">
1      ((      NP      <fs af='@లడకా,n,m,pl,,d,, ' head="లడకే">
1.1    @ఉస    DEM    <fs af='@ఉస,pn,any,sg,3,o,, '>
1.2    లడకే  NN      <fs af='@లడకా,n,m,pl,,d,@నై, ' name="@లడకే">
1.3    @నై   PSP     <fs af='@నై,psp,,,,, '>

```

```

))
2 (( NP <fs af='@रादा,n,f,sg,3,,,' head="@रादा">
2.1 @रादा NN <fs af='@रादा,n,f,sg,3,,,' name="रादा">
2.2 का PSP <fs af='का,n,m,sg,,d,,,'>
))
3 (( NP <fs af='केला,n,m,sg,,d,,,' head="केला">
3.1 केला NN <fs af='केला,n,m,sg,,d,,,' name="केला">
))
4 (( VGF <fs af='@क्या,n,f,sg,,d,,,' head="@क्या">
4.1 @क्या VM <fs af='@क्या,n,f,sg,,d,,,' name="@क्या">
))
5 (( NP <fs af='ता,n,m,sg,,d,,,' head="ता">
5.1 ता NN <fs af='ता,n,m,sg,,d,,,' name="ता">
))
</Sentence>

```

2. Input file is Text format

INPUT:

राणा सांगा को हराने के लिए मुगलों ने यहाँ कई बार आक्रमण किया जिनमें कई बार राणा
 सांगा घायल हुए |
 यहाँ का सबसे बड़ा आकर्षण समुद्र का संगम है |
 यहाँ पहुंचना भी मुश्किल नहीं है |
 पांडिचेरी में स्मारकों की कमी नहीं है |
 लेकिन स्कॉटलैंड की पहचान सिर्फ यहीं तक सीमित नहीं है |

OUTPUT:

రాణా సాంగా కో హరానే కే లిఏ ముగలోం నే యహాఁ కఈ బార ఆక్రమణ కియా జినమేం కఈ బార రాణా
 సాంగా ఘాయల హుఏ |
 యహాఁ కా సబసే బడ్రా ఆకర్షణ సముద్రా సంగమ హై |
 యహాఁ పుంచనా బీ ముశ్కిల నహీం హై |
 పాండిచేరీ మేం స్మారకోం కీ కమీ నహీం హై |
 లేకిన స్కాట్లెండ్ కీ పహచాన సిర్ఫ యహీం తక సీమిత నహీం హై |

3. Flow Chart of Transliteration module

Flow chart of the Transliteration module will show the over all view of the program control flow from one module to other.

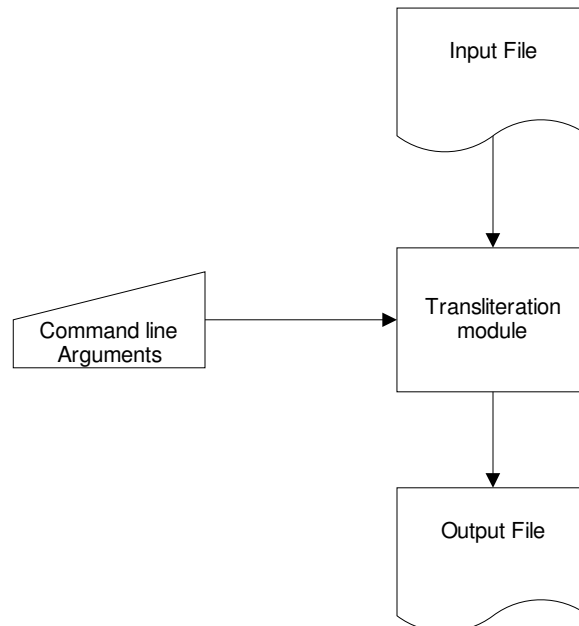


Fig.1- Basic Flow

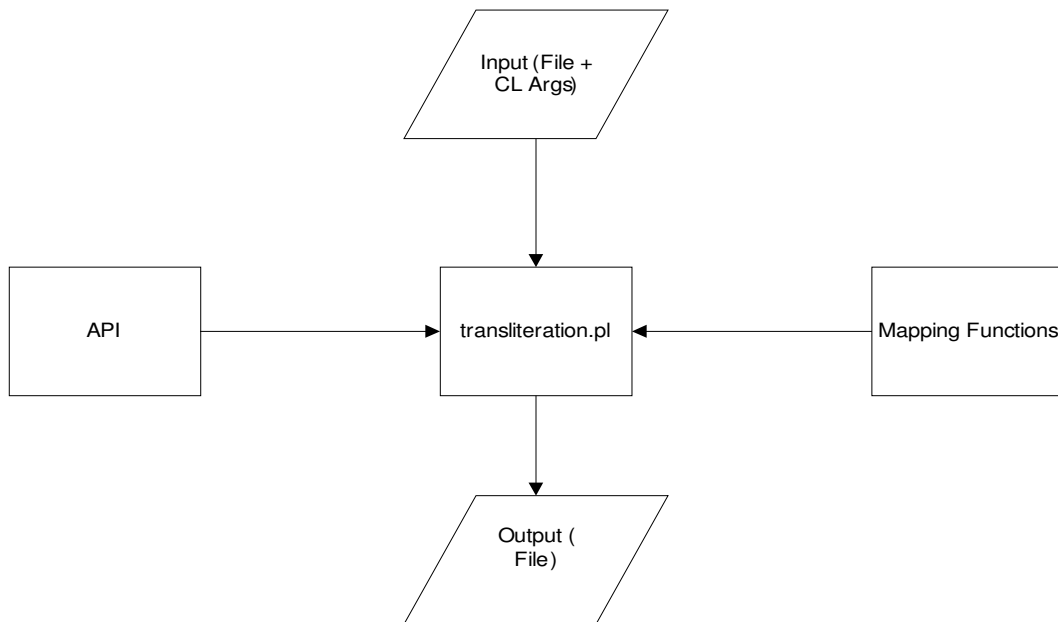


Fig.2- Transliteration

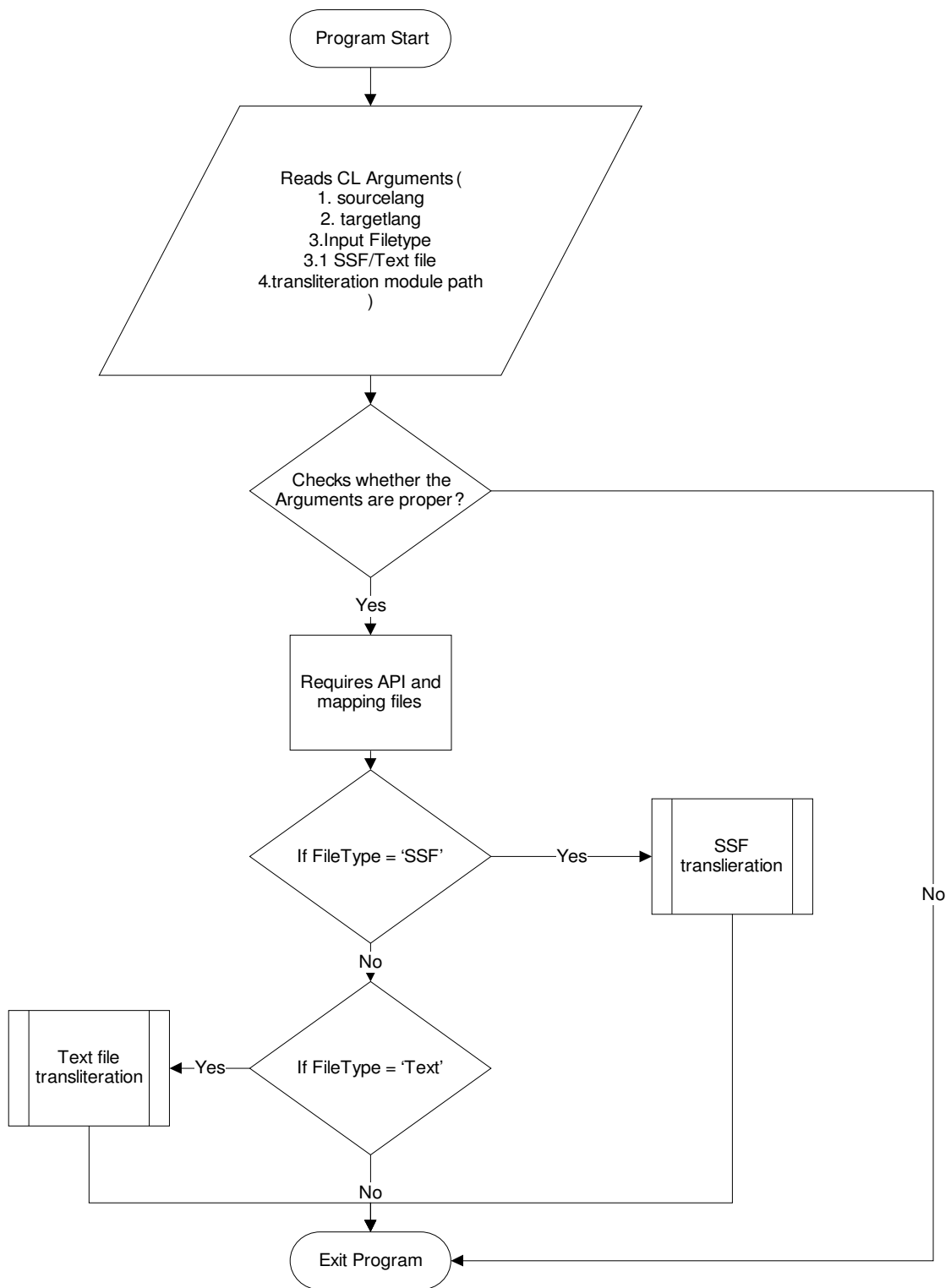


Fig.3-Transliteration.pl (Main program execution)

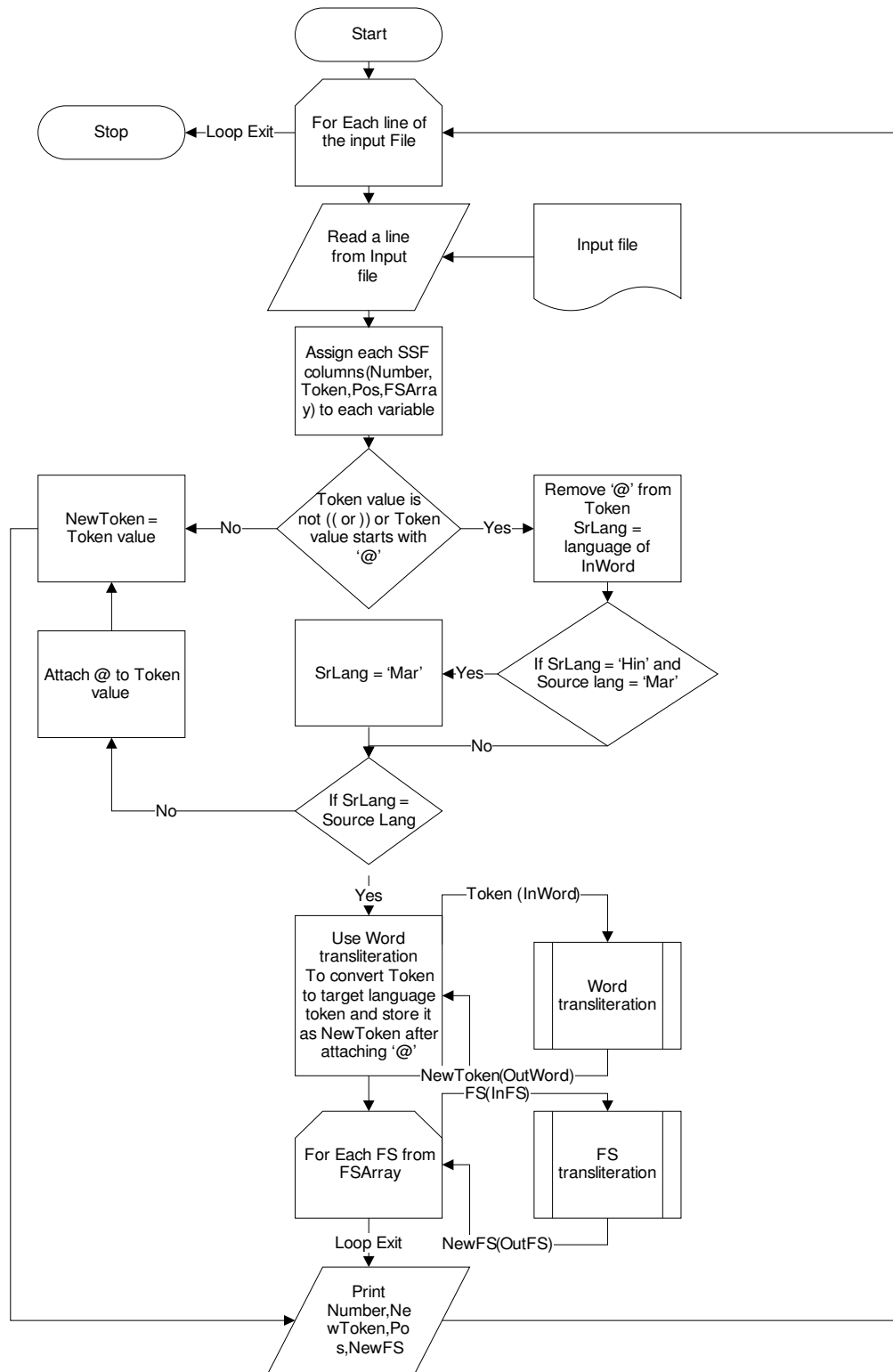


Fig.4- SSF transliteration

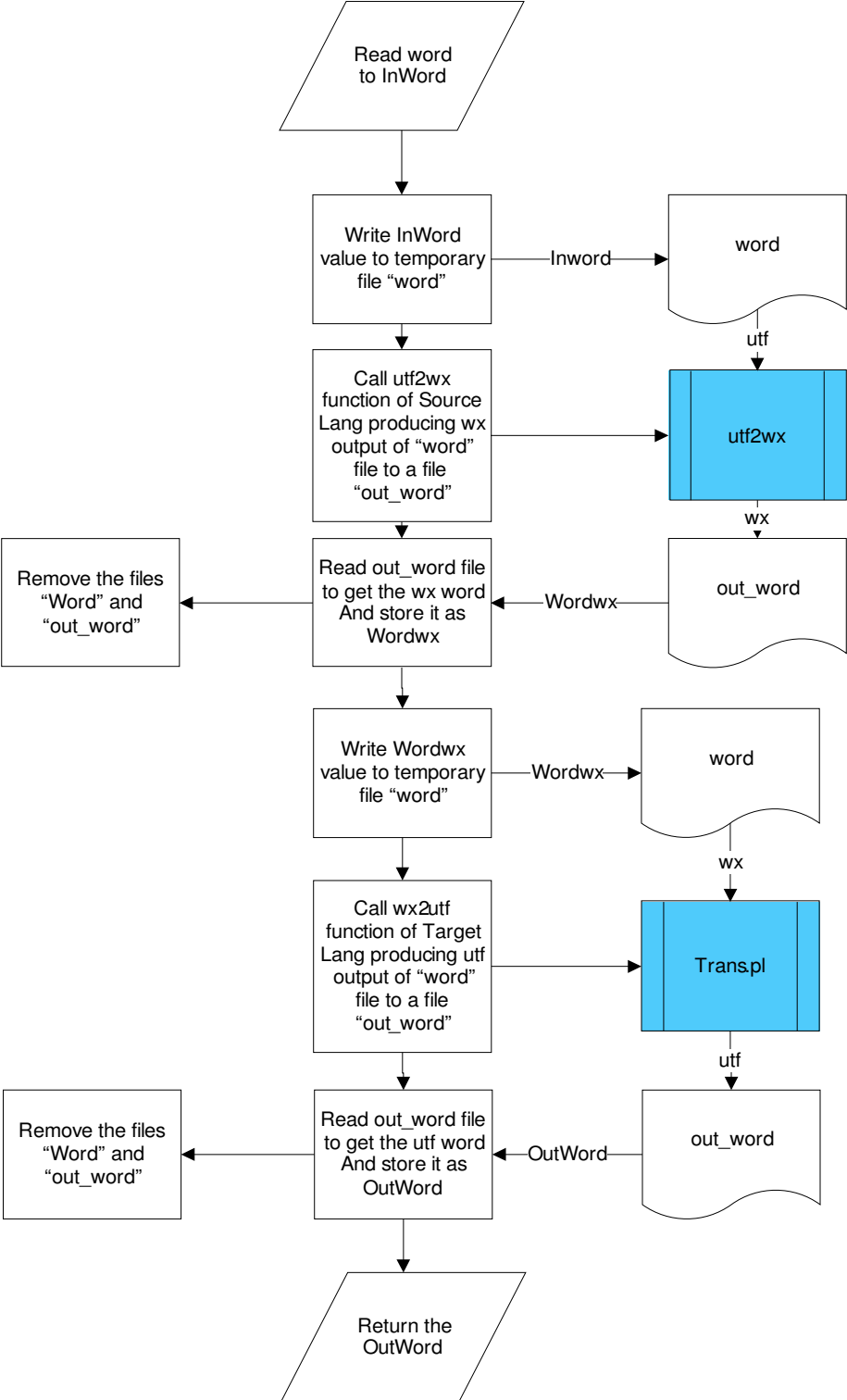


Fig.5- Word transliteration

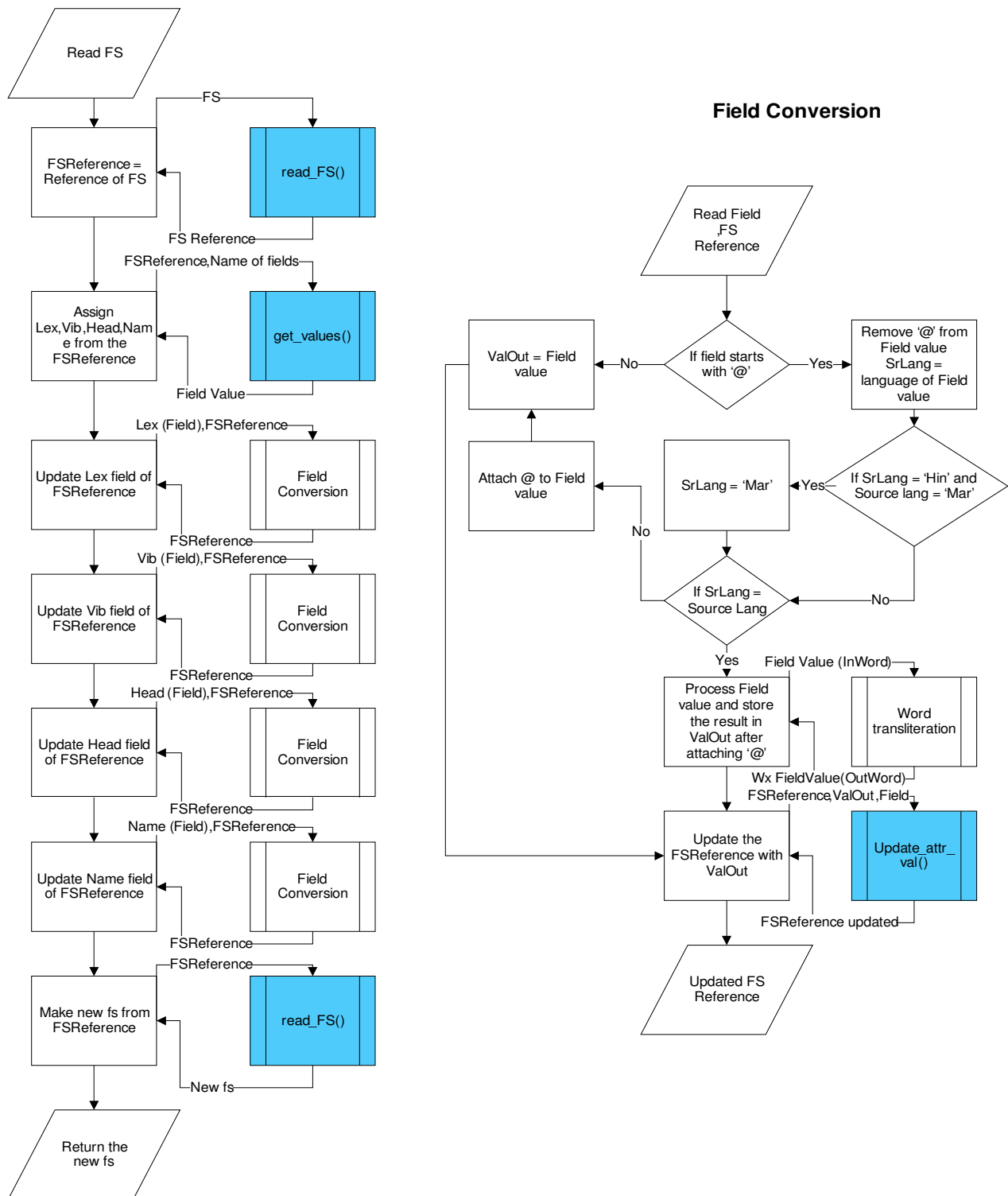


Fig.6- FS transliteration

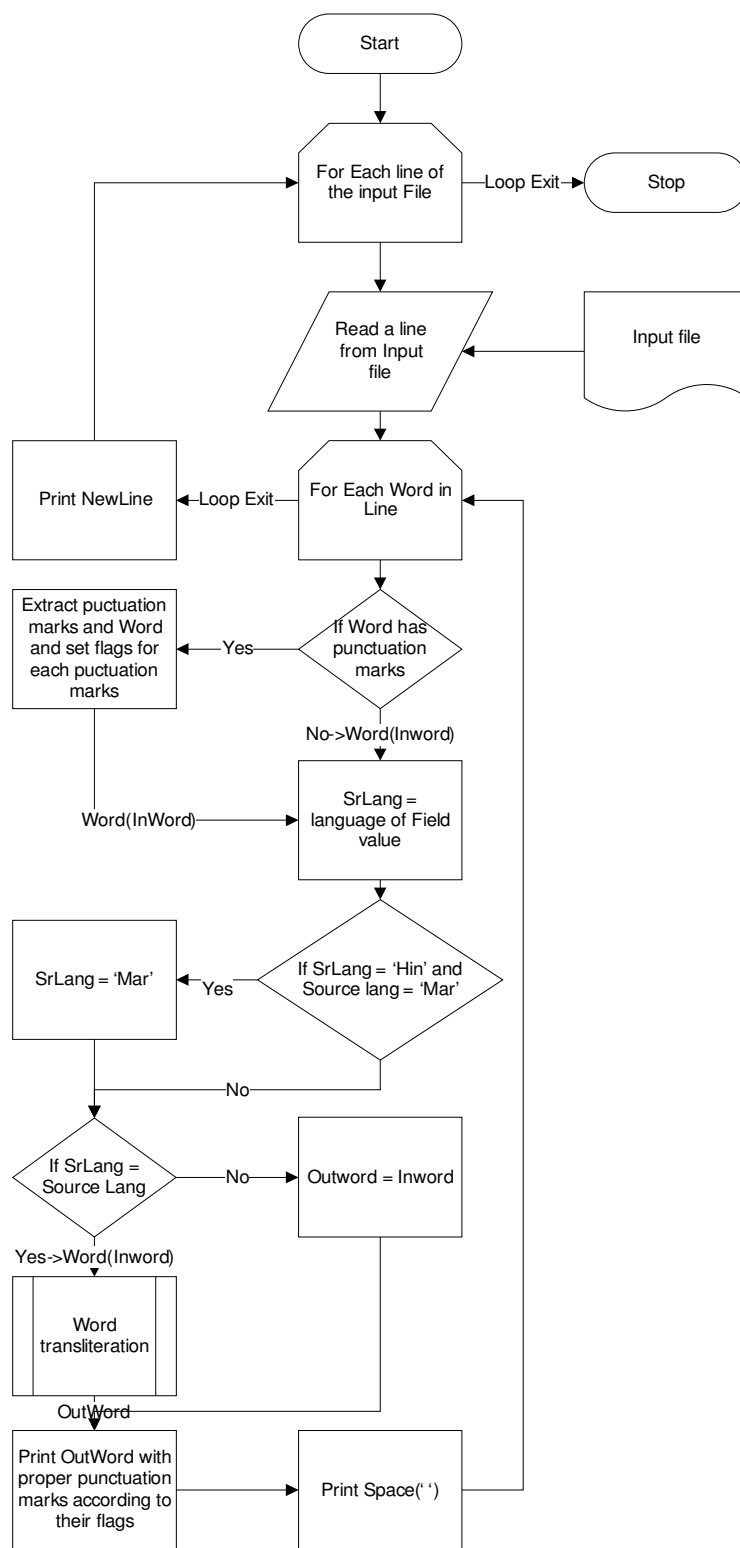


Fig.7- Text File transliteration

4. Process Description

4.1. Decide the type of Conversion and include the necessary resources

Based on the command line arguments a couple of parameters are identified

1. Languages of Conversion – Source Language and Destination Language
2. Type of Input File – SSF or UTF/Text file

Include the API and conversion module (based on the language selection)

4.2. Read the Input File

Read the input file line by line. Repeat the next four steps till the EOF.

4.3. Extract the Words to be converted

In case of SSF format extract the TKN and Feature structure using the API functions.

In case of text file read all words in the line separated by space.

4.4. Analyze and Convert the word if required

In case of Feature Structure, it is further divided into LEX, VIB, CAT, NAME and HEAD fields. Except the CAT field all the other fields are converted. In case of SSF Files only words prefixing with '@' is considered for conversion.

Remove and Store the extra characters (punctuation marks) in the word if it has any. Store the resulting word to a temporary file. Convert that file using the conversion module's mapping function and store the result in another temporary file. Read that file to get the converted word and remove the temporary files. Re-attach the extra characters (and '@' in case of SSF Files) in the converted word.

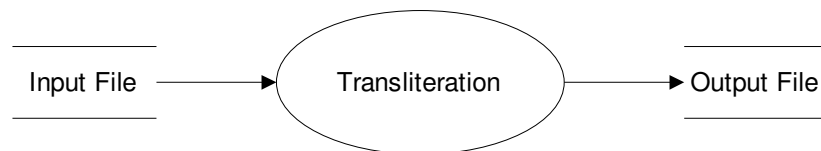
4.5. Reform the line.

Reform the SSF line or simple line based on the file type.

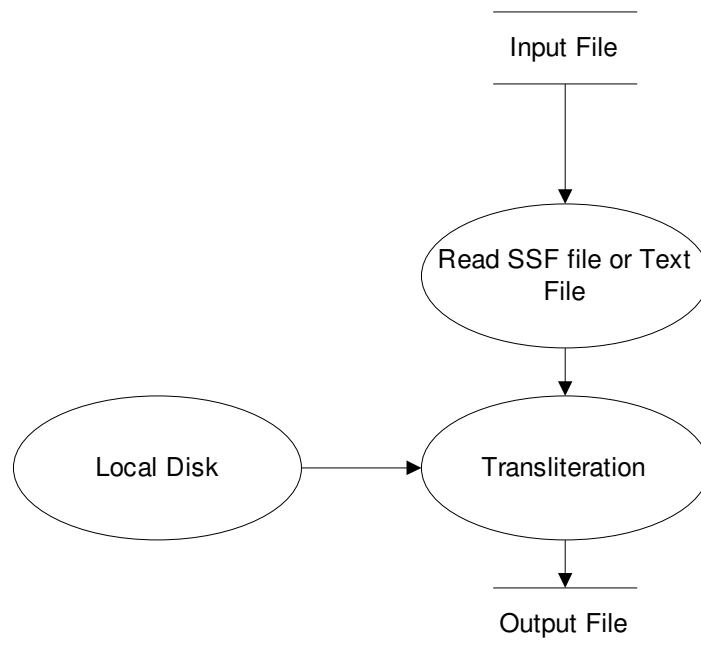
4.6. Print the resulting line to Output File

Print the converted line to the Standard Output.

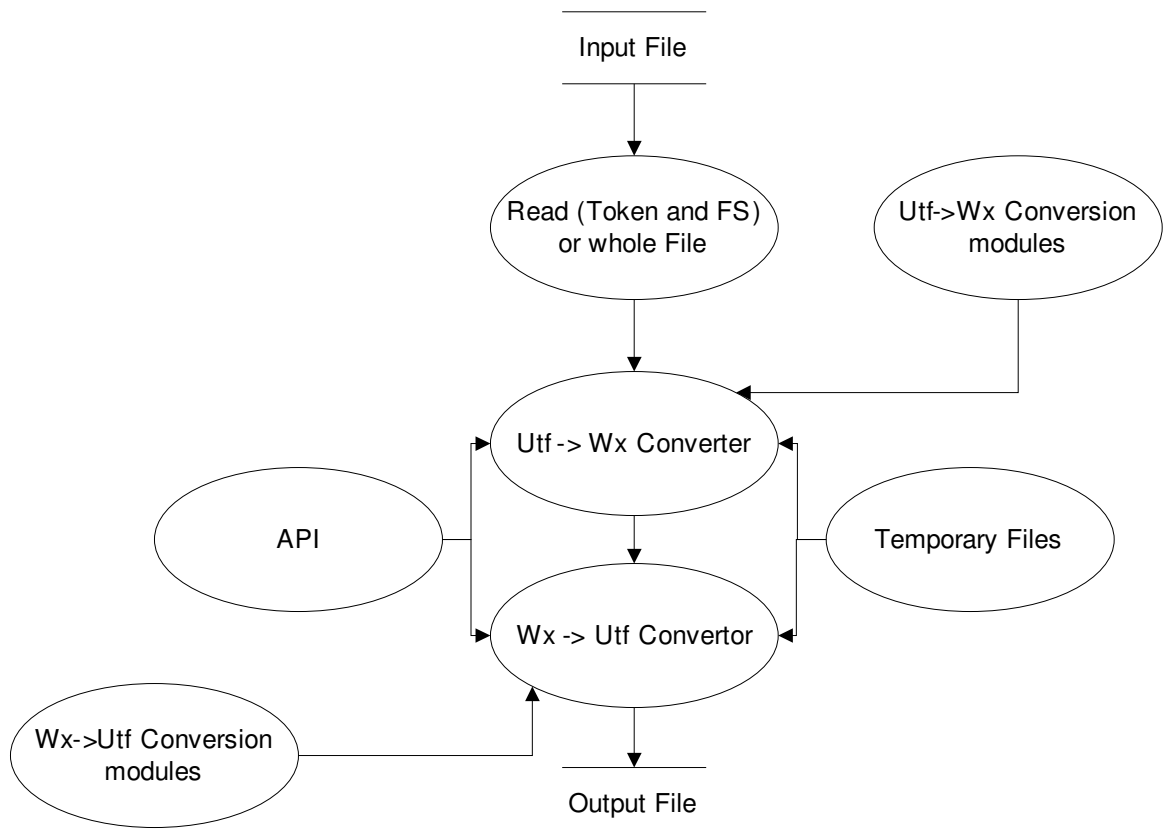
5. Data Flow Diagram



Level 0 DFD



Level 1 DFD

**Level 2 DFD**