Lexical normalization of roman Urdu text

Zareen Sharf, Dr Saif Ur Rahman
PhD Scholar SZABIST Associate Professor SZABIST

Summary
Social media text usually comprises of short length messages, which typically contain a high percentage of abbreviations, typos, phonetic substitutions and other informal ways of writing. The inconsistent manner of text representation poses challenges in performing Natural Language Processing and other forms of analysis on the available data. Therefore, to overcome these issues the text requires to be normalized for effective processing and analysis.
In this work, we have performed a comparative study of how social media text in different languages like Chinese, Arabic, Japanese, Polish, Bangla, Dutch and Roman Urdu has been normalized to achieve consistency. We have discussed in detail the normalization methods proposed, their success rate and their shortcomings. Based on our analysis we have also proposed a model for achieving lexical normalization of text in Roman Urdu.

Index Terms
Normalization, Standardization, Transliteration, Roman Urdu.

1. Introduction

Social Media has played an extremely pivotal role in projecting the social and political dynamics of every society around the globe. Short and quick text messages have proven to be a very strong and popular means of communication in recent times. Social media users tend to become more experimental in their manner of written communication thus creating several variations of similar content. Since social media perforation in our lives has been rising consistently therefore it becomes imperative to develop processes and techniques suitable for retrieving relevant social content in an effective manner. There are many reasons why analytically processing informal text, such as Twitter posts or text messages, could be useful. For example, during the January 2010 earthquake in Haiti, volunteers translated Creole text messages that survivors sent to English speaking relief workers. Machine translation could supplement or replace such crowdsourcing efforts in the future. However, working with user data presents several challenges. Messages may have non-standard spellings and abbreviations, which need to be normalized into standard language.
One of the most important and complex issue encountered while performing analysis on social media text is presence of numerous abbreviations, typos emoticons and other informal ways of writing. These issues produce inconsistency in the representation of comments made by different users thus making the analysis of text an error prone and challenging task. To deal with this problem lexical normalization techniques need to be developed. This has been an active research area in recent times and many models for different languages have been proposed to overcome this issue (Choi and Kim 2014) Urdu is the national language of Pakistan. It is written in Perso-Arabic script. However, in social media and short text messages (SMS), a substantial proportion of Urdu speakers use roman script for writing, called Roman Urdu. Roman Urdu lacks standard lexicon and usually many spelling variations exist for a given word, e.g., the word zindagi (life) is also written as zindagee, zindagy, zaindagee and zndagi. Specifically, the following normalization issues arise:

(1) Differently spelled words (see example above)
(2) Identically spelled words that are lexically different (e.g., ‘bahar’ can be used for both [outside] and [spring]
(3) Spellings that match words in English (e.g., ‘had’ in Roman Urdu [meaning limit] for the English word ‘had’).

These inconsistencies cause a problem of data sparsity in basic natural language processing tasks such as Urdu word segmentation, part of speech tagging, spell checking, machine translation, etc.
In this paper, we present a comparative analysis on how lexical normalization was achieved for languages like Arabic, Japanese, Chinese, Polish, Dutch, Finnish, Bangla, English, Croatian, Vietnamese and Roman Urdu as it is more relevant and specific to our society.

2. Motivation & Contribution

Social media sites are highly attractive for extraction of information and text mining because of the huge amount of real-time data they generate on daily basis. However, the quality of content varies significantly ranging from professional newswire-like text to pointless strings. Presence of typos, abbreviations, phonetic variants, structure less grammatical phrases, and emoticons make it harder for the natural language processing tools to process data accurately and effectively. It has been observed that lexical parsers tend to produce incorrect interpretation of text which is processed in its original form as extracted from the source but if the same text is normalized to

Manuscript received December 5, 2017
Manuscript revised December 20, 2017

produce a more standard version then the quality of analysis improves significantly.
The lexical normalization task is nevertheless a challenging task. Most of the techniques proposed and models developed are extremely domain-specific and tend to produce accuracy under strict constraints. The motivation of this study was therefore to perform a comparative analysis of how the task of standardizing unconventional text has been achieved for different languages. This study helped us to understand and evaluate the techniques and models developed by other researchers. Their accuracy rate as well as shortcomings/limitations gave us valuable insight on the future enhancement required to improve the existing models.
Our Contribution is to highlight the strength and weaknesses of existing models and based on our findings propose a Lexical Normalization Model to perform standardization of text in Roman Urdu.

3. Lexical Normalization

User generated content (UGC) appearing in the form of text messages and comments on social media sites like Twitter, Facebook, blogs and discussion forums varies widely in content and composition. Nonstandard words utilization and informal manner of writing create numerous issues for the text analysis models and tools to produce accurate results. For example, ‘kinda’ for ‘kind of’ or, ‘toook’ for ‘took’. Natural Language Processing on UGC thus requires development of methods and techniques for normalizing content prior to its submission as input to the NLP tools. (Grzegorz 2014)

A Wikipedia Definition of Text Normalization is:
Text normalization is the process of transforming text into a single canonical form that it might not have had before. 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.

4. Literature Review

An Urdu Romanization scheme named “Uddin and Begum Urdu-Hindustani Romanization” was proposed by Fasih Uddin and Quader Unissa Begum (1992) and was accepted as an international standard for Romanizing Urdu. Uddin and Begum modified and modernized Gilchrist's system by introducing a scheme that provided a one to one mapping for Urdu and Hindi characters. Also, diacritics indicated vowel phonics, whereas in the Gilchrist system the reader was required to infer vowel pronunciation from context. To facilitate Urdu-Hindustani Romanization in a much wider range of computer software, Uddin and Begum limited their character set to the common ASCII standard.

Ahmed (2009) proposed a technique for mapping Urdu characters to Roman Urdu by assigning similar sounding letters in Urdu to a single similar sounding letter in English thus creating clusters in some cases. Various other rules to handle consonants, vowels and other alphabet forms were formulated from this set. The procedure used for transliteration began with encoding a list of 5000 most frequently used Urdu words into Roman Urdu by using the rules defined according to the scheme mentioned earlier. It then takes a Roman Urdu word as input and compares it to the words in the encoded form to determine the correct spelling of the input word.

Nahir (2003) worked extensively on codification of Hebrew language specifically focusing on bridging the gap that existed in Hebrew because of the limited vocabulary set. This task was accomplished by retrieving old words and roots, creating new words from old words and roots, loan-translations, combining existing words, blending, filling in pattern with root "fillers", borrowing words and roots, etc. The Hebrew language has matured quite significantly after these implementations and is now considered a well-established modern language.

Liyew (2002) performed lexical standardization of Oromo a popular Language form in Ethiopia. The process was divided into four phases: (1) selection (2) codification (3) elaboration and (4) implementation. Base dialects were first collected from archived documents, mass media, and accent of the speakers and existing status of the dialect. The criteria for the formulation of a standard mainly involved parameters like number of speakers, word frequency, uniqueness, efficiency, economy and semantic acceptance. Methods used for extending the lexical capabilities were blending, semantic extension, compounding, derivation and borrowing. The proposed model produced a significant success rate of achieving the desired task.

Malik and Abbas (2008) discussed the UIT (Universal Intermediate Transcription) scheme which is an encoding scheme using ASCII range 32-126 for representation of characters in different languages like Hindi, Urdu, Punjabi, etc. UIT is an extension of SAMPA (Speech Assessment Methods Phonetic Alphabet) broadly used for encoding the IPA (International Phonetic Alphabet) into ASCII. A model named HUMT was developed that used finite-state transducers for encoding natural languages into ASCII. The model was validated to produce an accuracy of 97.5

percent when applied on the Hindi-Urdu corpora containing 412,249 words.

5. Comparative analysis of normalization techniques

The research work has been broadly divided into two parts. The first part focuses on a comparative analysis of how text/lexical normalization was achieved for different languages. A summarized comparison is shown in Table 1. Based on this analysis the second phase proposes a theoretical model for achieving normalization for data in Roman Urdu.

Table 1. Comparative Analysis of Normalization Techniques for Different Languages

S#	Language	Technique Used	Accuracy	Limitation
1	English	(Han and Bo 2011) NSW are detected by first separating IV words from OOV words and then the OOV words are compared to a list of domain specific words to replace them with the closest match.	The model achieved a higher level of accuracy as compared to state-of art models for Recall, Precision and F-score Test.	DSM Model is highly constraint because of its domain specific nature. Detection of NSW and replacement of OOV words using DSM model are independent operations and combining these two to produce a single model is identified as future work.
		(Han and Bo 2013) Lexical Normalization of Short Text Messages is done by generation of a confusion set where possible candidates for normalization of a word are defined. After identifying ill-formed words, they are compared with the possible candidates from the confusion set and the best possible candidate based on morphophonemic variation is selected for normalization.	Significantly better normalization results were achieved by combining dictionary lookup, word similarity and context support techniques to the proposed model.	Ill-formed word detection classifier needs to be further improved for producing better accuracy. This can be achieved by introducing an OOV word list.
		(Supranovich and Dmitry, 2015) The technique proposed comprises of two mechanisms. First uses CRF approach to identify candidates suitable for normalizing a word and the second phase addresses normalization of words that do not have candidates defined from the lexicon using DYM(Did-You-Mean) model. This model is a variant of SVM model and helps to normalize words that are not found in the dictionary.	The F-measure test showed better results for the proposed model as compared to baseline models.	DYM tool requires fine tuning for better results. Also, enhancement of lexicons to add more words can improve normalization process. Filtering of non-English words as a pre-processing step is also suggested for achieving better performance.
2	Japanese	(Kaji and Nobuhiro 2011) A normalization dictionary was created by merging a tag dictionary with the standard Japanese lexicon named JUMAN. The normalization dictionary was further enhanced by adding normal forms and normal POS of ill-formed words determined by hand-crafted rules. The model not only achieves normalization but also performs word segmentation and POS tagging using a lattice based approach.	The proposed model delivered better performance as compared to baseline model in terms of Precision, Recall and F1 score for POS tagging and Word Segmentation.	The model does not perform very well with normalization of misspelled words and need to be further enhanced for achieving better accuracy.
3	Chinese	(Wang and Aobo 2013) A two-step general classification model for word normalization was developed. In the first step, potential formal candidates for the word to be normalized were generated using Google 1T Corpus. In the second step a binary classifier was used for identifying the most suitable candidate for substitution. The classifier used both rule-based and statistical features for achieving this task.	The developed model produced better results as compared to SVM and LR models in terms of Recall, Precision and F1-Score.	The three major channels identified for ill formed words were phonetic substitutions, Abbreviations, and Paraphrasing. The classifier designed is based heavily on these channels. Therefore, the performance of the classifier can be significantly enhanced with better channel knowledge.
4	Bangla	(Alam and Firoj 2008) The proposed model is based on tokenizing and assignment of input data to belong to one of the twelve predefined semiotic classes. Verbalization of NSW words were achieved using lexicon based approach.	Accuracy Rate for predicted semiotic classes was above 60 percent.	POS tagging for verbs was identified as a limitation as the model did not deliver accurate results for this activity.
5	Dutch	(Schulz and Sarah 2016) A multi-modular system for text normalization was developed. A combination of token based and context based modules was used to achieve better results. A rule-based tokenizer was used for tokenization. Character flooding was eliminated by restricting the number of repetitions of a character to at	The developed model showed an accuracy rate of more than 70 percent for both normalization and POS tagging as compared to baseline models.	Named Entity recognition did not produce desirable results and is identified as a limitation of the developed model.

		the most two times with the exception for the vowel 'e'. The suggestion layer is divided into two parts. One of token-based modules that normalize words that are NSW. The other is context-based modules that deal with problems of phonology, spellings and abbreviations and normalize such terms taking their context into consideration.		
6	Finnish	(Korenius and Tuomo 2004) The developed model performed normalization through two different processes. First stemming was used based on Porter Stemmer. Next dictionary-based lemmatization was used for transformation of words into their basic morphological form and to handle compound words that were not handled by the stemmer module. This was done to retrieve clusters of relevant documents based on search query.	The results showed lemmatization performed better than stemming. This result was obtained from comparison between four hierarchical clustering methods.	Clustering of Documents for non-normalized text is identified as future research topic for sake of comparison between the two techniques.
7	Spanish	(Ruiz and Pablo 2014) The proposed model defined 110 manually annotated mappings between strings and OOV items. The mappings also provided a rectification for the expressions matched by the patterns. The mappings were implemented as case-insensitive regular expressions. A dictionary for IV items was used for validation of generated correction candidates. .	The proposed system performed better than the base-line model for accuracy but did not give better results for recall.	Improvement in the results for recall require modifications to the proposed model. Candidate selection method can also be improved by using better statistical methods rather than selection from K-best candidates determined based on distance formula.
8	Arabic	(Darwish and Kareem 2012) An existing model for tokenization developed for Arabic language was used but was modified to incorporate stemming for achieving normalization.	Significant performance gain was not achieved as compared to the baseline model.	Stemming and stop word handling requires significant modification for better performance.
9	Vietnamese	(Nguyen and Vu 2016) The proposed system first detects spelling mistakes from the input text by using a built-in dictionary for all Vietnamese morphosyllables. A morphosyllable in the input string was identified as an error if it did not appear in the morphosyllable dictionary. These mistakes are corrected using an improved Dice's coefficient.	The experimental results showed that proposed system achieved state-of-the-art performance with F1 score of 82.13 percent.	Better results are hoped to be achieved by using larger datasets and to apply bigram, trigram, and four-gram to the proposed model for improving the system performance.
10	Polish	(Brocki and Łukasz 2012) A rule based-approach was proposed with over 1500 manually defined rules applied for achieving normalization of text.	The proposed system achieved more than 80 percent accuracy of result as was confirmed by human expert analysis.	Only works well with domain dependent data. Data from other sources might not produce accurate results.
11		(Irvine and Ann 2012) The developed model is designed to carry out the deromanization as well as normalization of text in a single step. The Hidden Markov Model (HMM) was used for this purpose. A 5000 sms dataset was used to convert each text in its Roman Urdu, Urdu and English form. Both dictionary based and transliteration methods were used for this purpose.	Accuracy of results were significantly better for both evaluation by character and word error rate as compared to the baseline model.	The model performs better for deromanization then it performs for normalization. Therefore, normalization techniques need to be further modified.
	Roman Urdu	(Rafae and Abdul 2015) The developed model is based on a phonetic encoding scheme called UrduPhone designed for text in Roman Urdu. The scheme is similar to Soundex algorithm but produces better results as the encoding length of the proposed scheme is six whereas that of Soundex in four due to which Soundex is prone to producing identical root for different words whereas this scheme maps different words to more relevant root forms. Consonant groups are also introduced in this scheme where as they are missing in Soundex algorithm.	The developed model achieved an accuracy gain of up to 12 percent and 8 percent in Web and SMS datasets respectively as compared to the baseline model.	Performance gain can be enhanced by fine tuning the proposed scheme for reducing of a word to its root form

6. Need for unification

Roman Urdu is a form of writing readily adopted by Urdu speaking community in an English oriented environment. It happens to be widely used for communication on social networks by Pakistani users to voice their opinions. Our research is basically centered towards analyzing data in Roman Urdu taken from social media websites to perform opinion mining.

Roman Urdu is a just a symbolic expression of words in Urdu language using English character set so as to overcome the shortcoming of not knowing English language well enough to communicate while still being able to use interfaces that are designed to comprehend English language commands. Although the Urdu speaking community adapted to this arrangement quite naturally but the problem that arose at a mass level was the consistent representation of words used for communication. Since Roman Urdu is just a representation of Urdu it did not employ any defined standards for word representation. Consequently, word forms varied due to accent or pronunciation difference therefore producing multiple representations of a single string for example a common word like 'popular' was found to have the following representations in the datasets used for analysis 'mashhoor', 'mashoor', 'mashor', 'mashour', 'mashhur', 'mashhor'. Similarly, the word 'beautiful' in English was found to have four forms in Roman Urdu 'khobsorat', 'khoobsurat', 'khobsurat', 'khubsorat'.
Therefore, the need for standard representation of a word was felt deeply to perform any kind of analysis on the transliterate data One of the issues regarding standardization that surfaced quite prominently was the fact that the mapping between the character set of Urdu and English Language didn't have a one to one correspondence. To come up with a scheme that could provide us with accurate means of providing standardization of typed text we explored the ideas as proposed by the SOUNDEX Phonetic Algorithm and Zipf's Law.

The Soundex search algorithm takes as input a word, such as a person's name and produces a character string that identifies a set of words that are phonetically alike. It is very useful for searching large databases when the user has incomplete data. The method used by Soundex is based on the six phonetic classifications of human speech sounds (bilabial, labiodental, dental, alveolar, velar, and glottal), which are themselves based on the position of the lips and tongue to make the sounds. Many versions of Soundex algorithm have been proposed and successfully implemented to overcome the issues posed by large data sets ever since it was first introduced in 1981. We have specifically considered NYSIIS (New York State Identification and Intelligence System) as bases for our codification technique.

Zipf's Law states that, the probability of occurrence of words or other items starts high and tapers off. Thus, a few occur very often while many others occur rarely.
For example, in English language words like 'and', 'the', 'to', and 'of' occur often while words like 'undeniable' are rare. This law also applies to words in computer languages, operating system calls, colors in images, etc. and is the basis of many compression approaches.

The standard representation of a word required uniformity in the form in which the input text was received. One of the issues regarding standardization was that the mapping between the character set of Urdu and that of English Language don't have a one to one correspondence. To come up with a scheme that could provide us with a concrete and accurate means of providing standardization of typed text we explored the ideas as proposed by the SOUNDEX Phonetic Algorithm and Zipf's Law. We selected a list of 1000 most frequently used words in Urdu communication and extracted possible variants of these words from the data corpus collected from various sources. We then formulated rules for computing hash values of similar sounding words following the guidelines given by NYSIIS algorithm. The results showed approximately 70 percent success rate of transformation of each string with multiple forms into a standardized representation. The failure cases may be due to weakness of the rules that are being used for transformation. The application also requires to be extended to process words in English that may be included in the comments mostly comprising of text in Roman Urdu and may have significant impact on the overall sentiment of the comment. Also extending the list of most frequently used words that were considered for the formulation of the rules might help in achieving better accuracy. The standard representation of a word produced by the application depends heavily on the size of the data corpus used for extracting the word representation with highest frequency. It might produce different standard forms for the same word if the size of the dataset varies significantly. All the shortcomings identified can be taken on as research areas in the future work

7. Lexical codification of roman Urdu

Roman Urdu has become a popular form of expression on social media websites but it still lacks standard written forms of commonly used words. In order to alleviate

problems arising from inconsistent forms of writing style, a conscious lexical standardization effort becomes imperative. The specific aim of this research was to establish criteria which could assist in choosing standard form out of different forms of a single word and to propose standard forms based on some defined criteria.

We started the process with the retrieval of tweets/comments typed in Roman Urdu from multiple social media sites including twitter, urdubiography, IT Duniya, Reddit, Names4muslims, Pakish News and Shashca.com. For extraction of data we used Tweepy, one of the existing twitter API's in Python. Selenium, another library in Python was used for web scraping along with urllib2. We also used PhantomJS to scrap the web sites. Since we have retrieved data from multiple websites different libraries were included to overcome the structural difference in which data was represented by these websites.

For transliteration we used the site ijunoon's.com instead of Google transliterate merely because it was easy to access as compare to the latter and served the purpose equally well. Most of the cleaning and transformation were handled by the API and library functions used but minor issues such as removal of RT's, hash tags and URLs were handled by the designed application. The cleaning process also performed removal of extra spaces preceding a line of text, extra spaces following any line of text, numeric values and strings composed of non-English characters. Multiple spaces were reduced to a single space. Once the data is cleaned we feed it to a hashing algorithm that performs the necessary transformation based on the following rules:

TABLE 2. Rules for Transformation

S#	Substring	Replaced by
1	"ain" (at the end)	ein
2	"ar" (except at the start) "	r
3	"ai"	ae
4	"iy" (with multiple y's)	I
5	"ay" (at the end) "	e
6	"ih" (with multiple h's)	eh
7	"ey" (at the end) "	e
8	(multiple "s")	s
9	"ie" (at the end) "	y
10	"ry" (except at the end) "	ri
11	"es" (at the start) "	is
12	"sy" (except at the end) "	si
13	(multiple "a") "	a
14	"ty" (except at the end) "	ti
15	(multiple "j")	j
16	(multiple "o")	o
17	"(multiple "ee")	i
18	changing "i" in the end when it is preceded by (acefghijlmnoqrstuvwxyz)	y
19	(multiple "d")	d
20	'u'	o
21	removing 'h' if h is preceded by (acefghijlmnoqrstuvwxyz)	

These rules have been formulated on similar patterns as are followed by NYSIIS (an extension of SOUNDEX Phonetic Algorithm) but designing a Phonetic Algorithm for Roman Urdu required a lot of human annotated information as we lacked comprehensive resources to perform the desired operation. We also made use of ZIPF's law that states that the most frequently occurring words constitute only 20 percent of the vocabulary of a language. This led us to consider a list of 1000 most frequently used words in Urdu communication. We extracted all possible variants of each word from ijunoon transliteration service that led us to devise the rules for transformation and standardization.

The phonetic algorithm has been designed to begin with finding instances of different substrings, present within the word, and then replace them with destination string as specified by the transformation rules listed in Table 2. The order of substitution is very important, as the code is run sequentially once, and a different order would produce a different hash value. This step produces a common hash value for all the words that have similar sound.

After the common hash produces groups of similar sounding words, we then select the word representation with highest frequency in the document and replace all occurrences of the words in the group by that one representation to accomplish standardization of typed form in Roman Urdu. For Example: zaroori, zaruri, zarori map to the common hash zrory. So zrory becomes the group value for all representations mentioned above. We then pick zaruri as standard representation of all forms of this word as it was the most frequently used representation in the given document or set of documents.

8. Algorithm for Romanizing Urdu

1. Scrap data from websites into a text file
2. Clean raw data
    a. Remove Retweets, hash tags and URLs.
    b. Remove extra spaces preceding a line of text
    c. Remove extra spaces following any line of text
    d. Remove numeric values and words composed of non-English characters.
    e. Replace multiple spaces by a single space.
3. Read input from text file
4. Compare each string with a pre-compiled list of Proper Nouns
5. IF the string is NOT FOUND
    Retain it for further processing
6. ELSE
    Discard it

7. Compute HASH values for each string according to the RULES

IF RULE== TRUE

REPLACE (Source string, destination string)

8. Group strings with same hash values

COMMON_HASH (string 1, string 2…string N)

9. Compute frequency of each string in the group

$$\sum_{i=1}^{n} (\text{string N})$$

10. SELECT the string with highest Frequency

$$\text{MAX}\left[\sum_{i=1}^{n} (\text{string N})\right]$$

11. Replace all instances of the strings in the group with the HIGHEST Frequency string.

9. Findings

After a detail analysis of the methods used for normalization we found that most of the proposed models were using one of the following techniques:

a. Lexicon Based Approach
b. Rule-Based Approach (Classification Techniques)
c. Machine Learning Algorithms (SVM, CRF, HMM)
d. OOV Model (highly domain specific word list)
e. Phonetic Algorithms (Soundex, NYSIIS, etc.)
f. Stemming & Lemmatization.

Some of these techniques produce consistent results regardless of the domain they are applied on. For example, MLA, Lemmatization and Phonetic algorithms showed trends of better results as compared to OOV Model and Stemming.

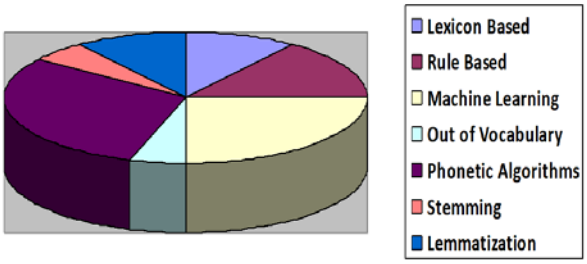

Figure 1. Comparative Analysis of Techniques Used for Normalization

The proposed algorithm for normalization of Roman Urdu text is based on the Phonetic Algorithms. The transformation rules defined accomplished a decent level of accuracy but can be further tailored to produce better results. Also, Machine Learning Techniques might help in producing better results as they have proven from the literature surveyed that they deliver better performance than other techniques.

We have analyzed data from different websites namely, Twitter, Reddit, Urdu Poetry and Social Workers Biographies. We have also handpicked some data files for further analysis to establish better credibility of our results. Our dataset comprises of 10 input Files from sources stated in Table 3.

TABLE 3. Datasets

S#	Source	Word Count
1	Bio Social Workers	12000
2	Bio Graphies	123000
3	Blog Khuwaar	3000
4	Reddit	1300
5	City News Tweets	110000
6	Express Urdu Tweets	23000
7	Nida Imranist	2000
8	Urdu SMS	5000
9	Shashca	500
10	Pakish News	1000

The results show a 70 percent and above success rate of transformation of each input string with multiple forms into a standardized representation. The failure cases may be attributed to weakness of the rules that are being used for transformation in some cases. For example, parhe, paray, parey, pre, pare, pharhe are being clustered into a distinct group whereas they are two different words ‘Parhe’ means study, ‘paray’ means lying on something or it might also mean away or far. The results shown in the table have also been presented as graphs in Figure 2 and 3. For Figure 2 we have considered the data file from Bio Graphies and for Figure 3 we have excluded this file, for clarity sake, as its size is significantly different from other files and was dominating the results in Figure 2.

TABLE 4. Lexical Normalization Results

Source	Groups	Correct	Wrong	Success Rate
Bio Social Workers	450	394	56	87.5%
Bio Graphies	1700	1550	150	91.1%
Blog Khuwaar	63	51	12	80.9%
Reddit	26	19	7	73.0%
City News Tweets	226	200	26	88.4%
ExpressUrdu Tweets	400	348	52	87.0%
Nida Imranist	43	35	8	81.3%
Urdu SMS	115	102	13	88.6%
Shashca	11	11	0	100%
Pakish News	6	6	0	100%

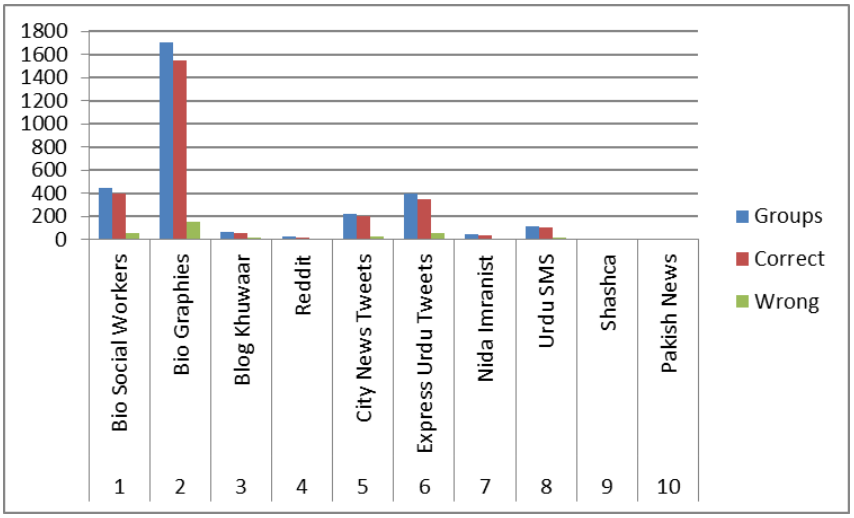

Figure 2. Comparative Analysis with Biographies Data Source

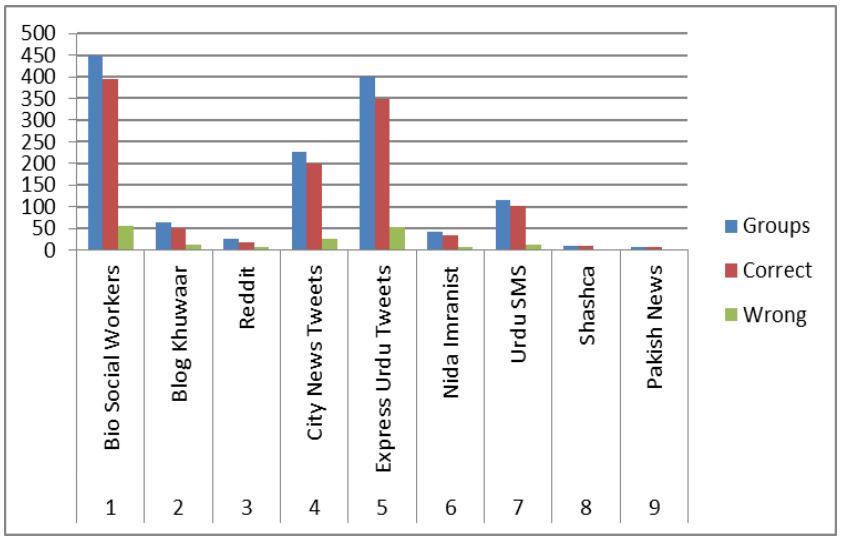

Figure 3. Comparative Analysis without Biographies Data Source

10 .Conclusion

Normalization of social media content is a difficult and challenging task. The normalization task is compulsory to transform informal text into a consistent standard format. Since user generated content is very rich in the use of abbreviations, nonstandard words, and out of vocabulary terms this causes a lot of complications for the natural language processing tools to perform analysis of such text. In this study, we have conducted a detailed survey of the text normalization techniques used for achieving text normalization for different languages like English, Arabic, Chinese, Japanese, Dutch, Finnish, Polish, Bangla, Vietnamese and Roman Urdu. We have highlighted the accuracy rate of the proposed models as well as their shortcomings. We have also proposed an algorithm for performing normalization of text in Roman Urdu with a high accuracy rate.

References

[1] Ahmed Tafseer 2009. Roman to Urdu Transliteration using word list. Proceedings of Conference of Language and Technology 09, Lahore.

[2] Alam and Firoj 2008. Text normalization system for Bangla. BRAC University.

[3] Beaufort and S. Roekhaut, et al 2010. A hybrid rule/model-based finite-state framework for normalizing SMS messages. 48th Annual Meeting of the Association for Computational Linguistics, 770-779, Uppsala, Sweden.

[4] Brocki and Łukasz 2012. Multiple model text normalization for the polish language. International Symposium on

Methodologies for Intelligent Systems. Springer Berlin Heidelberg.
[5] Choi D and Kim J 2014. A method for normalizing non-standard words in online social network services: A case study on twitter. Second International Conference Context-Aware Systems and Applications, ICCASA.
[6] Darwish and Kareem 2012. Language processing for Arabic microblog retrieval. Proceedings of the 21st ACM international conference on Information and knowledge management. ACM.
[7] Fasih Uddin Syed and Begum Quader Unissa 1992. The Modern International Standard Letters of Alphabet for URDU - (HINDUSTANI) - The INDIAN Language, script for the purposes of hand written communication, dictionary references, Computerized Linguistic Communications (CLC). Chicago.
[8] Gonzalez and Marco 2006. Lexical normalization and relationship alternatives for a term dependence model in information retrieval. International Conference on Intelligent Text Processing and Computational Linguistics. Springer Berlin Heidelberg.
[9] Grzegorz and Chrupala 2014. Normalizing tweets with edit scripts and recurrent neural embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics Volume 2: Short Papers, pages 680–686.
[10] Han Bo, Cook Paul, and Baldwin Timothy. 2012. Automatically constructing normalization dictionary for microblogs. In Proceedings of Empirical Methods on Natural Language Processing EMNLP-CoNLL, pages 421–432.
[11] Han, B. and Baldwin, T. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 pp. 368-378. Association for Computational Linguistics.
[12] Han, B., Cook, P., and Baldwin, T. 2013. Lexical normalization for social media text. ACM Transactions on Intelligent Systems and Technology (TIST), 4(1), 5.
[13] Henrich Verena and Hinrichs Erhard 2010. Standardizing wordnets in the ISO standard LMF: Wordnet-LMF for GermaNet, Proceedings of the 23rd International Conference on Computational Linguistics.
[14] Irvine and Ann 2012. Processing informal, romanized Pakistani text messages. Proceedings of the Second Workshop on Language in Social Media. Association for Computational Linguistics.
[15] Javed Iqra and Afzal Hammad 2014. Towards creation of linguistic resources for Bilingual Sentiment Analysis of Twitter Data. 19th International Conference on Application of Natural Language to Information Systems.
[16] Kaji and Nobuhiro 2014 Accurate Word Segmentation and POS Tagging for Japanese Microblogs: Corpus Annotation and Joint Modeling with Lexical Normalization. Empirical Methods on Natural Language Processing EMNLP.
[17] Karmani B.M Nadia, Soussou Hsan and Alimi M. Adel 2014. Building a standardized Wordnet in the ISO LMF for aeb language. 7th Global WordNet Conference.
[18] Korenius and Tuomo 2004. Stemming and lemmatization in the clustering of Finnish text documents. Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM.
[19] Kumar Parteek, Sharma R.K. and Narang Ashish 2014. Creation of Lexical Relations for IndoWordNet. 7th Global WordNet Conference.
[20] Li Chen and Liu Yang 2014. Improving text normalization via unsupervised model and discriminative reranking. In Proceedings of Association for Computational Linguistics ACL.
[21] Li Chen and Liu Yang 2015. Joint POS tagging and text normalization for informal text. In Proceedings of International Joint Conference on Artificial Intelligence IJCAI.
[22] Liu, Fei Weng Fuliang, and Jiang Xiao. 2012 A broad-coverage normalization system for social media language in Proceedings of Association for Computational Linguistics ACL.
[23] Liyew Zelalem, Hundie and Mekonnen 2002 Lexical Standardization in Oromo. Addis Ababa University Institutional Repository.
[24] Malik, Abbas M, and Boitet, 2008. Hindi Urdu Machine Transliteration using Finite-state Transducers. Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK.
[25] Mariooryad, Soroosh, and Busso Carlos 2014. Compensating for speaker or lexical variabilities in speech for emotion recognition. Speech Communication 57: 1-12.
[26] Meenakshi and Petchiamma 2016. Normalization of NSW (Non Standard Words) using DSM model in case of OOV (Out-Of- Vocabulary words. International Journal of Innovative Research in Science, Engineering and Technology Vol. 5, Issue 8.
[27] Nahir and Moshe 2003. Micro-corpus codification in the Hebrew Revival. Digithum 5.
[28] Nguyen andVu H 2016.Text normalization for named entity recognition in Vietnamese tweets." Computational Social Networks 3.1: 10.
[29] Rafae and Abdul 2015. An Unsupervised Method for Discovering Lexical Variations in Roman Urdu Informal Text." Empirical Methods on Natural Language Processing EMNLP.
[30] Pablo Ruiz, Montse Cuadros and Thierry Etchegoyhen 2014. Lexical Normalization of Spanish Tweets with Rule-Based Components and Language Models. Procesamiento del Lenguaje Natural, Sociedad Espanola para el Procesamiento del Lenguaje Natural, pp.8.
[31] Schulz and Sarah 2016. Multi-Modular Text Normalization of Dutch User-Generated Content. ACM Transactions on Intelligent Systems and Technology.
[32] Supranovich and Dmitry 2015. IHS_RD: Lexical Normalization for English Tweets. Association for Computational Linguistics ACL-IJCNLP: 78.
[33] Wang and Aobo 2013. Chinese Informal Word Normalization: An Experimental Study. International Joint Conference on Natural Language Processing IJCNLP.
[34] Wijesiri, Indeewari and Gallage Malaka 2014. Building a WordNet for Sinhala, 7th Global WordNet Conference.