English-Hindi Transliteration using Multiple Similarity Metrics

Niraj Aswani and Robert Gaizauskas
n.aswani@dcs.shef.ac.uk r.gaizauskas@dcs.shef.ac.uk
University of Sheffield

Abstract

In this paper, we present an approach to measure the transliteration similarity of English-Hindi word pairs. Our approach has two components. First we propose a bi-directional mapping between one or more characters in the Devanagari script and one or more characters in the Roman script (pronounced as in English). This allows a given Hindi word written in Devanagari to be transliterated into the Roman script and vice-versa. Second, we present an algorithm for computing a similarity measure that is a variant of Dice's coefficient measure and the LCSR measure and which also takes into account the constraints needed to match English-Hindi transliterated words. Finally, by evaluating various similarity metrics individually and together under a multiple measure agreement scenario, we show that it is possible to achieve a 0.92 f-measure in identifying English-Hindi word pairs that are transliterations. In order to assess the portability of our approach to other similar languages we adapt our system to the Gujarati language.

1 Introduction

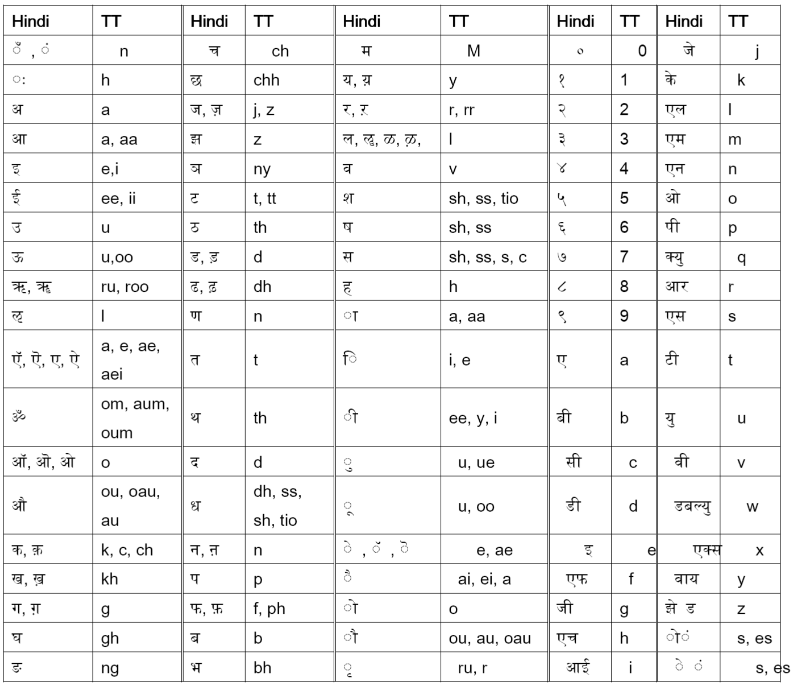
In India, English is one of the most widespread foreign languages. It is becoming very common for people to use both English and Hindi vocabulary in the same sentence (Clair, 2002). When writing such mixed code sentences, people use the Devanagari script to write Hindi words and use equivalent Hindi transliterations for the English words. There are many words in Hindi which have been derived from the English (e.g. *School, Doctor*) and vice-versa (e.g. *Bungalow*). Apart from such cognates, named entities such as person names, names of places, organizations are other types of words that need transcribing into another writing system (Kondrak et al., 2003). We estimate that in the EMILLE English-Hindi corpora approximately 7% of words are transliterations.

In this paper, we present a Transliteration Similarity metric (TSM) that is based on the letter correspondences between the writing systems of the English and the Hindi languages. It is a part of our effort to develop a general framework for text alignment (Aswani and Gaizauskas, to appear) where it is currently used in an English-Hindi word alignment system for aligning words such as proper names and cognates. We give a mapping for one or more characters in the Devanagari script into one or more characters in the Roman script. Given a Hindi word, this mapping allows one or more candidate transliterated forms in the Roman script to be obtained. To choose which of these candidates most closely matches a candidate target word requires a string similarity measure. We review some of the well known string similarity metrics and propose an algorithm for computing a similarity measure. We evaluate the performance of these similarity metrics individually and in various combinations to discover the best combination of similarity metrics and a threshold value that can be used to maintain the optimal balance between accuracy and coverage. To test the portability of our approach to other similar languages we adapt our system to the Gujarati language.

2 Our Approach

Figure 1 lists letter correspondences between the writing systems of the two languages where one or more Hindi characters are associated with one or more English characters. For example [f] can be f, or ph (e.g. frame, photo). This transliteration mapping (TM) was derived manually and provides a two way lookup facility. The following illustration explains how to use the TM to obtain possible transliterations for the Hindi word [kensar] which means *cancer* in English. For Hindi letter at the i^{th} position in the Hindi word HW (where $i = 1..n$ and $n = |HW|$ (i.e. the length of HW)), we define a set TS_i that contains all possible phonetic mappings for that letter.

In order to optimize the process, we remove from the TS_i all mapped characters that do not exist in the candidate target string. Below, we list mappings for the letters of the word [kensar]. The mappings which need to be removed from the TS_i are enclosed in round brackets: [k] = [c, (k), (ch)]; [e] = [e, a, (ai)]; [n] = [n]; [s] = [c,(s)]; [r] = [r]. From these mappings we define a set TS of n-tuples such that $TS = TS_1 \times TS_2 \times ... \times TS_n$ (i.e. TS is a Cartesian product of all the previously defined sets ($TS_{i=1..n}$) for each letter in the Hindi word). Each n-tuple in TS is one possible transliteration of the original Hindi word. In total there are $|TS|$ transliterated strings. In the above example the value of $|TS|$ is 2 (1 x 2 x



Hindi	TT	Hindi	TT	Hindi	TT	Hindi	TT	Hindi	TT
◌ँ , ◌ं	n	च	ch	म	M	०	0	जे	j
◌ः	h	छ	chh	य, य़	y	१	1	के	k
अ	a	ज, ज़	j, z	र, ऱ	r, rr	२	2	एल	l
आ	a, aa	झ	z	ल, ल्ह, ळ, ऴ,	l	३	3	एम	m
इ	e,i	ञ	ny	व	v	४	4	एन	n
ई	ee, ii	ट	t, tt	श	sh, ss, tio	५	5	ओ	o
उ	u	ठ	th	ष	sh, ss	६	6	पी	p
ऊ	u,oo	ड, ड़	d	स	sh, ss, s, c	७	7	क्यु	q
ऋ, ॠ	ru, roo	ढ, ढ़	dh	ह	h	८	8	आर	r
ऌ	l	ण	n	◌ा	a, aa	९	9	एस	s
ऍ, ऎ, ए, ऐ	a, e, ae, aei	त	t	◌ि	i, e	ए	a	टी	t
ॐ	om, aum, oum	थ	th	◌ी	ee, y, i	बी	b	यु	u
ऑ, ऒ, ओ	o	द	d	◌ु	u, ue	सी	c	वी	v
औ	ou, oau, au	ध	dh, ss, sh, tio	◌ू	u, oo	डी	d	डबल्यु	w
क, क़	k, c, ch	न, ऩ	n	◌े , ◌ॅ , ◌ॆ	e, ae	इ	e	एक्स	x
ख, ख़	kh	प	p	◌ै	ai, ei, a	एफ	f	वाय	y
ग, ग़	g	फ, फ़	f, ph	◌ो	o	जी	g	झे ड	z
घ	gh	ब	b	◌ौ	ou, au, oau	एच	h	◌ो◌ं	s, es
ङ	ng	भ	bh	◌ृ	ru, r	आई	i	◌े ◌ं	s, es

Figure 1: English-Hindi Transliteration mapping

1 x 1 x 1) (i.e. Cencr and Cancr). Each transliterated string ($S_{j=1..|TS|\in TS}$) is compared with the English word using one of the string similarity metrics (explained in the next section). If the English word and any of the transliterated strings has a similarity score above a specified threshold, the strings are deemed to be transliterations.

3 String Similarity Metrics

Given a pair of strings, string similarity metrics give us a measure of how similar the two strings are. The matching coefficient is the simplest measure of all, where only the count of characters that match is taken as the similarity measure. There are different variants of the matching coefficient such as Dice's coefficient (DC) and overlap coefficient. However, for these measures the number of matching characters is more important than positions of the characters. In case of LCSR (Longest Common Subsequence Ratio) and n-gram similarity metrics, the position of characters is also taken into consideration. The Levenshtein distance measure (LD) (Levenshtein, 1966) is used for calculating the minimum number of edit operations needed to transform one string into an other. In case of the Jaro-Winkler metric (JW) (Jaro; Winkler, 1999), characters in the source string need to match within a window (calculated from the lengths of the two strings) of corresponding indices in the target string. The Soundex metric (Russell, 1918, 1922) groups consonants according to their sound similarity and ignores vowels unless they appear at the start of the strings.

In the case of English-Hindi strings it was observed during our experiments that for the two strings to be similar the first and the last characters from both the strings - the English word (E) and the transliterated string (T), must match. This ensures that the words have same phonetic starting and same ending. However some English words start or end with silent vowels (e.g. *p* in *psychology* and *e* in *programme*). Therefore in such cases the first character of the transliterated string should be compared with second character of the E and similarly the last character of the transliterated string should be compared with the second last character of the E. Our experiments show that unless the length of the shorter string is at least 65% of the length of the other string, they are unlikely to be phonetically similar.

The similarity algorithm takes two strings, E and S, as input where $E_{i=1..n}$ and $T_{j=1..m}$ refer to characters at position i and position j in the two strings with lengths n and m respectively. Starting with $i = 1$ and $j = 1$, character E_i is compared with characters T_j, T_{j+1} and T_{j+2}. If E_i matches with one of the T_j, T_{j+1} and T_{j+2}, the pointer i advances one position and the pointer j is set to one position after the letter that matches with E_i. If there is no match, the pointer i advances and j does not. We award every match a score of 2 and calculate similarity using the matchScore/(f(s) + f(t)) where f(x) = number of letters in the x string.

4 Experiments

We compare our similarity metric TSM with other string similarity metrics such as the standard DC metric, LSCR metric, JW metric, n-gram metric and LD metric. In order to perform this comparison we manually obtained 1000 unique words pairs from the EMILLE corpus. Out of the 1000 words pairs collected, 732 pairs were correct transliterations of each other and 268 pairs were not. We obtained a set of transliterations (using the TMs) for each Hindi word in the collected sample data. For each similarity metric the task was to identify correct transliteration pairs and avoid recognizing incorrect pairs by giving them a very low similarity score. The following procedure was repeated for each similarity metric. For each Hindi word in these test pairs we obtained a transliteration with highest score. Then, we clustered the results in six predefined groups: >= 0.95, >= 0.90, >= 0.85, >= 0.80, >= 0.75, and >= 0.70 where, the group >= *Sim* contains pairs with similarity greater than or equal to *Sim*. For each group, we calculated the precision, recall and f-measure.

From this, we were able to obtain the best threshold value for each of the similarity metrics. For example, in case of the DC metric, the best f-measure score (0.77) was recorded when the threshold value was >= 0.80. Similarly, for TSM, LCSR, JW, n-gram, and LD, the recorded f-measure and threshold values were (0.77,>=0.75), (0.67,>=0.7), (0.73,>=0.7), (0.39,>=0.7) and (0.74,>=0.7) respectively. It must be noted that the DC metric does not take positions of characters into account where as the TSM does. Although the f-measure figures for these two metrics are same, this is because our dataset does not have examples such as *teacher* vs [cheater] which according to the DC is a correct transliteration pair (even with threashold set to 100).

Similarity Metrics	Threshold	F-Measure
DC + TS + JW + LD	>= 70	0.84
DC + TSM + JW	>= 0.75	0.85
DC + TSM + JW	>= 0.78	0.86
DC + TSM + JW	>= 0.79	**0.92**
DC + TSM + JW	>= 0.80	**0.92**
DC + TSM + JW	>= 0.81	**0.91**
DC + TSM + JW	>= 0.85	0.78
DC + TSM + LCSR	>= 0.90	0.62
DC + LCSR + JW	>= 0.95	0.4

Table 1: Multiple Measure Agreement Strategy Results

Given the different criteria that these similarity metrics work on, it is possible that given a pair of strings one metric gives it a very high score where as the others very low. In order to exploit the multiple measure agreement strategy[1], we conducted a further experiment, whereby we recorded top combination of metrics that performed best (f-measure) given different threshold values. We found that the combination of DC, TSM and the JW metrics works best with threshold value set between 0.79 and 0.81.

Although the scripts used by Gujarati and Hindi are different, the consonants and vowels in their scripts are similar and pronounced the same way. With the help of a native Gujarati speaker, we replaced the Hindi letters in our mappings table with their corresponding Gujarati letters. Using the same combination of similarity metrics and the same threshold, we obtained 0.91 f-measure on the Gujarati test data.

References

N. Aswani and R. Gaizauskas. Evolving a general framework for text alignment: Case studies with two south asian languages. In *Proceedings of the International Conference on Machine Translation: Twenty-Five Years On*, Cranfield, Bedfordshire, UK, November to appear.

St. R.N. Clair. Managing multilingualism in india: Political and linguistic manifestations. volume 26, pages 336–339. John Benjamins Publishing Company, 2002.

[1] A word pair is a valid transliterated pair only if it receives majority vote from the members of the similarity metrics group.

M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa. In *Journal of the American Statistical Association*, volume 84, pages 414–420. Florida.

G. Kondrak, D. Marcu, and K. Knight. Cognates can improve statistical machine translation models. In *Human Language Technology (NAACL)*, 2003.

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Proceedings of the Soviet Physics Doklady 10*, pages 707–710. 1966.

R. C. Russell. Index. US1.261.167, April 1918.

R. C. Russell. Index. US1.435.663, November 1922.

W. E. Winkler. The state of record linkage and current research problems. In *Statistics of Income Division*. Internal Revenue Service Publication R99/04, 1999.