Pre-processing Phase to Develop an Interface to Query Relational Databases in Punjabi Language: Query Normalization

[1]Harjit Singh, [2]Ashish Oberoi
[1]Assistant Professor, [2]Professor
[1]APS Neighbourhood Campus,
[1]Punjabi University Patiala, Punjab, India

Abstract: A natural language sentence needs some pre-processing before it is used for Natural Language Processing (NLP). The pre-processing step depends on the task to be performed on that natural language sentence. It is often called normalization of text. It is the step for preparing the sentence for further processing and so largely depends on the further processes. The first step for preparing the raw text for further processing involves cleaning the unwanted special characters from text. After cleaning, the next step is to replace some words or multiword expressions with alternative standard terms that are easy to process. In third step the sentence is split into tokens called tokenization. Last important step is stemming the words to remove any affixes attached to them. This paper presents a technique to normalize text for the development of an interface to query relational databases in Punjabi language. In this development the important words will be extracted from the query sentence and the sentence as a whole is not taken into consideration; instead some undesired words are ignored during further processing. These undesirable words are those words that may not be the information themselves but may be helpful in extracting the information from the text. This paper presents a normalization methodology that includes four steps that are Cleaning, Substituting, Tokenizing and Stemming.

IndexTerms **- Normalization, Natural Language Processing, Punjabi Language Processing, Cleaning, Substituting, Tokenization, Stemming.**

I. INTRODUCTION

Text processing is a task in research which belongs to Natural Language Processing (NLP) [1]. A natural language sentence needs some pre-processing before it is used for Natural Language Processing (NLP). The pre-processing of text is done to simplify the further processing. The pre-processing step depends on the task to be performed on that natural language sentence. For example, some pre-processing tasks remove un-desirable words from text while some others preserve them for further processing. It is often called normalization of text [2]. It is the step for preparing the sentence for further processing and so largely depends on the further processes. The normalization of text is done in several ways. Preparing the raw textfor further processing involves cleaning the unwanted special characters from text [3]. Any noiseother than proper words from text is removed while cleaning [4].Normalizing the text may also involve to replace some words or multiword expressions with alternative standard terms that are easy to process. In languages, same thing may be written in a number of ways making the text processing complicated. So, replacing these variations with standard alternative words may also be performed as per the need. For easy handling the sentence may be split into tokens called tokenization [5]. Rather than processing the whole sentence, it is divided into individual words for easy processing. Stemming may be done to remove any affixes attached to the words [6]. Affixes are attached to wordsto make the sentence meaningful and grammatically correct, but they are not suitable during text processing [7]. This paper presents a technique to normalize text for the development of an interface to query relational databases in Punjabi language. In this development named entities are the important words in a sentence and the sentence as a whole is not taken into consideration; instead some undesired words are ignored during further processing. These undesirable words are those words that may not be the information themselves but may be helpful in extracting the information from the text. This paper presents a normalization methodology that includes four steps that are Cleaning, Substituting, Tokenizing and Stemming.

II. RELATED WORK

Punjabi is a very low digital resources availability language. A research paper has been found related to the normalization of Punjabi words written by Gupta et al. [8]. The methodology discussed by the author is to generate a normalization database containing non-standard Punjabi words along with their equivalent standard Punjabi words. The author used a rule based algorithm to extract the words from 50 news articles. The presence of some characters such as ◌ਂ (Bindi), ◌ੱ (Adhak) and ◌ੁ (Bindi at Foot) is replaced with null and some Punjabi characters (ਹ,ਵ,ਰ)that are used in short form as Foot Characters(◌੍)which are replaced with their equivalent Characters in full form i.e. as ਹ,ਵ and ਰ. The purpose is to remove any variations from same words written in multiple ways.

III. METHODOLOGY

As already discussed that the process of normalization is greatly dependent on the further processing for which the normalization is done. The related work done by Gupta et al. [8] assists in removing any variations in similar words in Punjabi input text. But the pre-processing phase discussed in this paper is for the development of an interface to query relational databases in Punjabi language. So, the requirements are different and hence the methodology also differs.

This paper presents the methodology to make the Punjabi sentence ready to process for the development of an interface to query relational databases in Punjabi language. The whole methodology is divided into three categories as shown in Figure 1.:-

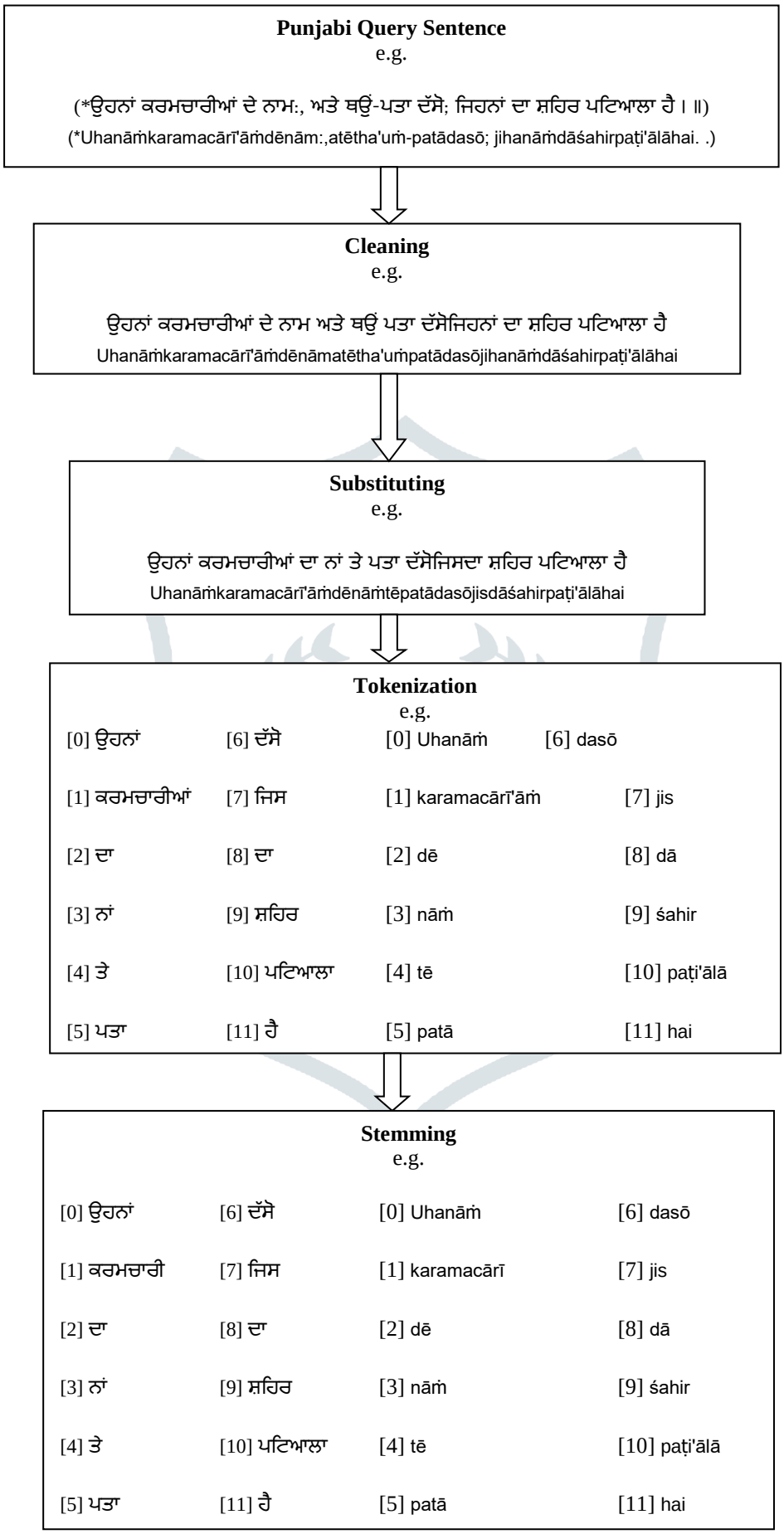

Figure 1: Steps of Pre-Processing Phase of Query Normalization

3.1 Cleaning

The first step is to clean the raw text to remove any unwanted characters from it. Preparing the raw text for further processing involves cleaning the unwanted special characters from text. Any noise other than proper words from text is removed while cleaning. For example, if the input Punjabi sentence is something like:

(*ਉਹਨਾਂ ਕਰਮਚਾਰੀਆਂ ਦੇ ਨਾਮ:, ਅਤੇ ਥਉਂ-ਪਤਾ ਦੱਸੋ; ਜਿਹਨਾਂ ਦਾ ਸ਼ਹਿਰ ਪਟਿਆਲਾ ਹੈ । ॥)

(*Uhanāṁkaramacārī'āṁdēnāma:,Atētha'uṁ-patādasō; jihanāṁdāśahirapaṭi'ālāhai. .)

ਉਹਨਾਂ, ਮਰੀਜ਼ਾਂ? ਦਾ {ਨਾਮ} {ਬਿਮਾਰੀਆਂ} ਅਤੇ ਥਉਂ-ਪਤਾ ਦੱਸੋ ਜਿਹਨਾਂ ਦੀ ਉਮਰ 50 ਨਾਲੋਂ ਜਿਆਦਾ ਹੈ

Uhanāṁ, marīzāṁ? dā {nām} {bimārī'āṁ} atētha'uṁ-patādasōjihanāṁdīumar 50 nālōṁji'ādāhai

In the above Punjabi sentence, there are many unwanted special characters that are not suitable to process the valid Punjabi words during further processing. So these unwanted characters are removed to clean the sentence as:

ਉਹਨਾਂ ਕਰਮਚਾਰੀਆਂ ਦੇ ਨਾਮ ਅਤੇ ਥਉਂ ਪਤਾ ਦੱਸੋ ਜਿਹਨਾਂ ਦਾ ਸ਼ਹਿਰ ਪਟਿਆਲਾ ਹੈ

Uhanāṁkaramacārī'āṁdēnāmaatētha'uṁpatādasōjihanāṁdāśahirapaṭi'ālāhai

ਉਹਨਾਂਮਰੀਜ਼ਾਂਦਾਨਾਮਬਿਮਾਰੀਆਂਅਤੇ ਥਉਂਪਤਾ ਦੱਸੋ ਜਿਹਨਾਂ ਦੀ ਉਮਰ 50 ਨਾਲੋਂ ਜਿਆਦਾ ਹੈ

Uhanāṁmarīzāṁdānāmbimārī'āṁatētha'uṁpatādasōjihanāṁdīumar 50 nālōṁji'ādāhai

3.2 Substituting

This is a major step of normalization in which some words are replaced with substitute words to make the sentence simpler. The words are replaced with their simpler forms using a database table of words and their synonyms. To create the database table, dataset is taken from IndoWordNet [9]. The database table contains all possible synonyms of simple words. The concept is to replace any synonym with its equivalent simple word to simplify the sentence. Some replacements that are not covered using this database table of synonyms, belongs to multiword expression such as 'ਥਉਂ ਪਤਾ'. To simplify these types of multiword expressions with their single word equivalents, another database table of words is created by manually analyzing text from various news articles, books and online text.

For example, the sentence obtained after cleaning from previous step is:

ਉਹਨਾਂ ਕਰਮਚਾਰੀਆਂ ਦੇ ਨਾਮ ਅਤੇ ਥਉਂ ਪਤਾ ਦੱਸੋ ਜਿਹਨਾਂ ਦਾ ਸ਼ਹਿਰ ਪਟਿਆਲਾ ਹੈ

Uhanāṁkaramacārī'āṁdēnāmaatētha'uṁpatādasōjihanāṁdāśahirapaṭi'ālāhai

ਉਹਨਾਂਮਰੀਜ਼ਾਂਦਾਨਾਮਬਿਮਾਰੀਆਂਅਤੇ ਥਉਂਪਤਾ ਦੱਸੋ ਜਿਹਨਾਂ ਦੀ ਉਮਰ 50 ਨਾਲੋਂ ਜਿਆਦਾ ਹੈ

Uhanāṁmarīzāṁdānāmbimārī'āṁatētha'uṁpatādasōjihanāṁdīumar 50 nālōṁji'ādāhai

After substituting the simple words by replacing their synonyms and replacing multiword expressions, the sentence obtained is:

ਉਹਨਾਂ ਕਰਮਚਾਰੀਆਂ ਦੇ ਨਾਂ ਤੇ ਪਤਾ ਦੱਸੋ ਜਿਸ ਦਾ ਸ਼ਹਿਰ ਪਟਿਆਲਾ ਹੈ

Uhanāṁkaramacārī'āṁdēnāṁtēpatādasōjisadāśahirapaṭi'ālāhai

ਉਹਨਾਂਮਰੀਜ਼ਾਂਦਾਨਾਂਬਿਮਾਰੀਆਂਤੇ ਪਤਾ ਦੱਸੋ ਜਿਸ ਦਾ ਉਮਰ 50 ਤੋਂ ਵੱਧ ਹੈ

Uhanāṁmarīzāṁdānāṁbimārī'āṁtēpatādasōjisdāumar 50 tōṁvadhhai

3.3 Tokenization

The complete sentence is almost simplified, but some words may contain affixes attached to them. To normalize these words, the sentence needs to be split into individual tokens, so that those tokens can further be processed one by one. Tokenization splits the whole sentence into individual tokens based on delimiter such as space in example sentence used in this paper [5]. The individual tokens are stored in an array to easily traverse the tokens back and forth. No change in any word is done at this step, only individual words are extracted from the whole sentence and stored in one dimensional array.

For example, the sentence obtained after substituting in previous step is:

ਉਹਨਾਂ ਕਰਮਚਾਰੀਆਂ ਦਾ ਨਾਂ ਤੇ ਪਤਾ ਦੱਸੋ ਜਿਸ ਦਾ ਸ਼ਹਿਰ ਪਟਿਆਲਾ ਹੈ

Uhanāṁkaramacārī'āṁdēnāṁtēpatādasōjisadāśahirapaṭi'ālāhai

In tokenization step, this sentence is split using space as delimiter. All other special characters have already been removed from the sentence in first step of cleaning. So, after tokenization the individual words (tokens) are:

[0] ਉਹਨਾਂ	[0] Uhanāṁ
[1] ਕਰਮਚਾਰੀਆਂ	[1] karamacārī'āṁ
[2] ਦਾ	[2] dē
[3] ਨਾਂ	[3] nāṁ
[4] ਤੇ	[4] tē
[5] ਪਤਾ	[5] patā
[6] ਦੱਸੋ	[6] dasō
[7] ਜਿਸ	[7] jisa
[8] ਦਾ	[8] dā
[9] ਸ਼ਹਿਰ	[9] śahira
[10] ਪਟਿਆਲਾ	[10] paṭi'ālā
[11] ਹੈ	[11] hai

Similarly, the following sentence obtained after substituting step is tokenized:

ਉਹਨਾਂਮਰੀਜ਼ਾਂਦਾਨਾਂਬਿਮਾਰੀਆਂਤੇ ਪਤਾ ਦੱਸੋ ਜਿਸ ਦਾ ਉਮਰ 50 ਤੋਂ ਵੱਧ ਹੈ

Uhanāṁmarīzāṁdānāṁbimārī'āṁtēpatādasōjisdāumar 50 tōṁvadhhai

[0] ਉਹਨਾਂ	[0] Uhanāṁ
[1] ਮਰੀਜ਼ਾਂ	[1] marīzāṁ
[2] ਦਾ	[2] dā
[3] ਨਾਂ	[3] nāṁ
[4] ਬਿਮਾਰੀਆਂ	[4] bimārī'āṁ
[5] ਤੇ	[5] tē
[6] ਪਤਾ	[6] patā
[7] ਦੱਸੋ	[7] dasō
[8] ਜਿਸ	[8] jis
[9] ਦਾ	[9] dā
[10] ਉਮਰ	[10] umar
[11] 50	[11] 50
[12] ਤੋਂ	[12] tōṁ
[13] ਵੱਧ	[13] vadh
[14] ਹੈ	[14] hai

3.4 Stemming

There may remain some words that have affixes attached to them. Stemming is the process of removing any affixes and use the root form of the word. A root form of a word is the base word which is appended with prefixes or suffixes to generate another word that is required to make the sentence grammatically correct. But these prefixes and suffixes create variations in the same word and these variations make text processing more complicated. For stemming these words, Punjabi language stemmer algorithm proposed by Gupta et al. [9] is used. This algorithm uses a set of rules to stem Punjabi Nouns and Proper names.

For example, the token ‘ਕਰਮਚਾਰੀਆਂ’ used in the example in previous step is stemmed to ‘ਕਰਮਚਾਰੀ’ and the final list of tokens is:

[0] ਉਹਨਾਂ	[0] Uhanāṁ
[1] ਕਰਮਚਾਰੀ	[1] karamacārī
[2] ਦਾ	[2] dē
[3] ਨਾਂ	[3] nāṁ
[4] ਤੇ	[4] tē

[5] ਪਤਾ	[5] patā
[6] ਦੱਸੋ	[6] dasō
[7] ਜਿਸ	[7] jisa
[8] ਦਾ	[8] dā
[9] ਸ਼ਹਿਰ	[9] śahira
[10] ਪਟਿਆਲਾ	[10] paṭi'ālā
[11] ਹੈ	[11] hai

Similarly, the words ਮਰੀਜ਼ਾਂ and ਬਿਮਾਰੀਆਂ are stemmed to ਮਰੀਜ਼ and ਬਿਮਾਰੀ from the previous second example:

[0] ਉਹਨਾਂ	[0] Uhanāṁ
[1] ਮਰੀਜ਼	[1] marīz
[2] ਦਾ	[2] dā
[3] ਨਾਂ	[3] nāṁ
[4] ਬਿਮਾਰੀ	[4] bimārī
[5] ਤੇ	[5] tē
[6] ਪਤਾ	[6] patā
[7] ਦੱਸੋ	[7] dasō
[8] ਜਿਸ	[8] jis
[9] ਦਾ	[9] dā
[10] ਉਮਰ	[10] umar
[11] 50	[11] 50
[12] ਤੋਂ	[12] tōṁ
[13] ਵੱਧ	[13] vadh
[14] ਹੈ	[14] hai

These tokens are now ready to be processed for extracting useful information for the development of an interface to query relational databases in Punjabi language.

IV. Implementation and Testing

Above methodology is implemented in c# using SQL Server database to store data. The implementation is done in Visual Studio 2010 as Windows Application. Some smaller modules were also developed during database preparation. Dataset was taken from IndoWordNet[10]as text files and converted to SQL Server format.Testing queries are prepared manually due to the absence of such queries as Punjabi language resource. The screen shots related to Implementation of Query Normalization process are shown in Figure 2 and Figure 3:

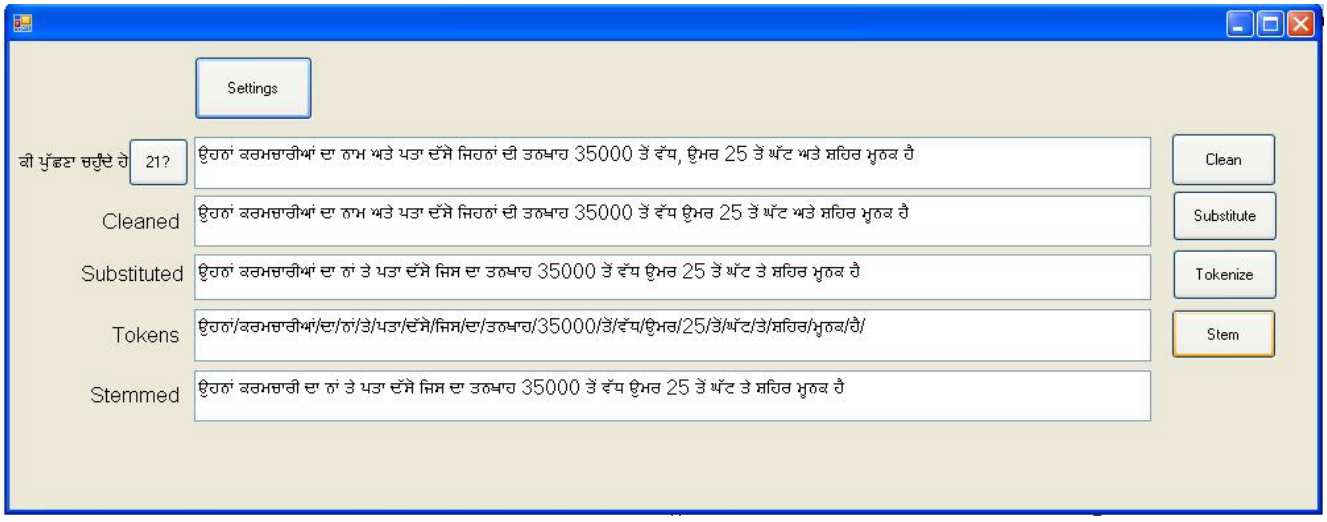

Figure 2: Implementation Interface to perform Query Normalization

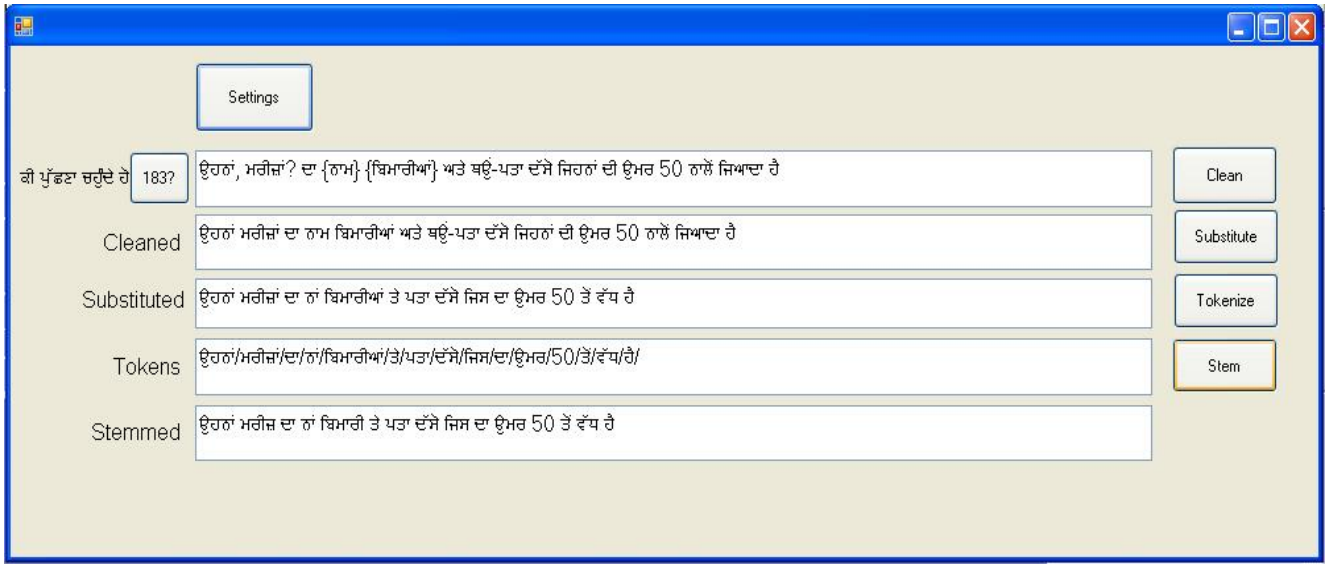

Figure 3: Implementation Interface to perform Query Normalization

The above implementation is tested with 510 Punjabi language sentences (queries) from different domains and with different sentence formats. To automate the Query Input, the queries are stored in a Unicode text file and are read one by one on pressing the button shown as 

in Figure 1. The button shows the query number being currently used as input. Some of the test sentences and the output of Query Normalization are shown in Table 1:

Table 1: Some of the Test Sentences and output of Query Normalization

Punjabi Language Sentence (Query)	**Output of Query Normalization**
ਉਹਨਾਂ;:,. ਵਿਦਿਆਰਥੀਆਂ/?.□}ਦਾਨਾਮ{□ਅਤੇਥਉ-ਪਤਾਦੱਸੋਜਿਹਨਾਂਦੇਅੰਕ50ਨਾਲੋਂਜਿਆਦਾਹਨ।	ਉਹਨਾਂਵਿਦਿਆਰਥੀਦਾਨਾਂ ਤੇਪਤਾਦੱਸੋਜਿਸਦਾਅੰਕ50ਤੋਂਵੱਧਹਨ
ਉਹਨਾਂ, ਮਰੀਜ਼ਾਂ? ਦਾ {ਨਾਮ} {ਬਿਮਾਰੀਆਂ} ਅਤੇ ਥਉ-ਪਤਾ ਦੱਸੋ ਜਿਹਨਾਂ ਦੀ ਉਮਰ 50 ਨਾਲੋਂ ਜਿਆਦਾ ਹੈ	ਉਹਨਾਂ ਮਰੀਜ਼ ਦਾ ਨਾਂ ਬਿਮਾਰੀ ਤੇ ਪਤਾ ਦੱਸੋ ਜਿਸ ਦਾ ਉਮਰ 50 ਤੋਂ ਵੱਧ ਹੈ
ਉਹਨਾਂਕਰਮਚਾਰੀਆਂਦਾਨਾਮਅਤੇਪਤਾਦੱਸੋਜਿਹੜੇ 35000 ਤੋਂਵੱਧ ਜਾਂ ਬਰਾਬਰਤਨਖਾਹਲੈਰਹੇਹਨ	ਉਹਨਾਂਕਰਮਚਾਰੀਦਾਨਾਂਤੇਪਤਾਦੱਸੋਜਿਸ 35000 ਤੋਂਵੱਧ ਜਾਂ ਬਰਾਬਰਤਨਖਾਹਲਿਆਹਨ
ਝੋਨੇ ਦੀ ਕਿਹੜੀ ਕਿਸਮ ਸਭ ਤੋਂ ਜਿਆਦਾ ਝਾੜ ਦਿੰਦੀ ਹੈ	ਝੋਨਾ ਦਾ ਕਿਹੜਾ ਕਿਸਮ ਸਭ ਤੋਂ ਵੱਧ ਝਾੜ ਦਿੰਦਾ ਹੈ
ਕਰਮਚਾਰੀਆਂਦਾਨਾਮ, ਪਤਾਅਤੇਸ਼ਹਿਰਜਿਹਨਾਂਦੀਵੱਧੋ-ਵੱਧਤਨਖਾਹ 4000 ਹੈ	ਕਰਮਚਾਰੀਦਾਨਾਂਪਤਾਤੇਸ਼ਹਿਰਜਿਸਦਾਘੱਟ ਜਾਂ ਬਰਾਬਰਤਨਖਾਹ 4000 ਹੈ
35000 ਤੋਂਵੱਧਤਨਖਾਹਲੈਣਵਾਲੇਕਰਮਚਾਰੀਆਂਦਾਨਾਮਤੇਪਤਾਕੀਹੈ	35000 ਤੋਂਵੱਧਤਨਖਾਹਲਿਆਵਾਲੇਕਰਮਚਾਰੀਦਾਨਾਂਤੇਪਤਾਕੀਹੈ
ਉਹਨਾਂ ਕਿਸਾਨਾਂ ਦਾ ਨਾਮ, ਜਿਲ੍ਹਾ ਅਤੇ ਜਮੀਨ ਕਿੰਨੀ ਹੈ ਜਿਹਨਾਂ ਦੀ ਆਮਦਨ 250000 ਤੋਂ ਘੱਟ ਹੈ?	ਉਹਨਾਂ ਕਿਸਾਨ ਦਾ ਨਾਂ ਜਿਲ੍ਹਾ ਤੇ ਜਮੀਨ ਕਿੰਨਾ ਹੈ ਜਿਸ ਦਾ ਆਮਦਨ 250000 ਤੋਂ ਘੱਟ ਹੈ
ਵਿਦਿਆਰਥੀਦਾਪਤਾਤੇਮੋਬਾਇਲਨੰਬਰਜਿਸਦਾਨਾਮਰਕੇਸ਼ਆ?	ਵਿਦਿਆਰਥੀਦਾਪਤਾਤੇਮੋਬਾਇਲਨੰਬਰਜਿਸਦਾਨਾਂਰਕੇਸ਼ ਹੈ
ਕਿੰਨੇਵਿਦਿਆਰਥੀਆਂਦੇਅੰਕ 50 ਤੋਂਵੱਧਨੇ?	ਕਿੰਨੇਵਿਦਿਆਰਥੀਦਾਅੰਕ 50 ਤੋਂਵੱਧਹੈ
ਉਹਨਾਂਕਰਮਚਾਰੀਆਂਦਾਨਾਮਅਤੇਪਤਾਦੱਸੋਜਿਹਨਾਂਦੀਤਨਖਾਹ 35000 ਤੋਂਵੱਧਨਹੀਂਜਾਂਉਮਰ 25 ਤੋਂਘੱਟਨਹੀਂਤੇਸ਼ਹਿਰਮੂਨਕਨਹੀਂਜਾਂਤਨਖਾਹ 25000 ਤੋਂਘੱਟਨਹੀਂਹੈ	ਉਹਨਾਂਕਰਮਚਾਰੀਦਾਨਾਂਤੇਪਤਾਦੱਸੋਜਿਸਦਾਤਨਖਾਹ 35000 ਤੋਂਵੱਧਨਹੀਂਜਾਂਉਮਰ 25 ਤੋਂਘੱਟਨਹੀਂਤੇਸ਼ਹਿਰਮੂਨਕਨਹੀਂਜਾਂਤਨਖਾਹ 25000 ਤੋਂਘੱਟਨਹੀਂਹੈ

A total of 510 query sentences from different domains are used to test the preprocessing phase and output of each step is stored in a database tablefor evaluationusing the following structure shown in Table 2.:-

Table 2: Structure of the database table used to store Inputs and Outputs of each step

Field Name	Description
Punjabi_Query	Inputted Punjabi Language Query Sentence.
Substituted_Query	Result of Substituting step of replacing complex words and multiword expressions with their single equivalent simple words.
Tokenized_Words	Result of Tokenization step of dividing Substituted_Query into individual tokens.
Stemmed_Words	Result of Stemming step of removing affixes from the generated tokens.

The output table is evaluated to count the number of correct outputs and the results of evaluation are as follows:

Total number of Query Sentences used in Test = 510
Correctly Normalized as required = 476
Not Normalized as required = 34

The pre-processing phase can be considered as efficient if it is able to normalize the inputted Punjabi Query sentence to the format that will be used for further processing.

The efficiency of Pre-processing phase is as follows:
Efficiency= (No. of Correctly Normalized Sentences/Total No. of Input Sentences) x 100= (476/510) x 100= 93.3%

Some of the query sentences used as input are not normalized as required. It is due to those words that were not available in substituting table or were not properly stemmed by the stemmer. By adding more possible words to the substituting table the required output can be obtained. Those words which are not properly stemmed by the stemmer can also be added to substituting table and will be covered in 'Substituting' step and no need toget those words stemmed by the stemmer.

The implementation of methodology including testing and test results are to show the efficiency of pre-processing phase. Since the pre-processing phase is particularly implemented to develop an interface to query relational databases in Punjabi language, so the efficiency cannot be compared with normalization methodologies implemented for other type of NLP applications.

V. Conclusion

A natural language query needsnormalization before it is used for Natural Language Processing (NLP). The pre-processing step depends on the task to be performed on that natural language sentence. It is often called normalization of text. At this step the sentence is prepared for further processing and so highly depends on the further processes. This paper presented a pre-processing phase of development of an interface to query relational databases in Punjabi language. The first step for preparing the raw text for further processing involved cleaning the unwanted special characters from text. After cleaning, the next step was to replace some words or multiword expressions with alternative standard terms that are easy to process. In third step the sentence was split into tokens called tokenization. Last important step was stemming the words to remove any affixes attached to them. The paper presented the methodology that included four steps that were Cleaning, Substituting, Tokenizing and Stemming. The paper also presented the implementation of methodology including testing and test result to show the efficiency of pre-processing phase. Since the pre-processing phase is particularly implemented to develop an interface to query relational databases in Punjabi language, so the efficiency cannot be compared with normalization methodologies implemented for other type of NLP applications.

References

[1] Joseph, Sethunya&Sedimo, Kutlwano&Kaniwa, Freeson&Hlomani, Hlomani&Letsholo, Keletso, "Natural Language Processing: A Review",Natural Language Processing: A Review, 6, 207-210, 2016
[2] https://en.wikipedia.org/wiki/Text_normalization
[3] https://www.nltk.org/book/ch03.html
[4] Subramaniam, L.V. & Roy, Shourya&Faruquie,Tanveer&Negi, Sumit. "A survey of types of text noise and techniques to handle noisy text",Proceedings of the Third Workshop on Analytics for Noisy Unstructured Text Data, AND 2009, Barcelona, Spain, July 23-24, 2009
[5] https://www.techopedia.com/definition/13698/tokenization
[6] Cristian Moral, Angélica de Antonio, Ricardo Imbert and Jaime, Ramírez, "A survey of stemming algorithms in information retrieval", Information Reseach, Vol. 19, No. 1, March, 2014
[7] https://en.wikipedia.org/wiki/Affix
[8] Vishal Gupta, "Automatic Normalization of Punjabi Words", International Journal of Engineering Trends and Technology (IJETT) – Volume 6 Issue 7- Dec 2013
[9] Vishal Gupta, Gurpreet Singh Lehal, "Punjabi Language Stemmer for nouns and proper names", Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011, Chiang Mai, Thailand, pp. 35–39, November 8, 2011
[10] http://www.cfilt.iitb.ac.in/indowordnet/