Processing Informal, Romanized Pakistani Text Messages

Ann Irvine and **Jonathan Weese** and **Chris Callison-Burch**
Center for Language and Speech Processing
Johns Hopkins University

Abstract

Regardless of language, the standard character set for text messages (SMS) and many other social media platforms is the Roman alphabet. There are romanization conventions for some character sets, but they are used inconsistently in informal text, such as SMS. In this work, we convert informal, romanized Urdu messages into the native Arabic script and normalize non-standard SMS language. Doing so prepares the messages for existing downstream processing tools, such as machine translation, which are typically trained on well-formed, native script text. Our model combines information at the word and character levels, allowing it to handle out-of-vocabulary items. Compared with a baseline deterministic approach, our system reduces both word and character error rate by over 50%.

1 Introduction

There are many reasons why systematically 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 (MT) could supplement or replace such crowdsourcing efforts in the future. However, working with SMS data presents several challenges. First, messages may have non-standard spellings and abbreviations ("text speak"), which we need to *normalize* into standard language. Second, many languages that are typically written in a non-Roman script use a romanized version for SMS, which we need to *deromanize*. Normalizing and deromanizing SMS messages would allow us to use existing MT engines, which are typically trained on well-formed sentences written in their native-script, in order to translate the messages.

With this work, we use and release a corpus of 1 million $(4,195$ annotated) anonymized text messages sent in Pakistan[1]. We deromanize and normalize messages written in Urdu, although the general approach is language-independent. Using Mechanical Turk (MTurk), we collect normalized Arabic script annotations of romanized messages in order to both train and evaluate a Hidden Markov Model that automates the conversion. Our model drastically outperforms our baseline deterministic approach and its performance is comparable to the agreement between annotators.

2 Related Work

There is a strong thread of research dedicated to normalizing Twitter and SMS informal English (Sproat et al., 2001). Choudhury et al. (2007) use a supervised English SMS dataset and build a character-level HMM to normalize individual tokens. Aw et al. (2006) model the same task using a statistical MT system, making the output context-sensitive at the cost of including a character-level analysis. More recently, Han and Baldwin (2011) use unsupervised methods to build a pipeline that identifies ill-formed English SMS word tokens and builds a dictionary of their most likely normalized forms. Beaufort et al. (2010) use a large amount of training data to supervise an FST-based French SMS normalizer. Li and Yarowsky (2008) present methods that take advantage of monolingual distributional similarities to identify the full form of abbreviated Chinese words. One challenge in working with SMS data is that public data is sparse (Chen and Kan, 2011). Transliteration is well-studied (Knight and Graehl, 1997; Haizhou et al., 2004; Li et al., 2010) and is usually viewed as a subproblem of MT.

With this work, we release a corpus of SMS messages and attempt to normalize Urdu SMS texts. Doing so involves the same challenges as normalizing English SMS texts and has the added complexity that we must also deromanize, a process similar to the transliteration task.

[1] See http://www.cs.jhu.edu/~anni/papers/urduSMS/ for details about obtaining the corpus.

Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012), pages 75–78,
Montréal, Canada, June 7, 2012. ©2012 Association for Computational Linguistics

Original Message	**Vicky Kahan gaib ho tamam log? Lgta he parhai ho rhi he. Chalo shabash parh lo. MUBASHRA**
Language	Urdu
De-Romanization	کہاں غائب ہو تمام لوگ ، لگتا ہے پڑھائی ہو رہی ہے ، چلو شاباش پڑھ لو
English Translation	where are you people? seems everyone is studying. ok study its good

Figure 1: Example of SMS with MTurk annotations

3 Data and Annotation

Our Pakistani SMS dataset was provided by the Transnational Crisis Project, and it includes 1 million (724,999 unique) text messages that were sent in Pakistan just prior to the devastating July 2010 floods. The messages have been stripped of all metadata including sender, receiver, and timestamp. Messages are written in several languages, though most are in Urdu, English, or a combination of the two. Regardless of language, all messages are composed in the Roman alphabet. The dataset contains 348,701 word types, 49.5% of which are singletons.

We posted subsets of the SMS data to MTurk to perform language identification, followed by deromanization and normalization on Urdu messages. In the deromanization and normalization task, we asked MTurk workers to convert all romanized words into script Urdu and use full, non-abbreviated word forms. We applied standard techniques for eliminating noise in the annotation set (Callison-Burch and Dredze, 2010) and limited annotators to those in Pakistan. We also asked annotators to indicate if a message contained private, sensitive, or offensive material, and we removed such messages from our dataset.

We gathered deromanization and normalization MTurk annotations for 4,195 messages. In all experiments, we use 3,695 of our annotated SMS texts for training and 500 for testing. We found that 18% of word tokens and 44% of word types in the test data do not appear in the training data. An example of a fully annotated SMS is shown in Figure 1.

Figure 2 shows that, in general, productive MTurk annotators also tend to produce high quality annotations, as measured by an additional round of MTurk annotations which asked workers to choose the best annotation among the three we gathered. The raw average annotator agreements as measured by character and word level edit distance are 40.5 and 66.9, respectively. However, the average edit distances between 'good' MTurk workers (at least 50% of a worker's messages are voted best) and the deromanization which was voted best (when the two are different) are 25.1 (character) and 53.7 (word).

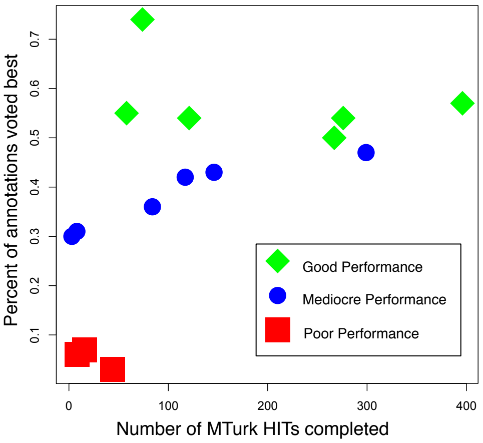

Figure 2: Productivity vs. percent of annotations voted best among three deromanizations gathered on MTurk.

Script	Romanizations [Frequency]	English
کرم	kram [3] karam [3] karm [2] karem [1]	grace
کونسا	konsa [6] kn sa [4] knsa [3] kon sa [2]	which
دوسروں	dusra [1] **2ro** [1] dosaro [1] dusron [1]	other people
خوش	khush [32] ki [1] khus [1]	happy
جگنو	jugno [1] ganjo [1]	firefly
باتیں	batein [7] baten [7] baatein [4] batain [4] btein [3]	chit-chat

Figure 3: Urdu words romanized in multiple ways. The Urdu word for "2" is pronounced approximately "du."

We used an automatic aligner to align the words in each Arabic script annotation to words in the original romanized message. The alignments show an average fertility of 1.04 script words per romanized word. Almost all alignments are one-to-one and monotonic. Since there is no reordering, the alignment is a simplified case of word alignment in MT.

Using the aligned dataset, we examine how Urdu words are romanized. The average entropy for non-singleton script word tokens is 1.49 bits. This means it is common for script words to be romanized in multiple ways (4.2 romanizations per script word on average). Figure 3 shows some examples.

4 Deromanization and Normalization

In order to deromanize and normalize Urdu SMS texts in a single step, we use a Hidden Markov Model (HMM), shown in Figure 4. To estimate the probability that one native-script word follows an-

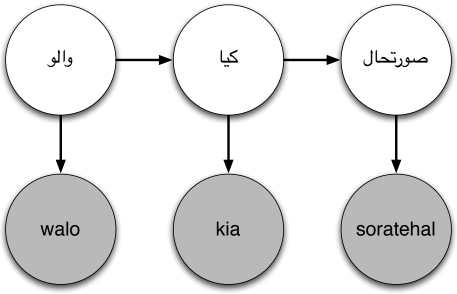

Figure 4: Illustration of HMM with an example from SMS data. English translation: "What's the situation?"

other, we use a bigram language model (LM) with add-1 smoothing (Lidstone, 1920) and compare two sources of LM training data.

We use two sources of data to estimate the probability of a romanized word given a script word: (1) a dictionary of candidates generated from automatically aligned training data, (2) a character-based transliteration model (Irvine et al., 2010).

If r is a romanized word and u is a script Urdu word, the dictionary-based distribution, $p_{\text{DICT}}(r|u)$, is given by relative frequency estimations over the aligned training data, and the transliteration-based distribution, $p_{\text{TRANS}}(r|u)$, is defined by the transliteration model scores. We define the model's emission probability distribution as the linear interpolation of these two distributions:

$$p_e(r|u) = (1-\alpha)p_{\text{DICT}}(r|u) + \alpha p_{\text{TRANS}}(r|u)$$

When $\alpha = 0$, the model uses only the dictionary, and when $\alpha = 1$ only the transliterations.

Intuitively, we want the dictionary-based model to memorize patterns like abbreviations in the training data and then let the transliterator take over when a romanized word is out-of-vocabulary (OOV).

5 Results and discussion

In the eight experiments summarized in Table 1, we vary the following: (1) whether we estimate HMM emissions from the dictionary, the transliterator, or both (i.e., we vary α), (2) language model training data, and (3) transliteration model training data.

Our baseline uses an Urdu-extension of the Buckwalter Arabic deterministic transliteration map. Even our worst-performing configuration outperforms this baseline by a large margin, and the best configuration has a performance comparable to the agreement among good MTurk workers.

	LM	Translit	α	CER	WER
1	News	—	Dict	41.5	63.3
2	SMS	—	Dict	38.2	57.1
3	SMS	Eng	Translit	33.4	76.2
4	SMS	SMS	Translit	33.3	74.1
5	News	SMS	Both	29.0	58.1
6	News	Eng	Both	28.4	57.2
7	SMS	SMS	Both	25.0	50.1
8	SMS	Eng	Both	**24.4**	**49.5**
Baseline: Buckwalter Determ.				64.6	99.9
Good MTurk Annotator Agreement				25.1	53.7

Table 1: Deromanization and normalization results on 500 SMS test set. Evaluation is by character (CER) and word error rate (WER); lower scores are better. "LM" indicates the data used to estimate the language model probabilities: News refers to Urdu news corpus and SMS to deromanized side of our SMS training data. "Translit" column refers to the training data that was used to train the transliterator: SMS; SMS training data; Eng; English-Urdu transliterations. α refers to the data used to estimate emissions: transliterations, dictionary entries, or both.

Unsurprisingly, using the dictionary only (Experiments 1-2) performs better than using transliterations only (Experiments 3-4) in terms of word error rate, and the opposite is true in terms of character error rate. Using *both* the dictionary derived from the SMS training data and the transliterator (Experiments 5–8) outperforms using only one or the other (1–4). This confirms our intuition that using transliteration to account for OOVs in combination with word-level learning from the training data is a good strategy[2].

We compare results using two language model training corpora: (1) the Urdu script side of our SMS MTurk data, and (2) the Urdu side of an Urdu-English parallel corpus,[3] which contains news-domain text. We see that using the SMS MTurk data (7–8) outperforms the news text (5–6). This is due to the fact that the news text is out of domain with respect to the content of SMS texts. In future work, we plan to mine Urdu script blog and chat data, which may be closer in domain to the SMS texts, providing better language modeling probabilities.

[2] We experimented with different α values on held out data and found its value did not impact system performance significantly unless it was set to 0 or 1, ignoring the transliterations or dictionary. We set $\alpha = 0.5$ for the rest of the experiments.

[3] LDC2006E110

Training Freq. bins			Length Diff. bins		
Bin	CER	WER	Bin	CER	WER
100+	9.8	14.8	0	23.5	43.3
10–99	15.2	22.1	1, 2	29.1	48.7
1–9	27.5	37.2	-1, -2	42.3	70.1
0	73.5	96.6	$\geq$3	100.3	100.0
			$\leq$-3	66.4	87.3

Table 2: Results on *tokens* in the test set, binned by training frequency or difference in character length with their reference. Length differences are number of characters in romanized token minus the number of characters in its deromanization. $\alpha = 0.5$ for all.

We compare using a transliterator trained on romanized/deromanized word pairs extracted from the SMS text training data with a transliterator trained on *English* words paired with their Urdu transliterations and find that performance is nearly equivalent. The former dataset is noisy, small, and in-domain while the latter is clean, large, and out-of-domain. We expect that the SMS word pairs based transliterator would outperform the English-Urdu trained transliterator given more, cleaner data.

To understand in more detail when our system does well and when it does not, we performed additional experiments on the token level. That is, instead of deromanizing and normalizing entire SMS messages, we take a close look at the kinds of romanized word tokens that the system gets right and wrong. We bin test set word tokens by their frequencies in the training data and by the difference between their length (in characters) and the length of their reference deromanization. Results are given in Table 2. Not surprisingly, the system performs better on tokens that it has seen many times in the training data than on tokens it has never seen. It does not perform perfectly on high frequency items because the entropy of many romanized word types is high. The system also performs best on romanized word types that have a similar length to their deromanized forms. This suggests that the system is more successful at the deromanization task than the normalization task, where lengths are more likely to vary substantially due to SMS abbreviations.

6 Summary

We have defined a new task: deromanizing and normalizing SMS messages written in non-native Roman script. We have introduced a unique new annotated dataset that allows exploration of informal text for a low resource language.

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of COLING/ACL*.

Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing SMS messages. In *Proceedings of ACL*.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *NAACL-NLT Workshop on Creating Speech and Language Data With Mechanical Turk*.

Tao Chen and Min-Yen Kan. 2011. Creating a live, public short message service corpus: The NUS SMS corpus. *Computation and Language*, abs/1112.2468.

Monojit Choudhury, Vijit Jain Rahul Saraf, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. In *International Journal on Document Analysis and Recognition*.

Li Haizhou, Zhang Min, and Su Jian. 2004. A joint source-channel model for machine transliteration. In *Proceedings of ACL*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of ACL*.

Ann Irvine, Chris Callison-Burch, and Alexandre Klementiev. 2010. Transliterating from all languages. In *Proceedings of the Association for Machine Translation in the America*, AMTA '10.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proceedings of ACL*.

Zhifei Li and David Yarowsky. 2008. Unsupervised translation induction for Chinese abbreviations using monolingual corpora. In *Proceedings of ACL/HLT*.

Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouchine. 2010. Report of NEWS 2010 transliteration generation shared task. In *Proceedings of the ACL Named Entities WorkShop*.

George James Lidstone. 1920. Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192.

Richard Sproat, Alan W. Black, Stanley F. Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, pages 287–333.