6-26-2018

Text Classification in an Under-Resourced Language via Lexical Normalization and Feature Pooling

Omayya Sohail
Information Technology University of the Punjab, mscs16017@itu.edu.pk

Inam Elahi
Information Technology University of the Punjab, mscs14021@itu.edu.pk

Ahsan Ijaz
ADDO AI, ahsan@addo.ai

Asim Karim
Lahore University of Management Sciences, akarim@lums.edu.pk

Faisal Kamiran
Information Technology University of the Punjab, faisal.kamiran@itu.edu.pk

Text Classification in an Under-Resourced Language via Lexical Normalization and Feature Pooling

Completed Research Paper

Omayya Sohail
Information Technology University of the Punjab
Lahore, Pakistan
omayya.sohail@itu.edu.pk

Inam Elahi
Information Technology University of the Punjab
Lahore, Pakistan
mscs14021@itu.edu.pk

Ahsan Ijaz
ADDO AI
Lahore, Pakistan
ahsan@addo.ai

Asim Karim
Lahore University of Management Sciences, Pakistan
akarim@lums.edu.pk

Faisal Kamiran
Information Technology University of the Punjab
Lahore, Pakistan
faisal.kamiran@itu.edu.pk

Abstract

Automatic classification of textual content in an under-resourced language is challenging, since lexical resources and preprocessing tools are not available for such languages. Their bag-of-words (BoW) representation is usually highly sparse and noisy, and text classification built on such a representation yields poor performance. In this paper, we explore the effectiveness of lexical normalization of terms and statistical feature pooling for improving text classification in an under-resourced language. We focus on classifying citizen feedback on government services provided through SMS texts which are written predominantly in Roman Urdu (an informal forward transliterated version of the Urdu language). Our proposed methodology performs normalization of lexical variations of terms using phonetic and string similarity. It subsequently employs a supervised feature extraction technique to obtain category-specific highly discriminating features. Our experiments with classifiers reveal that significant improvement in classification performance is achieved by lexical normalization plus feature pooling over standard representations.

Keywords: Roman Urdu, document classification, feature pooling, lexical normalization

Introduction

Social media such as Twitter, WhatsApp and Short Messaging Service (SMS) texts are rapidly becoming an important channel for providing feedback on products and services. Knowledge gained from user feedback can help private and public sector entities in improving their service or product, thereby enhancing their user experience. However, feedback messages sent as tweets and SMS texts are short in length and commonly written in informal verbiage. Therefore, understanding such textual content is a challenging task. In particular, building accurate classification models for SMS texts is

difficult due to high dimensionality and sparsity of their representation obtained using standard text processing procedures. This problem is exacerbated for under-resourced languages. Such languages are not standardized and do not have lexical resources and processing tools required for building effective representations for classification.

The Citizen Feedback Monitoring Program (CFMP)[1] of the Government of Punjab (provincial government) in Pakistan provides a unique yet important application of automatic text classification in an under-resourced language. The CFMP aims to reduce the frequency of petty corruption and poor delivery in public service transactions by collecting and analyzing feedback from citizens sent via SMS texts. The CFMP was piloted in six districts of Punjab in 2010, and expanded to all 36 districts of Punjab and 17 public services ranging from property registration to driving license issuance from 2012-14. As of September 2015, the CFMP had recorded 11.18 million transactions, contacted 8.6 million citizens for feedback, and received SMS-based feedback from 1.12 million citizens. Currently, the SMS texts are manually labeled by human annotators into one of 19 categories such as corruption and appreciation. Subsequently, these texts are analyzed to improve management practices or reprimand concerned officials. To avoid this time-consuming procedure of manual annotation of SMS texts, we connected with Punjab IT Board (PITB) to deploy our proposed solution on their citizen feedback data.

The SMS texts in CFMP are written predominantly in Roman Urdu, which is an informal transliterated version of the Urdu language written with Latin characters rather than the Perso-Arabic script in Urdu. Roman Urdu is not standardized and no reliable specific resources and tools are available for its processing. Although there has been some recent research on Roman Urdu normalization (Rafae et al. 2015), no work has been reported for the practical task of Roman Urdu text classification.

In this paper, we present a methodology for generating enhanced representations of Roman Urdu short texts for their accurate classification. Our methodology employs normalization of spelling variations of words using phonetic and string similarity matching. After this reduction in dimensionality, we employ a supervised feature extraction technique to construct discriminative features for classification. This technique requires a single pass over the data but produces effective features for classification. We evaluate our methodology on real-world citizen feedback data from Pakistan (CFMP). The results show significant improvement in classification F-measure after application of our methodology.

The rest of the paper is organized as follows. Section 2 discusses the related work in short text and Roman Urdu processing. Section 3 details our methodology for building accurate text classification models for under-resourced languages. Our experimental setup and experimental results are discussed in Section 4. We present our concluding remarks in Section 5.

Related Work

Text classification is the task of automatically labeling a set of documents with one of many predefined classes or categories. This task requires term/word selection and document representation typically through assigning weights to the selected terms, and selection of an appropriate classifier (Sebastiani 2005). One of the most commonly used representation for text classification is through document-term matrix or bag-of-words (BoW) representation. This representation obtained by standard text processing has been shown to produce accuracies of around 90% for well written documents in standard languages such as English (Sebastiani 2005).

Unfortunately, the BoW representation of short-length informal texts like tweets and SMS texts is very sparse as the number of distinct terms is very large while each document contains a few terms only. Normalization of terms can reduce dimensionality but most proposed techniques for normalization are applicable to standard languages only for which in-vocabulary (available in the lexicon) terms are known. Identification and normalization of lexical variations in short texts can be achieved by building a classifier and generating correction candidates based on morphophonemic similarity (Han and Baldwin 2011), whereas the normalization of nonstandard terms can also be performed without explicitly classifying them (Liu et al. 2011). Use of noisy-channel framework with the incorporation of

1 http://cfmp.punjab.gov.pk/

orthographic, phonetic, contextual, and acronym expansion information to calculate the likelihood probabilities of terms also helps in normalizing informal terms (Xue et al. 2011). Phonetic and string similarity is also used in normalizing nonstandard terms obtained from Twitter and SMS feeds into standard English words (Liu et al. 2012). Previous literature on short texts show serious degradation of text preprocessing tasks due to the presence of nonstandard terms. For example, it has been reported that the Stanford named entity recognizer (NER) experiences a performance degradation from 91% to 46% on tweets (Liu et al. 2011). Therefore, it is highly recommended to normalize such noisy texts before building natural language and machine learning models on them.

For forward transliterated, under-resourced languages such as Roman Urdu, all of the above-mentioned techniques are not applicable since there is no in-vocabulary term in such languages. Similarly, the use of language morphology for normalization is not possible since the stems, root words, suffixes or prefixes cannot be identified (Argamon et al. 2009; Scott and Matwin 1999). Recently, a clustering approach and a phonetic approach has been proposed for normalizing Roman Urdu texts. The former approach uses a combination of phonetic, string, and contextual similarity for finding groups of variant terms (Rafae et al. 2015). However, this approach, like all other techniques for normalization, has not been designed or evaluated for text classification. The latter approach uses phonetic similarity for grouping homophones (Sharf and Rahman 2017). The transformation rules for Roman Urdu text are designed on similar patterns as the guidelines defined in NYSIIS (New York State Identification and Intelligence System) algorithm. But since these rules are formulated on the basis of a limited wordlist of most frequently used terms in Urdu communication, the results show approximately 70% success rate of normalizing nonstandard terms.

In addition to normalization, feature selection and extraction, an essential step for accurate text classification, is also restricted for under-resourced languages. In particular, the orthographic, morphological (Ng et al. 1997), contextual (Gabrilovich and Markovitch 2005), and synset (Bloehdorn and Hotho 2004) related information cannot be exploited.

Methodology

We present a stage-wise methodology for improving text classification in an under-resourced language. The methodology involves two incremental stages of processing and representation. The first stage performs normalization of terms using phonetic and string similarity, while the second stage applies a feature pooling technique to obtain discriminative features for classification. These enhanced representations are used instead of the standard bag-of-words (BoW) representation for text classification.

Standard BoW Representation

The bag-of-words (BoW) representation is popularly used for text classification (Liu 2011). It assumes that each textual content (SMS text, tweet, or in general a document) is a collection of words appearing in it. Let D be a set of documents and T be the set of distinct terms (words) appearing in them. Each term $t_j \in T$ is a distinct sequence of one or more characters delimited by spaces and/or punctuation marks. In the BoW model, each document $d_i \in D$ is a set of terms from T. The significance of term t_j in document d_i is quantified by the weight $w_{ij} \geq 0$. A larger weight implies that the corresponding term is more important in the given document within the document collection. Typically, we represent all weights in a document-term matrix $\boldsymbol{W}$ of size $N \times M$ where $N = |D|$ is the number of documents and $M = |T|$ is the number of distinct terms in the collection. As such, each document is represented by the corresponding row vector in $\boldsymbol{W}$ of length M.

Lexical Normalization

The standard BoW representation is usually high dimensional and sparse for short informal document collection. This is because each document contains only a few terms while the total number of distinct terms in the collection is large. This problem is exacerbated in under-resourced languages where standard forms of terms are not defined. For example, in Roman Urdu we find six commonly occurring

variations for the word [bribe]: *rishwat*, *rishwt*, *rshwt*, *rishwatt*, *rashwat*, *rashwt*. In standard BoW representation of such languages each of these variants will be identified as a distinct term even though semantically they capture the same concept and will be one entry in a lexicon.

For forward transliterated languages like Roman Urdu, words are spelled to sound similar to their corresponding words in Urdu. Therefore, phonetic encoding provides a natural way of finding lexical variations in such languages. Moreover, lexical variations have high string similarities. These ideas have been adopted recently for finding lexical variations in Roman Urdu (Rafae et al. 2015). We adopt a computationally efficient two-pass algorithm to build a normalized representation of Roman Urdu as opposed to using a clustering approach for finding lexical variations of words (Rafae et al. 2015).

Algorithm 1 shows our procedure for lexical normalization of textual content in an under-resourced language that takes as input the original sets of documents and terms (D, T) and outputs the normalized sets of documents and terms $(\bar{D}, \bar{T})$. In general, the number of terms $\bar{M} = |\bar{T}|$ after normalization is less than M, the number of terms before normalization. We use Soundex (SX) encoding to group variations based on phonetics, since it performs better in terms of recall as compared to other phonetic encoding algorithms like Metaphone, Caverphone and NYSIIS (Rafae et al. 2015). A group is split into two when terms in it have Levenshtein distance (LD) greater than 2 with the longest term in it. Subsequently, the longest term in each group is taken as its representative and the document collection is normalized accordingly. After this procedure, we obtain a normalized document-term matrix $\overline{\boldsymbol{W}}$ of size $N \times \bar{M}$.

Algorithm 1. Lexical Normalization

Input: D, T
Output: $\bar{D}, \bar{T}, \bar{M}$
$\bar{T} \leftarrow \emptyset, \bar{M} \leftarrow 0, \forall i\ T_i \leftarrow \emptyset$
for all $t_i \in T$ **do**
    $\bar{M} \leftarrow \bar{M} + 1$
    $T_{\bar{M}} \leftarrow t_i$
    for all $t_j \in T$ **do**
        if $SX(t_i) = SX(t_j)$ **then**
            $T_{\bar{M}} \leftarrow T_{\bar{M}} \cup t_j$
            $T \leftarrow T \backslash t_j$
        end if
    end for
    $t_l \leftarrow Longest\ term\ in\ T_{\bar{M}}$
    for all $t \in T_{\bar{M}}$ **do**
        if $LD(t_l, t) > 2$ **then**
            $\bar{M} \leftarrow \bar{M} + 1$
            $T_{\bar{M}} \leftarrow T_{\bar{M}} \cup t$
        end if
    end for
end for
for $i = 1 \rightarrow \bar{M}$ **do**
    $t_l \leftarrow Longest\ term\ in\ T_i$
    $\bar{T} \leftarrow \bar{T} \cup t_l$
    for all $t \in T_i$ **do**
        $\bar{D} \leftarrow$ Normalized collection by replacing all occurrences of t with t_l
    end for
end for

Feature Pooling

Since our intention is to solve the supervised problem of text categorization, the feature extraction techniques that rely on class disparity information are particularly attractive. Furthermore, we desire a feature extraction technique that exploits the semantics that similar words are used in similar contexts which in our case is defined by categories. For example, *rude* and *batmiz* [disrespectful], the two semantically similar words found in the dataset, fall into the same category, i.e., 'bad attitude'. Recently, it has been demonstrated that discrimination information provided by terms in a labeled document collection also quantifies the relatedness of the terms to the respective contexts in the collection (Junejo et al. 2016; Tariq and Karim 2011). Based on this semantic notation of relatedness, features are constructed by pooling the discrimination information of terms related to each context. Such a technique is especially suited to text in an under-resourced language because it considers the contextual usage of terms to overcome the limitations of phonetic and string similarity in effective representation.

Let $\{d_i, c_i\}_{i=1}^{N}$ be the collection of labeled documents where $d_i \in \overline{D}$ is the ith document in the normalized document collection and $c_i \in C$ is the category or class label of the ith document. We assume that $K = |C| \geq 2$ is the number of categories. The discrimination information provided by a term $t_j \in \overline{T}$ for category k $(k = 1,2, \dots, K)$ is given by its discriminative term (Junejo and Karim 2008; Junejo et al. 2016):

$$\varphi_{jk} = \frac{p(t_j|k)}{p(t_j|\neg k)} \geq 1;\ 0\ Otherwise \tag{1}$$

where $p(t_j|k)$ is the probability of term t_j appearing in documents belonging to category k and $\neg k$ denotes all categories but category k. The discriminative term weight is the relative risk of the term appearing in a category as opposed to the other categories; its value is greater than or equal to 1, or zero when the numerator is less than the denominator. Table 1 depicts the highest discriminative term weight in a category, for example, the term *ravia* [behavior] in $k = 2$ has the highest weight i.e., approximately 115.33; it also shows the top 5 terms per category.

Table 1. Highest Discriminative Term Weight (DTW) along with Top 5 Terms per Category

k	Category	Highest DTW	Top 5 Terms in Category *k*
1	appreciation	~13.10	*punjab*; *shukria*, *thx* [thanks]; *pakistan*; *effort*
2	bad attitude	~115.33	*ravia* [behavior]; *rude*; *batmiz* [disrespectful]; *handl* [handle]; *staaf* [staff]
3	bought all medicine from outside	193.12	*inject*; *store*; *drip*; *krwana*, *parte* [have to do]
4	bought some medicine from outside	~409.67	*facility*; *adwiyat* [medicines]; *except*; *hospital*; *outside*
5	corruption	40.00	*bnaya* [made]; *krani* [have to do]; *batayn* [tell]; *drkhwast* [request]; *direct*
6	corruption in other offices	1000.00	*kachahri* [court]; *leyi* [took]; *nikalvai* [issued upon request]; *putvari* [village accountant]; *aawam* [the public]
7	delayed response	226.50	*late*; *ponchi* [arrived]; *wardad* [crime]; *ayii* [arrived]; *batayn* [tell]
8	don't know	307.00	*leave*; *support*; *after*; *before*; *bhetrin* [best]
9	further inquiry	~9.27	*badtmizi* [disrespect]; *btain* [tell]; *clark* [clerk]; *drkhwast* [request]; *kadm* [step]
10	grievance ascribed to opponents	~245.40	*application*; *krva* [make someone do something]; *rest*; *approval*; *arrest* [arrest]

11	medicine/machinery not available	122.20	*card*; *exra* [x-ray]; *karey* [do]; *medicen* [medicine]; *barhaen* [increase]
12	no problem	~11.43	*everything*; *let*; *perform*; *satsifactori* [satisfactory]; *nice*
13	no response	~40.07	*accident*; *arrive*; *gantay* [hours]; *related*; *sarvic* [service]
14	objection raised	1231.00	*due*; *object*; *rashwat* [bribe]; *manghi* [demanded]; *still*
15	obnoxious or irrelevant	307.00	*prayer*; *mubrk* [congratulations]; *happi* [happy]; *special*; *response*
16	other	~72.67	*admit*; *bethi* [sitting]; *accident*; *badtmizi* [disrespect]; *bagher* [without]
17	other complaint	~29.19	*ilag* [treatment]; *krwaya* [did]; *hospital*; *zror* [must]; *behtr* [better]
18	unaware	250.00	*kes* [which]; *pleez* [please]; *tafsel* [details]; *pasay* [money]; *puch* [ask]
19	wrong person	87.00	*bnwaya* [made]; *countr* [counter]; *ilaj* [treatment]; *send*; *wrong*

Given the discriminative term weights, the discrimination information provided by document d_i for category k is given by linear opinion pool of the discriminative term weights of all terms in it for category k (Junejo and Karim 2008; Junejo et al. 2016):

$$w_{ik}^{*} = \frac{\sum_{j=1}^{\bar{M}} \bar{w}_{ij} \times \varphi_{jk}}{\sum_{j=1}^{\bar{M}} \bar{w}_{ij}} \tag{2}$$

The weight w_{ik}^{*} represents the kth feature for document d_i. All such weights can be combined in the document-feature matrix $\boldsymbol{W}^{*}$ of size $N \times M^{*}$ where $M^{*} = K$ is the number of categories. Table 2 shows the document-feature matrix (with weights rounded off to one decimal place) for a subset of dataset.

Table 2. Document-Feature Matrix of a Subset of Dataset

d_i	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	k11	k12	k13	k14	k15	k16	k17	k18	k19
1	5.1	0.9	1.3	1.3	0.6	0.4	0.8	1.6	0.5	0.9	0.8	0.6	1.2	0.4	0.1	1.0	1.2	0.1	1.1
2	1.0	1.4	1.5	1.2	0.8	1.4	1.0	3.6	0.5	2.2	0.9	0.9	1.4	6.2	0.6	1.1	1.3	1.8	1.4
3	0.8	2.7	1.7	0.3	9.0	5.1	1.8	0.9	0.5	2.4	0.7	0.4	1.8	1.4	0.4	2.5	1.5	0.2	0.4
4	0.7	1.3	2.0	2.2	4.6	0.7	1.0	4.7	0.5	1.8	1.8	0.6	1.4	0.7	0.3	2.0	1.7	1.7	2.4
5	1.3	1.4	1.6	0.7	1.4	1.5	0.9	0.9	0.5	5.1	1.3	0.8	1.3	0.4	0.5	4.3	1.4	0.1	0.9
6	0.8	1.3	0.6	0.3	9.1	4.6	1.3	5.9	0.6	0.5	4.7	0.5	1.2	0.8	0.3	1.3	0.8	0.2	0.3
7	0.6	40	1.7	0.2	2.2	2.0	0.7	0.9	0.2	14	1.9	0.4	0.6	0.2	0.2	3.2	1.2	0.3	0.3
8	1.9	3.0	3.5	0.9	0.7	0.2	0.6	0.1	0.1	0.2	2.4	0.4	0.1	0.2	0.0	0.8	1.5	0.1	0.1
9	0.9	0.0	0.0	0.0	1.9	0.0	0.0	0.0	0.2	3.1	0.0	2.5	0.0	0.0	0.0	0.9	0.2	0.0	0.0
10	0.8	2.1	1.7	0.3	1.2	0.6	1.2	0.7	0.3	1.7	19	0.6	1.8	0.4	0.3	1.6	1.5	1.0	0.6

Experiments and Results

In this section, we describe the dataset, the experimental setup, and the results of experimental evaluation of our methodology for text classification in an under-resourced language.

Data Description

The Citizen Feedback Monitoring Program (CFMP) is an ongoing project of the Government of Punjab in Pakistan that uses Information and Communication Technology (ICT) to proactively reach out to citizens who have recently interacted with government departments, and collect feedback on the quality of service delivery provided by these departments. The primary objective of the program is to curb petty corruption and improve service delivery of government departments. Proactively reaching out to citizens through an independent body increases the trust of the citizen in the state and increases the chances of honest feedback.

Feedback via SMS texts is collected from citizens who have recently availed a government service. The feedback process is initiated by a robocall to the citizen in the voice of the Chief Minister of Punjab followed by an SMS text requesting feedback. Citizens provide free-form responses through SMS texts. The writing is informal in nature and predominantly in Roman Urdu.

We received a dataset from CFMP containing 12,398 SMS texts that are manually classified into one of 19 categories. The corruption category had 506 texts, while appreciation and no-problem categories had the greatest number of SMS texts. Table 3 shows the percentage distribution of some categories in selected government services. For example, within the Health department, the three services listed are Dialysis (at Kidney Centers, etc.), Emergency (at Public Hospitals), and Rural Health Center (RHC) Indoor services. Four percent of all feedback is categorized in corruption, and 42.49% of all corruption feedback is related to the domicile service.

Table 3. Percentage Distribution of Selected Categories in Citizen Feedback on Selected Services in our Dataset

Service	Appreciation (%)	Corruption (%)	No Problem (%)
	34.9	4.0	30.6
Dialysis	0.41	0	0.18
Emergency	23.39	11.46	17.86
RHC Indoor	4.91	3.16	3.16
Character Certificate	6.84	5.73	5.05
Rescue 15	18.39	6.91	21.47
Domicile	20.38	42.49	25.92
LRMIS	0.46	0.19	0.15
Property	24.06	29.64	25.84

Table 4 shows some of the selected frequent terms from the appreciation and corruption classes. The values are normalized by the maximum frequency in each class. Appreciation class had terms including *shukriya* [thanks], *bohat* [very], *salook* [behavior], *thanks*. In comparison, terms in the corruption class include *rupy* [money], *diya* [gave], *liyay* [took].

Setup

We consider two text classification tasks: (a) a 19-category task and (b) a 2-category (corruption vs. all other categories) task. The latter task is motivated by the fact that corruption detection is a key objective of CFMP. We process the data and define terms by removing numbers and tokens of length 3 or less. By following our methodology, we present results for standard BoW representation (SB), normalized representation (SB+LN), and normalized plus feature extraction based representation (SB+LN+FP).

We report classification results using Random Forest (RF) and Support Vector Machine (SVM). We use Python's scikit-learn for these classifiers with their default settings. The results are measured on 40% test data, after training on 60% training data.

Table 4. Normalized Frequency of few Terms in Appreciation vs Corruption Class

Term	Appreciation Frequency	Corruption Frequency
acha [good]	1	0.070
thanks	0.710	0.050
bohat [very]	0.399	0.032
buhat [very]	0.077	0.006
diya [gave]	0.013	0.059
rupees	0.007	0.092
rupy [money]	0.002	0.114
shukriya [thanks]	0.100	0.003
salook [behavior]	0.238	0.050
liyay [took]	0.003	0.010

For task (a), we report weighted precision, recall, and F-measure values. These weighted measures are considered more reliable for evaluating multi-class problems in which class distribution varies widely (López et al. 2013). The weighting is done by the fraction of examples in each category. For task (b), we report standard precision, recall, and F-measure values for both categories separately.

Results

The processed data has 22,416 distinct terms. Thus, the SB representation has this many dimensions while each document has on average 15 distinct terms, making the SB representation very sparse. After lexical normalization, the number of distinct terms is reduced to 12,441. Thus, the SB+LN representation is richer especially considering that lexical semantics are retained by the normalization procedure.

We try different term weighting schemes (term occurrence, term count, and term-frequency-inverse-document-frequency) but find insignificant differences in classification performance among them. Therefore, we report results using term count as the term weighting scheme.

Figure 1 shows the weighted precision, recall, and F-measure values produced by both RF and SVM for the three different representations discussed in our methodology (SB, SB + LN, and SB + LN + FP).

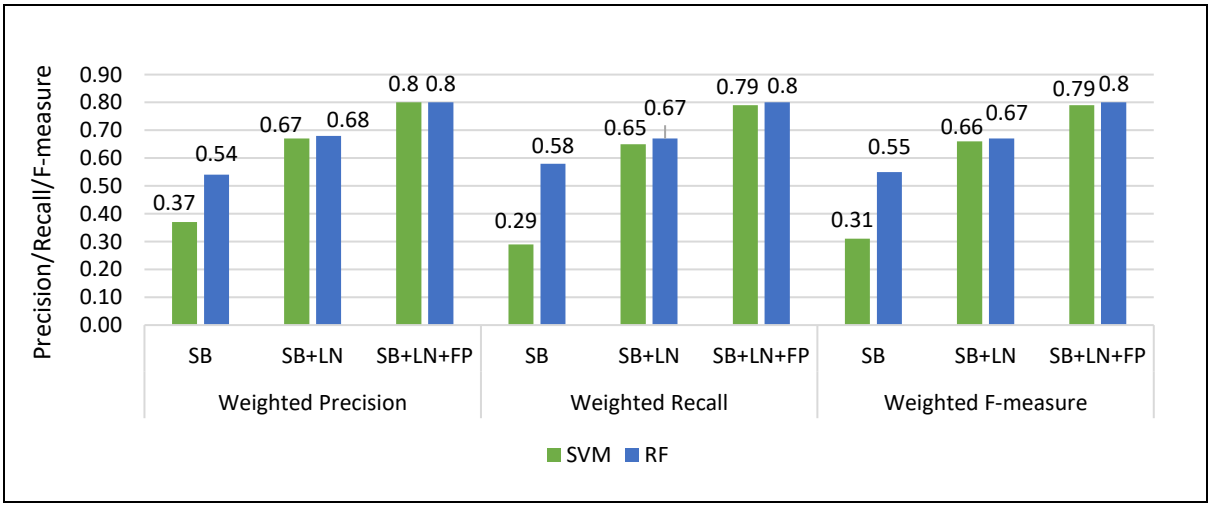

Figure 1. Weighted Precision, Recall, and F-Measure produced by Random Forest (RF) and Support Vector Machine (SVM) for Three Representations

It is observed that significant improvement in performance is observed after lexical normalization (SB+LN) and after feature pooling (SB+LN+FP) in both classifiers. For RF (Random Forest), F-measure values for SB, SB+LN, and SB+LN+FP are 0.55, 0.67, and 0.80, respectively. Thus, performance increases by over 45% from the standard BoW approach by applying our methodology. Between the two classifiers, RF is generally performing better than SVM on this data.

Our feature pooling technique exploits contextual semantics to build discriminative features. A popular technique for feature construction in text analysis is latent semantic indexing (LSI) through singular value decomposition (SVD) of the document-term matrix (Liu 2011). To highlight the effectiveness of our feature pooling technique, we also experiment with SVD of the normalized representation and linear discriminant analysis (LDA). Figure 2 shows the weighted F-measure values produced by RF and SVM when applied to SB+LN +SVD representations of 100, 250, and 500 dimensions. It is clear that SVD is not able to provide any improvement in classification performance with the best F-measure of 0.65 being similar to that produced for SB+LN (i.e., the full normalized representation). Similarly, LDA, which is a supervised learning technique provides an F-measure of 0.63. A study has also shown that text classification is generally better with feature pooling than with linear discriminant analysis (Tariq and Karim 2011).

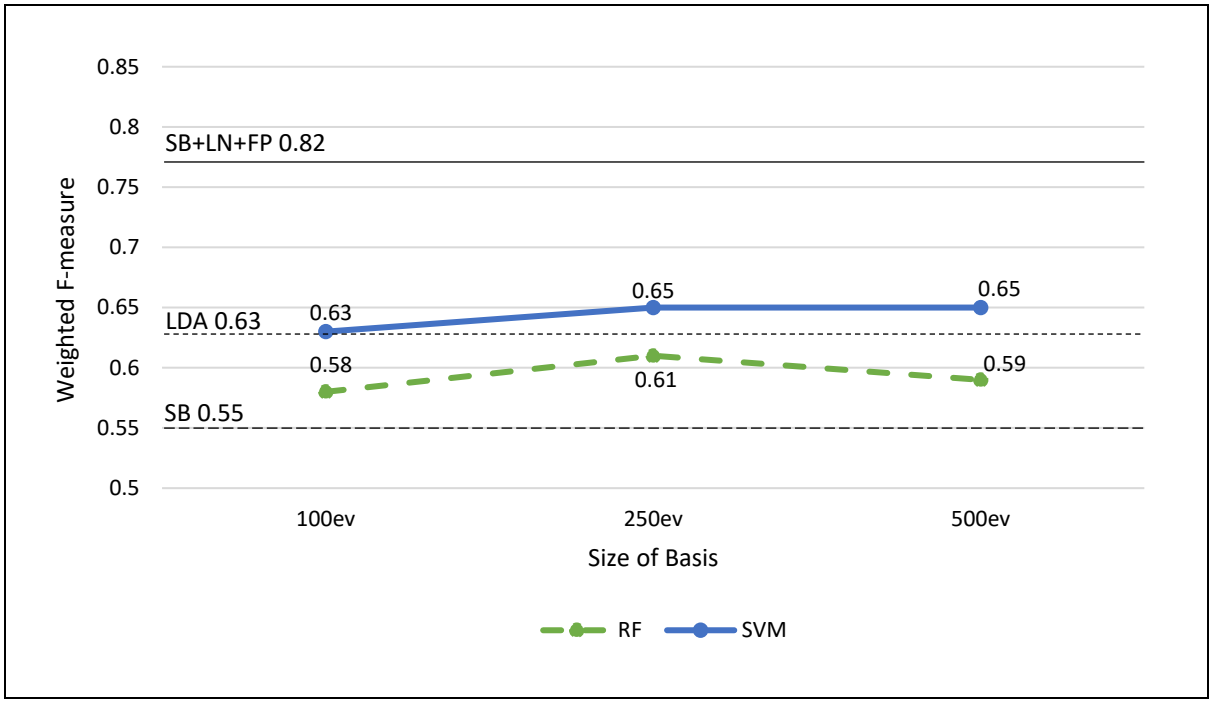

Figure 2. Weighted F-measure for SB+LN+SVD produced by RF is depicted with thick, spaced green line and by SVM with thick, solid blue line. The black dashed line shows the results of SB+LN+LDA (0.63); the black long dashed line is for SB representation (by RF) with F-measure of 0.55; the solid black line is the best performing case (SB+LN+FP by RF) with F-measure of 0.82.

Corruption Detection: As mentioned earlier, a key objective of CFMP is to detect and prevent petty corruption in government services. For this purpose, we evaluate our methodology for the two-class, corruption vs. all other categories, task. In this task, there are only 2 features in the SB+LN+FP representation. Figure 3 shows the precision, recall, and F-measure values produced by RF for this task. These are standard performance measures and hence separate values are given for each category. It is worth noting that F-measure for corruption has increased to 0.82 for SB+LN+FP. More importantly, the recall for corruption is 0.91, i.e., 91% of all corruption revealing feedback is detected by our methodology.

Summary: These results show the effectiveness of our methodology for improved text classification in

an under-resourced language. Our methodology provides a cost-effective and reliable approach for CFMP to quickly analyze citizen feedback. This is a major improvement over the laborious, time-consuming, and costly exercise of manually labeling the feedback.

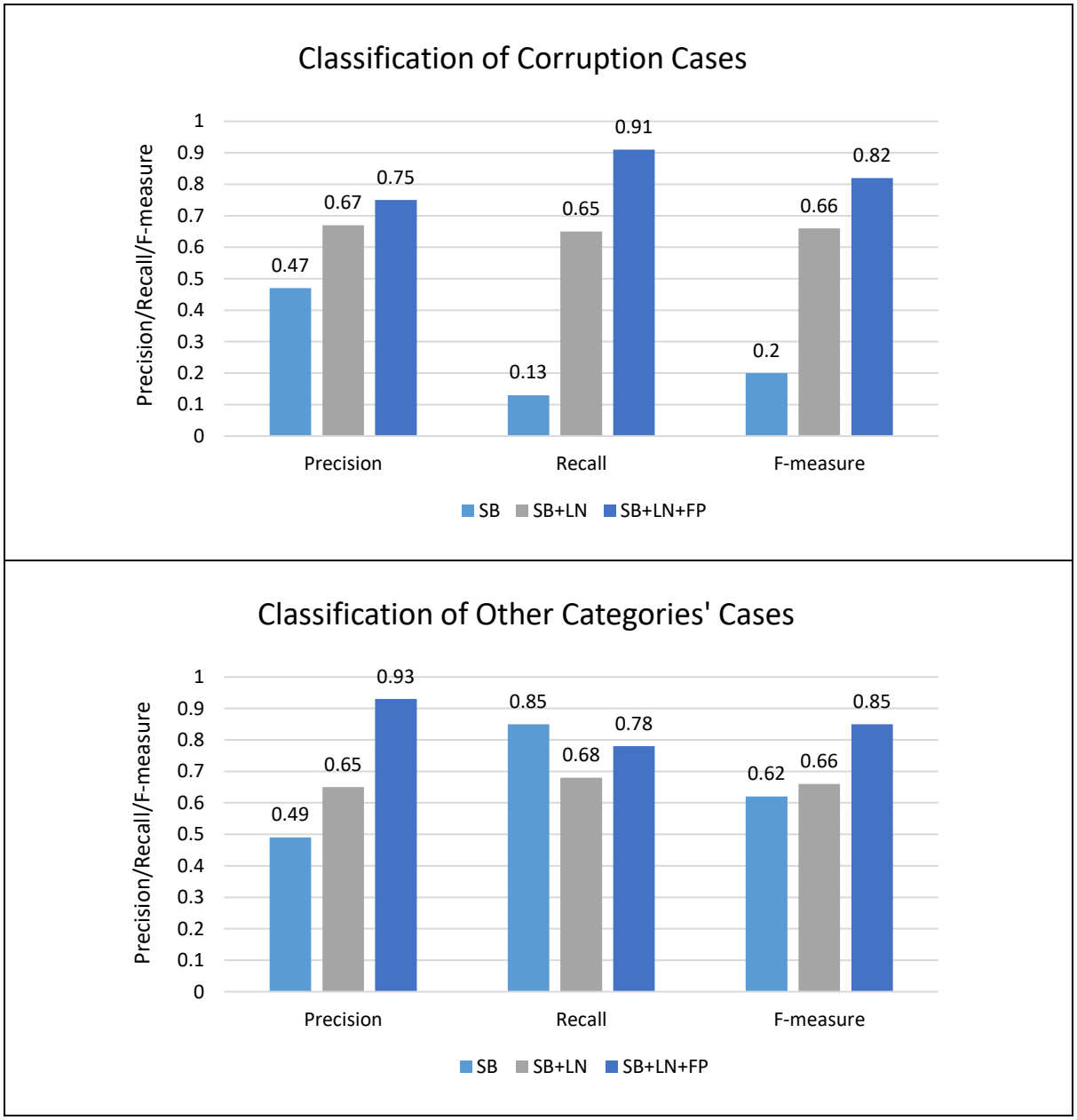

Figure 3. Precision, Recall and F-measure Values for Detecting Corruption Instances against Other Categories

Conclusion

We tackle a unique practical task of automatic classification of citizen feedback on government services provided in the form of SMS text written in Roman Urdu. Roman Urdu is an under-resourced language for which standard text processing and representation produces poor classification performance. We present a methodology for improving text classification in Roman Urdu through lexical normalization of terms to reduce dimensionality and discriminative feature pooling to enrich representation for classification. Our methodology exploits psycho-linguistic semantics for improvement and it is easy to apply in practice. Our experiments confirm that significant improvement in classification performance is achieved by using our methodology when compared with standard text classification techniques.

While we focus on Roman Urdu, the proposed methodology can be applied to other under-resourced languages as well. Furthermore, we believe there is tremendous potential for further research in representations and classification models for short informal texts in under-resourced languages.

References

Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. 2009. "Automatically Profiling the Author of an Anonymous Text," *Communications of the ACM* (52:2), pp. 119-123 (doi: 10.1145/1461928.1461959).

Bloehdorn, S., and Hotho, A. 2004. "Boosting for Text Classification with Semantic Features," in *Proceedings of the 6th International Workshop on Knowledge Discovery on the Web: Advances in Web Mining and Web Usage Analysis*, B. Liu, B. Masand, B. Mobasher and O. Nasraoui, (eds.), Berlin, Heidelberg: Springer-Verlag, pp. 149-166 (doi: 10.1007/11899402_10).

Gabrilovich, E., and Markovitch, S. 2005. "Feature Generation for Text Categorization using World Knowledge," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann Publishers Inc., pp. 1048-1053.

Han, B., and Baldwin, T. 2011. "Lexical Normalization of Short Text Messages: Makn Sens a #twitter," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto and R. Mihalcea (eds.), Stroudsburg, PA: Association for Computational Linguistics, pp. 368-378.

Junejo, K. N., and Karim, A. 2008. "A Robust Discriminative Term Weighting Based Linear Discriminant Method for Text Classification," in *Proceedings of 8th IEEE International Conference on Data Mining*, IEEE, pp. 323-332 (doi: 10.1109/ICDM.2008.26).

Junejo, K. N., Karim, A., Hassan, M. T., and Jeon, M. 2016. "Terms-Based Discriminative Information Space for Robust Text Classification," *Information Sciences* (372), pp. 518-538 (doi: 10.1016/j.ins.2016.08.073).

Liu, B. 2011. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, Berlin, Heidelberg: Springer-Verlag (doi: 10.1007/978-3-642-19460-3).

Liu, F., Weng, F., and Jiang, X. 2012. "A Broad-Coverage Normalization System for Social Media Language," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA: Association for Computational Linguistics, pp. 1035-1044.

Liu, F., Weng, F., Wang, B., and Liu, Y. 2011. "Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-Categorization nor Supervision," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto and R. Mihalcea (eds.), Stroudsburg, PA: Association for Computational Linguistics, pp. 71-76.

Liu, X., Zhang, S., Wei, F., and Zhou, M. 2011. "Recognizing Named Entities in Tweets," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto and R. Mihalcea (eds.), Stroudsburg, PA: Association for Computational Linguistics, pp. 359-367.

López, V., Fernández, A., García, S., Palade, V., and Herrera, F. 2013. "An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on using Data Intrinsic Characteristics," *Information Sciences* (250), pp. 113-141 (doi: 10.1016/j.ins.2013.07.007).

Ng, H. T., Goh, W. B., and Low, K. L. 1997. "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization," in *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, N. J. Belkin, F. Can, W. Hersh, A. D. Narasimhalu, P. Willett and E. Voorhees (eds.), New York, NY: Association for Computing Machinery, pp. 67-73.

Rafae, A., Qayyum, A., Uddin, M. M., Karim, A., Sajjad, H., and Kamiran, F. 2015. "An Unsupervised Method for Discovering Lexical Variations in Roman Urdu Informal Text," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, C. Callison-Burch, L. Màrquez and J. Su (eds.), Stroudsburg, PA: Association for Computational Linguistics, pp. 823-828 (doi: 10.18653/v1/D15-1097).

Scott, S., and Matwin, S. 1999. "Feature Engineering for Text Classification," in *Proceedings of the 16th International Conference on Machine Learning*, I. Bratko and S. Dzeroski (eds.), San Francisco, CA: Morgan Kaufmann Publishers Inc., pp. 379-388.

Sebastiani, F. 2005. "Text Categorization," in *Text Mining and Its Applications to Intelligence, CRM and Knowledge Management*, Southampton, UK: WIT Press, pp. 109-129.

Sharf, Z., and Rahman, S. U. 2017. "Lexical Normalization of Roman Urdu Text," *International Journal of Computer Science and Network Security* (17:12), pp. 213-221.

Tariq, A., and Karim, A. 2011. "Fast Supervised Feature Extraction by Term Discrimination Information Pooling," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, B. Berendt, A. de Vries, W. Fan, C. Macdonald, I. Ounis and I. Ruthven (eds.), New York, NY: Association for Computing Machinery, pp. 2233–2236 (doi: 10.1145/2063576.2063934).

Xue, Z., Yin, D., Davison, B. D., and Davison, B. 2011. "Normalizing Microtext," *Analyzing Microtext* (11:5).