TEXT NORMALIZATION OF CODE MIX TEXT AND SENTIMENT ANALYSIS

Manali Vedak, *Student, PCE*, Neha Pai, *Student, PCE*, Swetha Ramaswamy, *Student, PCE*, and Dr. Sharvari Govilkar, Faculty, *PCE*

***Abstract*— Now a days, Emotions and opinions are the most influential & solely the driving factor of all social media content. It is observed that people can accentuate their sentiments in regional languages and thus the ways of writing with code-mixing and code-switching has unconsciously taken up in the social media as a trend. Users feel that their opinions are more emphasizing when written in code mix language. Hence the recent challenge is understanding the opinions in code-mixed content in social media. Such text is available in Romanized English format in Indian social media, which is the transliteration of Indian regional language in Romanized English, which demands text normalization to get further insights into the text. Thus this paper aims to use and process code-mixed language (Hinglish: English + Hindi) input and perform Text Normalization for the same.**

***Index Terms*— Language identification, normalization, syllabification, tokenization, transliteration**.

1. INTRODUCTION

Text normalization is the process of transforming text into a single canonical form that it might not have had before. Text normalization is a prerequisite for a variety of speech and language processing tasks and hence there is a demand for such systems. Normalizing text before storing or processing it allows for separation of concerns, since input is guaranteed to be consistent before operations are performed on it. Text normalization requires being aware of what type of text is to be normalized and how it is to be processed afterwards; there is no all-purpose normalization procedure. The rapid expansion of Internet use, electronic communication and user-oriented media such as social networking sites, blogs and micro blogging services has led to a rapid increase in the need to understand casual written English, which often does not conform to rules of spelling, grammar and punctuation. Text Normalization is important to get further insights into such text and for sentiment analysis.

Text Normalization is a discipline of Natural Language processing and a necessity for various speech and language processing tasks. In this bilingual speech community, there is a natural tendency of speakers to mix phrases and sentences during conversation, which has led to substantial code switching in Hindi and English language. We came across systems that use a statistical language independent approach for automatic detection of foreign words in mixed language. For Hindi-English a bilingual syntactic parser has been implemented. Conditional random field method has been used to identify the language of the words in mixed language documents. As the users on the social media are commonly using Abbreviation (short form) or SMS language to communicate, just dealing with language identification and transliteration cannot help us in understanding the text in social media sites.

Social Media is the frontier for opinions. Its anonymity provides the perfect ground for public voice. It also witnesses an active participation of people across the world and we get diverse outlooks over the same topic making it the best source for text mining. However, extensive usage of net lingo, i.e. use of various acronyms for common phrases and slangs and different forms of short hand words in place of normal words is a limitation. Also, users employ bilingual (or more than 2) languages users for convenience and ease of communication making it difficult to analyse such text. Hence this system will be designed to understand casual written English and code-mixed text, which often does not conform to rules of spelling, grammar and punctuation.

With the rapid development of social media, the irregularity of language poses a barrier to automated task. Posts are often highly ungrammatical, and filled with spelling errors, and resorted to selecting clusters of spelling variations manually. The interest in content of this type, both from researchers and corporations, shows a pressing need for effective text normalization. Natural Language processing tasks such as Machine Translation, Information Retrieval and Opinion Mining, require Text Normalization due to the irregularity of the language featured.

Our objective is to use concept of text normalization for code-mixed (Hindi-English) and impure social media text which would help to perform sentiment analysis. Social Media is the most budding platform with a great global outreach and an important source for text mining for social media analytics. Text Normalization can be used to make such text consistent. Hence, this paper presents a method for Text Normalization and how it can be used to treat and process social media content which in the above form as mentioned. The scope of the system would be to normalize code-mixed Hindi-English content in social media which is available in Romanized English format. This paper will also deal with impure social English consisting of Abbreviations (Short forms), Word play/ Intentional misspelling for verbal effect and Slang words (acronyms). The system converts impure social English to pure English, followed by tagging English and Hindi words. Hindi words are transliterated to Devanagari script for sentiment analysis based on lexicon approach.

The recipients of the system would be organizations which use social media monitoring such as public opinion, reviews and rating of the product which provide valuable information about emerging trends and what consumers and clients think about specific topics, brands or products.

2. LITERATURE SURVEY

Shashank Sharma, PYKL Srinivas and Rakesh Chandra Balabantaray's [1] proposed paper proves to be the base paper for further research as it presents various methods to normalize code - mix text available on social media, which is the transliteration of one language into another. In this paper, firstly, the language of code-mix text, which includes Phonetic Typing, Abbreviation (Short forms), Word play, intentionally misspelt words and Slang words is identified. This is followed by transliteration of these Romanized English language words which have a variety of spellings. The words are then judged for the sentiment of the statement to be positive or negative based on lexicon approach. An accuracy of 85% was achieved with the model.

Dinkar Sitaram, Savitha Murthy's [2] discusses an approach for sentiment analysis of the mixed language arising through the fusion of two languages: Hindi and English. In the mixed language, the resulting grammar usually alternates between the source language and deviates significantly from the grammatical structure of its source languages. This methodology incorporates the determination of the grammatical transition and the application of sentiment combination rules across languages to evolve the overall sentiment of a sentence in the mixed language. The model uses a recursive neural tensor network (RNTN) for sentiment classification. The model is trained with a training set consisting of 345 text samples in mixed language. Accuracy of model is low due to small training set data. Many important words that determine the overall sentiment of a sentence are not known to the classifier and hence assigned a neutral sentiment.

Ben King and Steven Abney's [3] uses a weakly-supervised learning method for identifying the languages of individual words in mixed language documents. A conditional random field model trained with generalized expectation criteria, a hidden Markov model (HMM) trained with expectation maximization (EM), and a logistic regression model trained with generalized expectation criteria is implemented. The paper concludes that CRF trained with GE is clearly the most accurate option among the methods examined. It also outperforms sentence-level language identification, which is too coarse to capture most of the shifts between languages. Also, named entities are not handled properly.

Heeryon Cho, Jong-Seok Lee and Songkuk Kim's [4] presents a method of improving lexicon-based review classification by merging multiple sentiment dictionaries, and selectively removing and switching the contents of merged dictionaries. First, the system compares the positive/negative book review classification performance of eight individual sentiment dictionaries. Then, selects the seven dictionaries with greater than 50% accuracy and combine their results using (1) averaging, (2) weighted-averaging, and (3) majority voting. It is shown that the combined dictionaries perform only slightly better than the best single dictionary (65.8%) achieving (1) 67.8%, (2) 67.7%, and (3) 68.3% respectively. To improve this, the approach combines seven dictionaries at a deeper level by merging the dictionary entry words and averaging the sentiment scores. Moreover, it leverages the skewed distribution of positive/negative threshold setting data to update the merged dictionary by selectively removing the dictionary entries that do not contribute to classification while switching the polarity of selected sentiment scores that hurts the classification performance. The revised dictionary achieves 80.9% accuracy and outperforms both the individual dictionaries and the shallow dictionary combinations in the book review classification task.

Subhash Chandra, Bibekananda Kundu and Sanjay Kumar Choudhury's [5] analyses the reasons of language mixing and its characteristics. The paper focuses on the mixed language called Benglish and Hinglish which are actually fusion of English with Bangla and Hindi language. A hybrid approach combining rule based and statistical methods has been proposed here. After manual introspection of the sentences of CMIC (Computer Mediated Informal Communication), the system extracts some linguistic patterns for detection of English words in Benglish text. The statistical model has two components viz. (1) Grapheme Language Model (GLM) and (2) Phoneme Language Model (PLM). When tested on 9152 Benglish sentences containing 13795 unique mixed words collected from CMIC, the proposed approach yielded an accuracy of 95.96% comparatively higher than 83.67% and 54.70% achieved by rule based and statistical approach respectively.

G.Vinodhini and R M Chandrashekaran's [6] covers a survey of various techniques and methods that are employed in performing sentiment analysis. It also covers the challenges that are faced while doing so. The paper covers the Machine Learning algorithms that are based on supervised learning. The other problem that it focuses on is semantic orientation which focuses on unsupervised learning. The role of negation and the various models that have been developed to handle the problem of negation are also discussed. The paper also discusses the feature based learning and the various algorithms. The paper also maps the function of various models based on certain metrics such as precision, recall and F-measure. The paper focuses on various methods that are already in existence and measures their working based on a few metrics.

Eleanor Clarka and Kenji Arakia's [7] discusses problems involved in automatically normalizing social media English. The paper describes an experiment which examines the efficacy of conventional spell checkers on casual English, and to what extent this could be improved with pre-processing with the given system. Results showed that average errors per sentence decreased substantially, from roughly 15% to less than 5% with use of spell checker.

Different spell checker has strengths and weaknesses with different types of errors.

Thomas Gottron and Nedim Lipka's [8] compares the performance of some typical approaches for language detection on very short, query-style texts. The results show an accuracy of more than 80% and for longer texts, accuracy values close to 100% were achieved. The approach focuses on methods based on character n-grams, for short n-grams. Next method studied was Naive Bayes which is a classical approach. Markov processes are used to determine the language of the text. The last approach is based in vector space of all possible n-grams.

R. Mahesh K. Sinha and Anil Thakur's [9] presents a mechanism for machine translation of Hinglish to pure (standard) Hindi and pure English forms. The approach uses a Hindi and English morphological analyser. The morphological analyser yields part of speech information for each of the words and marks the words that are unknown in Hindi and English respectively. The words that remain unknown, are marked for cross morphological analysis for plural noun forms. Complex code-mixed (CCM) Hinglish sentence is segmented into simple code-mixed Hindi (SCMH) and simple code-mixed English (SCME) parts. Next, the method isolates SCMH and SCME from CCM using heuristics and shallow parsing. Then it translates SCMH/SCME into pure Hindi language/pure English using FSM. The system fails in case of polysemous verbs, due to a very shallow grammatical analysis used in the process, the system is unable to resolve their meaning.

3. IMPLEMENTED ARCHITECTURE

In this chapter we would be discussing about the system architecture. The input to the system would be would be code-mixed (Hinglish), impure language text available from different social media domains like Facebook, Twitter and YouTube. The first unit is tokenization. This will be followed by conversion of impure SMS language to English. Language Identification will then be done to tag English and Hindi words. The next part is transliteration of Hindi to Devanagari script.

3.1 Input Documents

The input to the system would be code-mixed (Hinglish), impure language text. This text will be in Romanized English format which is transliteration of Hindi in Romanized English. The text would be from different social media domains like Facebook, Twitter and YouTube.

3.2 Tokenization

Process of converting sentence into a chain of words so that processing word by word can be easily performed. Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. We use white space character for tokenization.

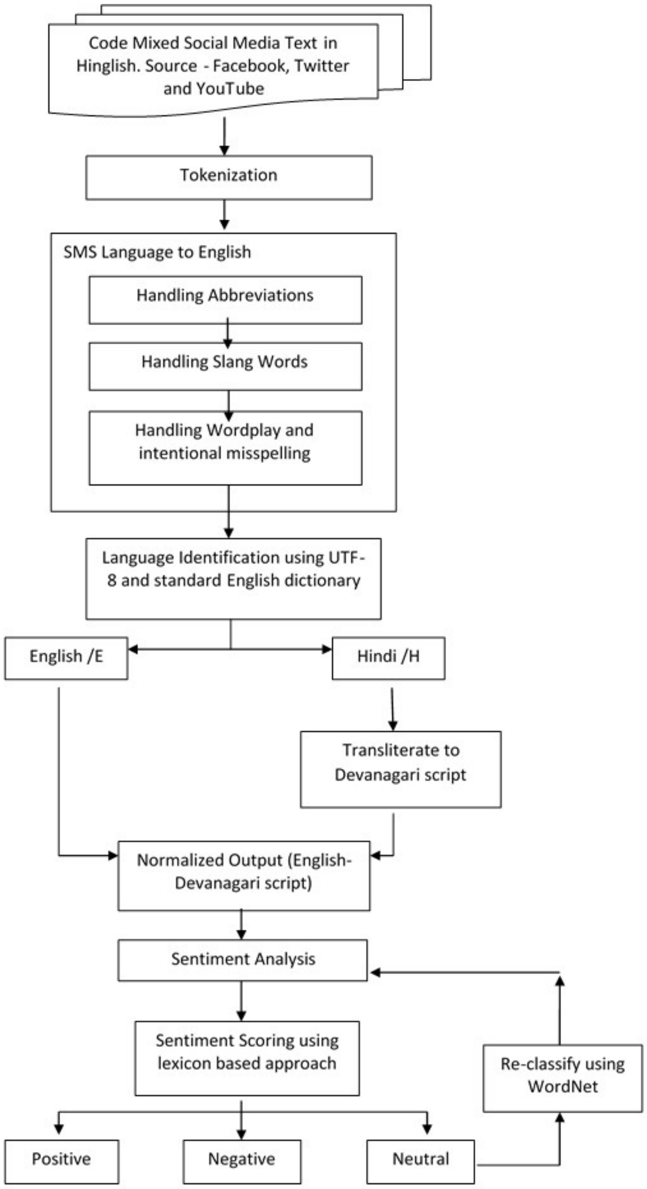
Code Mixed Social Media Text in Hinglish. Source - Facebook, Twitter and YouTube
Tokenization
SMS Language to English
Handling Abbreviations
Handling Slang Words
Handling Wordplay and intentional misspelling
Language Identification using UTF-8 and standard English dictionary
English /E
Hindi /H
Transliterate to Devanagari script
Normalized Output (English-Devanagari script)
Sentiment Analysis
Sentiment Scoring using lexicon based approach
Re-classify using WordNet
Positive
Negative
Neutral

Figure 3.1: Implemented Architecture

3.3 SMS Language to English

Input text contains different net lingo which is use of various acronyms for common phrases and slangs and different forms of short hand words in place of normal words. Before further processing this SMS language is transformed to pure English.

3.3.1 Handling Abbreviations

Now a days, on social media we find a lot of abbreviations (short forms) being used in English. We have mapped different kinds of shorthand notation spellings into one, for example: atb ("all the best") or lol ("laugh out loud"). A dictionary is used to handle and expand abbreviations.

Table 3.1: Sample of abbreviations dictionary.

aamof	as a matter of fact	gm	good morning
tc	take care	gtg	got to go
asap	as soon as possible	hand	have a nice day
atb	all the best	hhh	hip hooray
atm	at the moment	lol	laugh out loud
atst	at the same time	ruok	are you ok
atwd	agree that we disagree	aimb	as i mentioned before
awttw	a word to the wise	ttyl	talk to you later
bb	bye	tysm	thank you so much
gg	good name	wru	where are you

3.3.2 Handling Slang Words

The next most commonly used phenomenon on social media is the usage of slang words or acronyms, for example: 4get (“forget”). A slang dictionary with a number of words is used to train the system for correct language identification and transform slang words to pure English.

Algorithm for Handling Slang words:

Input: Tokens where abbreviations has been handled.

Output: Slang words converted to pure English.

A dictionary based approach is been utilized to handle slang words in the document. A dictionary of 250 words has been used for the comparison purposes. The algorithm is implemented as below given steps.

Steps:
1. A single token is read from token array.
2. Dictionary of slang words (“slang words.xls”) is opened.
3. Initialize a pointer variable to start of dictionary file.
4. Compare token and word from dictionary.
5. If match is found, replace slang word with its pure form.
6. If match is not found, increment pointer variable and go to step 4.
7. If end of file is reached, close dictionary file and stop.
8. Repeat the above process for all the tokens.

Table 3.2: Sample of slang dictionary.

2day	Today	gr8	Great
4u	for you	gud	Good
awsm	awesome	hv	Have
b	Be	hw	How
betn	between	k	Okay
bt	But	lyk	Like
d	The	r	Are
der	There	v	We
eg	Example	wat	What

3.3.3 Handling Wordplay

A lot of users often use creative spellings, which includes phonetic spelling and intentional misspelling for verbal effect e.g. that was soooooo big (“that was so big”). In this case, a simple algorithm may be used to identify the flaw and corrected it with the right English word.

Algorithm for Handling wordplay:

We use regular expressions (regex) to identify and remove multiple consecutive occurrences of characters which are typed intentionally.

```
1. re.sub(r'([a-zA-Z])\1{2,}', r'\1', inputs)
2. re.sub(r'([a-zA-Z])\1{2,}', r'\1\1', inputs)
```

3.4 Language Identification

Language Identification is an important unit and used to tag language of text as English (/E) or Hindi (/H). This is important for further processing of Romanized Hindi and eventually sentiment analysis. The process may start with identifying English followed by Hindi.

3.4.1 English Language Identification

The tokens are fed into an English Language Identifier. The tokens belonging to English language are tagged as /E at the end of each token. To identify the English tokens we use UTF-8 and a standard and vast English dictionary.

Unicode (or Universal Coded Character Set) Transformation Format means it uses 8-bit blocks to represent a character. UTF-8 is a variable-length byte encoding of Unicode, the character numbering system for all languages defined by Unicode. UTF-8 is the dominant character encoding for the World Wide Web and can support many languages and can accommodate pages and forms in any mixture of those languages. Its use also eliminates the need for server-side logic to individually determine the character encoding for each page served or each incoming form submission. This significantly reduces the complexity of dealing with a multilingual site or application and also allows many more languages to be mixed on a single page than any other choice of encoding.

Unicode has a total of 1,114,112 code points in the range 0(hex) to 10FFFF (hex).UTF-8 has the capacity of encoding all the code points defined in Unicode. Code points with lower numerical values (i.e., earlier code positions in the Unicode character set, which tend to occur more frequently) are encoded using fewer bytes. The encoding in UTF-8 has certain variations that are thoroughly followed. The first 128 characters of Unicode are encoded using a single octet with the same binary value as ASCII, making valid ASCII text valid UTF-8- encoded Unicode as well. And ASCII bytes do not occur when encoding non-ASCII code points into UTF-8, making UTF-8 safe to use within most programming and document languages that interpret certain ASCII characters in a special way.

Table 3.3: Samples of English dictionary.

a	do	heritage	`not	spread
adopt	excited	hurt	our	tell
all	feel	immortal	past	thank
anybody	field	in	pride	the
awesome	finally	like	richest	to
best	finals	match	rivalry	today
between	follow	match	science	was
cricket	from	miss	see	were
culture	good	much	sentiment	you
day	have	nice	so	yummy

Hindi Language Identification and Transliteration to Devanagari

The tokens which are not labelled as /E are considered to be in Hindi language which is written in Roman script and is tagged as /H.

Romanized Hindi words are transliterated to Devanagari script.

Machine transliteration refers to the process of automatic conversion of a word from one language to another without losing its phonological characteristics. Machine transliteration of English-Hindi is done using rule based approach. Some rules are constructed for syllabification. Syllabification is the process to extract or separate the syllable from the words.

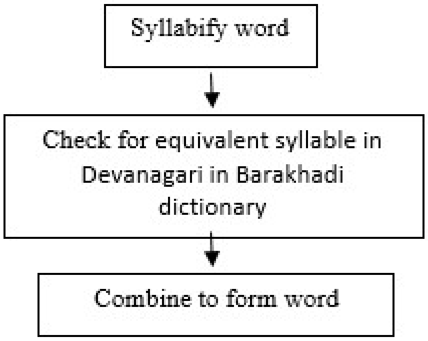
Syllabify word
Check for equivalent syllable in Devanagari in Barakhadi dictionary
Combine to form word

Figure 3.2: Transliteration Process

For constructing our rules, we are using syllabification approach. The syllable is a combination of vowel and consonant pairs such as V, VC, CV, VCC, CVC, CCVC and CVCC. Almost all the languages have VC, CV or CVC structures so we are using vowel and consonants as the basic phonification unit.

Some possible rules for syllabification are shown in table:

Table 3.4: Possible combinations of syllables

Syllable Structure	Example	Syllabified form (English)
V	Eka	[e][ka]
CV	Tarun	[ta][run]
VC	Angela	[an][ge][la]
CVC	Sidhima	[si][dhi][ma]
CCVC	Odisha	[o][di][sha]
VCC	Obhika	[o][bhi][ka]

V: Vowels, C: Consonants

If we have a word P then it will be syllabified in the form of {p1,p2,p3….pn} where p1,p2..pn are the individual syllables. Syllabification of English input is done using rule based approach for which the following algorithm is used.

Algorithm for syllable extraction

1. Enter input string in English.
2. Identify Vowels and Consonants.
3. Identify Vowel-Consonant combination and consider it as one syllable.
4. Identify Consonants followed by Vowels and consider them as separate syllables.
5. Identify Vowels followed by two continuous Consonants as separate syllable.
6. Consider Vowel surrounded by two Consonants as separate syllable.
7. Transliterate each syllable into Hindi.

Table 3.5: Samples of English-To-Hindi-Conversion list.

a	अ	ee	ई	ki	कि
aa	आ	ei	ऐ	me	मे
cha	च	ga	ग	o	ओ
chha	छ	ha	ह	oo	ऊ
e	ए	i	इ	ra	र
ea	ए	ja	ज	sa	स

4. Result Analysis

The quality of our system can be obtained by calculating the efficiency of the system. There is no particular method to calculate such an efficiency. Thus we have calculated the percentage of correct outputs for all the different datasets considered for testing.

$$\text{Efficiency} = \frac{\text{Total No of Correct words in the output}}{\text{Total No of words in input}}\%$$

Table 4.1: Sample test cases with output

Input Sentence	Output
2day's match was sooooo awsm. Hamesha ki tarah sab were excited to see match betn gr8 rivalry in cricket. Sachin aur Sehwag ke jodine ne to kamal kar diya	today/E s/E match/E was/E so/E awesome/E हमेशा/H की/H तरह/H सब/H were/E excited/E to/E see/E match/E between/E great/E rivalry/E in/E cricket/E साचिन/H और/H सेह्वाग/H के/H **जोदीने /H any/E** to/E **कमल/H** कर/H दीया/H
Finally v r in finals. ATB Indian Team ko. Feeling very proud 2 b Indian.	finally/E we/E are/E in/E finals/E all/E the/E best/E indian/E team/E को/H feeling/E very/E proud/E to/E be/E indian/E
GM all Ab India ke cultural heritage ke baare me kya batavu. India has d richest culture in d world.	good/E morning/E all/E ab/E india/E के/H cultural/E heritage/E के/H बारे/H me/E क्या/H बतावु/H india/E has/E the/E richest/E culture/E in/E the/E world/E
Aamof, Mohini Attam ne India ko world level pe represent kiya hai. A.R. Rehman ek aur zinda udaharan hai.	as/E a/E matter/E of/E fact/E मोहिनी/H अत्ताम/H **any/E** india/E को/H world/E level/E पे/H represent/E किया/H हैं/H a/E **are/E** रहमान/H एक/H और/H ज़िन्दा/H **उदहारं/H** हैं/H
Iirc he has won oscars 4 India in music. Not just art lekin science ke field me bhi bohot insaan hai who hv brought pride to our nation.	if/E i/E remember/E correctly/E he/E has/E won/E oscars/E for/E india/E in/E music/E not/E just/E art/E लेकिन/H science/E के/H field/E me/E भी/H बहुत/H इन्सान/H हैं/H who/E have/E brought/E pride/E to/E our/E nation/E
Hamare culture ko poore duniya ne adopt karne ki koshish ki hai. Tysm HAND.	हमारे/H culture/E को/H पूरे/H दुनियाँ/H **any/E** adopt/E करने/H की/H कोशिश/H की/H हैं/H thank/E you/E so/E much/E have/E a/E nice/E day/E
Heyyyyyyy……I am soooooo happyyyyy. I gt d jobbb! D intrvw ws soooooo gooood! D off s sooooo hugggeee!	hey/E i/E am/E so/E happy/E i/E got/E the/E job/E the/E interview/E was/E so/E **god/E** the/E **off/E s/E** so/E huge/E
ND d ppll! OMG! Dey r d nicessst ppl on earthhhh! U Free? Wanna celeb 2nigt!!! Lets gt dnnr!	and/E the/E **nullलल/H** oh/E my/E god/E they/E are/E the/E nicest/E people/E on/E earth/E you/E free/E want/E to/E celebrate/E **nullि◌null/H** lets/E **got/E** dinner/E
My Treat! Sooooooo! D buk I read ws sooooo gooooddd ☹ I wannna cryyyy nd laughhh nd smillle at d sime time!!!!!!	my/E treat/E so/E the/E book/E i/E read/E was/E so/E **god/E** i/E wanna/E cry/E and/E laugh/E and/E smile/E at/E the/E **sime/E** time/E
Omg ommmmmmmggggg! U hve to see dis pic! D place is in Russia! Bt itssss sooooo beautifulll! Nd the treessss nd animals r soooo cuuuttttteeeeeee!	oh/E my/E god/E oh/E my/E god/E oh/E my/E god/E you/E have/E to/E see/E this/E picture/E the/E place/E is/E in/E russia/E but/E its/E so/E beautiful/E and/E the/E trees/E and/E animals/E are/E so/E cute/E

A bigger test case has been demonstrated. This test case has a total of 70 sentences. The diagram shows the graph with the correct outputs and the incorrect outputs for the mixed test case. The total correct output words generated was 1076 and the incorrect outputs was 56. The efficiency of the system comes around 95%.

5. CONCLUSION

The proposed system mainly tackles the three areas as specified: Slang words, Abbreviations and intentional misspelling and wordplay. The system performs text identification and normalization. The system would also be able to handle the Hinglish (Hindi+English) and Minglish (Marathi+English) words as and when they are used as input. The system uses UTF-8 for identification of the Hindi part of the text.

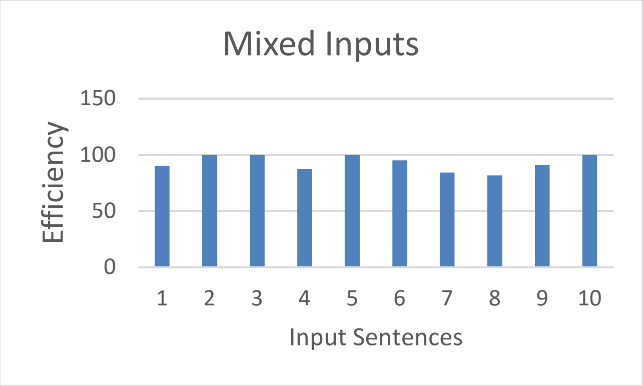

Figure 4.1: Efficiency graph

Acknowledgment

We remain immensely obliged to Dr. Sharvari Govilkar for providing us with the idea of this project topic, for her invaluable support in garnering resources for us either by way of information or computers and also her guidance and supervision which made this project happen. We would like to thank our college, Pillai College of Engineering, New Panvel and our principal Dr. R.I.K.Moorthy and the H.O.D. of Computer Department Dr. Madhumita Chatterjee and Prof. Rupali Nikhare and Prof. Varunaakshi Bhojane for their invaluable support. Also, we would like to say that it has indeed been a fulfilling experience for working out this project topic.

REFERENCES

[1] Shashank Sharma, PYKL Srinivas, Rakesh Chandra Balabantaray, 2015, Text Normalization of Code mix and Sentiment analysis.

[2] Dinkar Sitaram, Savitha Murthy, Debraj rai, 2015, Sentiment analysis of mixed language employing Hindi-English code switching.

[3] R. Mahesh K. Sinha, Anil Thakur, 2005, Machine Translation of Bi-lingual Hindi-English (Hinglish) Text.

[4] Ben King, Steven Abney, 2013 Labelling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods

[5] Heeryon Cho, Jong-Seok Lee, Songkuk Kim, 2013 Enhancing lexicon-based review classification by merging and revising sentiment dictionaries

[6] Subhash Chandra, Bibekananda Kundu and Sanjay Kumar Choudhury, 2013, Hunting Elusive English in Hinglish and Benglish text: Unfolding challenges and Remedies.

[7] Bing Liu, 2012, Sentiment Analysis and Opinion Mining.

[8] Eleanor Clark, Kenji Araki, 2011, Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English
[9] Thomas Gottron, Nedim Lipka2, 2011, A Comparison of Language Identification Approaches on Short, Query-Style Texts