Study to Evaluate Available Solutions for the Submission and Display of Internationalized Contact Data

June 2, 2014

Table of Contents

Executive Summary

This study documents and evaluates the potential solutions for submitting or displaying contact data in non-ASCII (American Standard Code for Information Interchange) character sets. It aims to help the ICANN community, who is investigating the possibility of transliteration, transcription or translation (collectively transformation) of internationalized registration data for its broader accessibility. The study looks at the current practices of handling internationalized contact data by e-merchants, registries and registrars. It also determines the support of such data by relevant protocols. Finally it assesses the accuracy implications for transforming internationalized contact data.

The study recommends specialized terminology to be used in this context, as proposed by United Nations (UN), and identifies at least three levels of transformation: accurate (for legal purposes; only manually possible), consistent (for searching; possible through tools, but requires standards) and ad hoc (for arbitrary representation; possible through tools without standard specification). For broadest access, transformation from any language to any other language should be enabled. However, due to practical limitations of developing such a large number of transformations, UN recommends formulating reversible romanization[1] for each language, which allows transforming any language to any other language by pivoting through the roman form. If standard language level transforms are not available, fall back to script and ad hoc options may be used, as suggested by the Unicode consortium.

The survey of e-commerce websites shows that some allow data in local languages but the verification of this contact data provided by the user is very limited and only for a subset of fields. Most websites accept the user input, putting the onus of valid input on the user or the other parties involved.

The registry and registrar survey indicates that multiple versions of the registrant data in different languages is not widely supported. No respondent is currently doing transformation of registrant data.

The protocol infrastructure needed to support internationalized registration data, such as EPP support, language/script tagging, is also not widely used. Analysis of WHOIS, EPP and RDAP protocols shows that earlier protocols have had none or limited support for internationalized data. However, the more recent work is supporting non-Latin scripts. These protocols are missing any scheme to record the source of the data and its transformation history. Moreover, there is very limited support of entries of multiple versions (such as in multiple scripts, forms, or Romanized) of the same data from the user.

There are multiple tools which can be used for transformation. These tools can be divided into three classes: general conversion (transliteration and transcription) tools, general translation tools and specialized tools which focus on name or address conversions. A subset of these tools are tested on four different kinds of writing systems (with multiple languages) to assess the challenges and quality of transformation across multiple kinds of scripts, including Logographic Han (simplified and traditional Chinese), Abugida Devanagari (Hindi and Marathi), Abjad Arabic (Arabic, Persian and Urdu) and Alphabetic Cyrillic (Russian, Bulgarian and Ukrainian). Within each language data elements are tested for each of the following categories of contact data: Name (person or organization), Address, City/State

[1] See Section 3.1 for terminology and its definition.

and Country. Two accuracy measures are taken to analyze the results: exact match between manually generated and transformed data (conservative comparison; higher score is better; best at 100%) and a more detailed Levenshtein distance to measure the string similarity even if the manual and transformed results are different (same strings have a distance of 0%; lower score is better). The results show that translation tools (66% accuracy) generally perform much better than conversion (including specialized) tools (16% accuracy) averaged across all data categories. The translation tools, though more accurate, differ in quality across various languages and err in determining when meaningful words are to be translated versus just transliterated. Conversion tools cannot translate the common nouns in addresses and place names and cannot deal with word order differences between source and target languages, causing significant errors, not meeting user expectations. Within the data categories, the addresses are the most difficult to transform, as they are longer, require re-ordering, and need a mix of translation and conversion as they contain both common and proper nouns, mixed with digits. Thus, they show low accuracy (Levenshtein distance of 55%). Country names show similar level of inaccuracy (Levenshtein distance of 56%), because they are arbitrarily different across languages. Names and City/State are more consistent across languages and thus show better transformation results (Levenshtein distance of as low as 29% and 30%). Accuracy across languages varies arbitrarily, based on the level of maturity of tools. Accuracy across scripts depends on the types of scripts and type of transformation, e.g. conversion of Arabic script to its romanized form is very inaccurate as the former does fully specify vowels, but translation can be accurate as it does not depend on script level differences. Finally reversibility of the transliteration tools is accurate only if the form (being reverse transformed) has been generated through the same standard process (however, in this case the romanized form may be complex, with diacritics, and hard to read by general users). Reversibility of manually generated or otherwise arbitrarily different form is not accurate. Reversibility is also less accurate with translation, as it undergoes the process twice. This implies that transformation from one language to any other language, though theoretically possible, is practically even less accurate. Finally, if the internationalized contact data is to be transformed, it may be necessary to store additional information related to its language, script, process of generation, etc. for effective use. And doing it consistently may require a significant coordination effort between the stakeholders involved in the transformation process to use same standards and mechanisms. A summary of findings is given in Section 8.

In summary, the study has found that provisioning and querying protocols are lacking either support or deployment for internationalized registration data, and that none of the tools tested is providing a high level of accuracy and consistency in its transformation of internationalized registration data.

1 Background

ICANN needs to define requirements for internationalized registration data, and the registrars and registry operators need to deploy systems and processes when dealing with submission, storage, transmission and display of internationalized registration data. Currently there are multiple working groups within ICANN community actively seeking these answers. The purpose of this study is to document current practices and transformation possibilities for internationalized contact data to inform the community. The study has the following scope of work.

1. The main aim of the study is to assess the accuracy implications for transforming internationalized contact data. The study will consider tools (i) transliterating internationalized contact information to ASCII, (ii) translating internationalized contact information to English, (iii) transcribing internationalized contact information to ASCII, or (iv) a combination of these techniques.
2. However, before going into details of transformation of contact data, the study will also look into practices of handling internationalized contact data in two cases:
    a. Electronic merchants and online service providers in other industries often have to accommodate submission or display of their content in multiple languages. The current study will look into how their websites handle internationalized contact information.
    b. Both registries and registrars already operate in geographies where local languages require use of character sets beyond ASCII in Latin or non-Latin writing systems. The study will survey the submission and display practices of internationalized registration contact information for such registries and registrars. The study will also look at what constraints current relevant protocols may also place on submission, storage, transmission and display of such data by registries and registrars.

2 Introduction

Domain Name Registration Data (DNRD) has many data elements which pertain to the registrant, the registrar and other information related to the transaction. The elements are of multiple types, including names of people, names of organizations, addresses (including street addresses, city and country names), phone numbers, email addresses, dates, IDs, status , domain names, etc. (see SAC054 (2012) for a complete list and detailed analysis of these elements). These elements can be divided into **contact elements** (names, addresses, phone nos., email addresses, domain names etc.) and **transaction elements** (IDs, Status flags, transaction dates, etc.). The transaction elements are largely added by the system and can be automated for generation in multiple scripts and languages (if needed). The contact elements pertain to the registrant and the registrar. Those related to the registrar (registrar name, registrar address, etc.) are also limited and largely fixed and can be made available in multiple languages and scripts without significant effort. However, the registrant centric contact information is highly ad hoc and would be a challenge to provide in a multi-lingual environment. This includes name, organization, street address, city, state, country, phone, fax and email.

A registrant for an internationalized domain name may be monolingual. Even if the registrant is not monolingual, there may still be reasons (of accuracy, legal requirements, etc.) which may still require this information in local languages and scripts. However, if such information is acquired in a local language only, it becomes incomprehensible for the internet users who speak another language and/or use another script, even if such information is publicly available. The purpose of Domain Name Registration Data (DNRD, originally WHOIS Data) has been to make such information publicly accessible. Language and script can become barriers to such accessibility, and this data acquired in the local language and script of the registrant would need to be presented in the local language and script of the end-user to address the challenge. The current study looks into the practices, standards and tools for transformation of internationalized DNRD for this purpose.

3 Requirement for Transformed Names

This study looks into the use of three methods of transforming the contact information in the DNRD into another language or script: translation, transliteration and transcription. The transformed data would need to be accurate and available in a variety of languages and scripts, as explained in more detail below.

Of the DNRD elements, enabling access for person names, organization names, addresses, cities and country names is challenging for global users. Additional fields, which include phone numbers, email addresses and domain names, can either be mapped across scripts and languages (e.g. digits in a phone number) or are script-bound (e.g. email address and domain names) and therefore cannot be made available in multiple scripts.

3.1 Terminology and Associated Definitions

For further discussion some relevant terminology is introduced in this section, based on the work by United Nations Group of Experts on Geographical Names (UNGEGN, 2002).

Proper Name or **Proper Noun** is a word that uniquely identifies an individual person, place or thing.

Common Noun is a word designating any one of a particular kind of being, place or thing. Examples: park, rue, center.

Toponym or **Place Name** is a proper noun applied to a topographic feature; a comprehensive term for geographical names and extraterrestrial names.

Allonym or **Alternate Name** or **Variant Name** is each of two or more toponyms employed in reference to a single topographic feature. Examples: Hull, Kingston upon Hull; Vesterhavet, Nordsee; Swansea, Abertawe; Johannesburg, Egoli.

Exonym is the name used in a specific language for a geographical feature situated outside the area where that language has official status, and differing in its form from the name used in the official language or languages of the area where the geographical feature is situated. Examples: Warsaw is the English exonym for Warszawa; Londres is French for London; Mailand is German for Milano. The officially romanized endonym Moskva for

Москва is not an exonym, nor is the Pinyin form Beijing, while Peking is an exonym. ... *The United Nations recommends minimizing the use of exonyms in international usage* (emphasis added).

Traditional name is an exonym in relatively widespread use by a particular linguistic community and usually found in its tradition and literature. Examples: Alexandrie (French) for al-Iskandarīyah (Arabic); Jerusalen (Spanish) for Yerushalayim (Hebrew); Peking (English) for Beijing (Chinese).

Endonym is the name of a geographical feature in one of the languages occurring in that area where the feature is situated. Examples: Vārānasī (not Benares); Aachen (not Aix-la-Chapelle); Krung Thep (not Bangkok); al-Uqşur (not Luxor); Teverya (not Tiberias).

Generic Term is a common noun that describes a topographic feature in terms of its characteristics and not by its proper name. Examples: mountain, sierra, san, shan, dagh, jabal, har, river, wadi, gang. It may form part of a toponym and called **Generic Element**.

In addition, UNGEGN also identifies some processes (UNGEGN 2002).

Name Transformation is a process in toponymy, general term covering the translation, transcription and transliteration of toponyms. The two latter terms constitute conversion.

Conversion is the process of transferring the phonological and/or morphological elements of a particular language to another, or from one script to another. Conversion is effected by either transcription or transliteration.

Translation is (a) The process of expressing meaning, presented in a source language, in the words of a target language.

(b) A result of this process. It is sometimes applied only to the generic element of a name. Examples: Mer Noire (French for Russian Čornoje More); Casablanca (Spanish for Arabic Dār al-Bay.dā'); Lake Como (English for Italian Lago di Como); Mount Fuji (English for Japanese Fuji San).

Transcription is (a) A method of phonetic names conversion between different languages, in which the sounds of a source language are recorded in terms of a specific target language and its particular script, normally without recourse to additional diacritics. The reverse process is called **Retranscription**.

(b) A result of this process. Examples: Turkish Ankara Greek Αγκαρα; Russian Щукино English Shchukino; Arabic جبلية French Djabaliya.

Transcription is not normally a reversible process. Retranscription (e.g. by computer) might result in a form differing from the original, for example in the above cases in Turkish Agkara, Russian Шчукино, Arabic دجبلية.

However, Pinyin romanization of Chinese, although being a conversion between scripts, but being phonetic and non-reversible, is also regarded as transcription and not as transliteration.

Transliteration is (a) A method of names conversion between different alphabetic scripts and syllabic scripts, in which each character or di-, tri- and tetragraph of the source script is represented in the target script in principle by one character or di-, tri- or tetragraph, or a diacritic, or a combination of these. Transliteration, as distinct from transcription, aims at (but does not necessarily achieve) complete reversibility, and must be accompanied by a transliteration key. The reverse process is called **Retransliteration**.

(b) A result of this process. Examples (with English exonyms in parentheses): القاهرة al-Qāhirah (Cairo); Владивосток Vladivostok; חיפה Hefa (Haifa); አዲስ አበባ Adis Abeba (Addis Ababa).

Reversibility is a characteristic of transliteration that permits a written item to be converted from one script or writing system into another, and subsequently to be reconverted back into the source script, the result being identical with the original.

Romanization is conversion from non-Roman into Roman script. Examples: Αθήνα Athina; Москва Moskva; بيروت Bayrūt; תל-אביב Tel-Aviv; 二ホン Nihon.

3.2 Information Accuracy for Effective Use

There are at least three kinds of use the transformed contact data in the DNRD may have in another language or script (based on the level of accuracy of the transformation):

1. Requiring **accurate transformation** (e.g. valid in a court of law, matching information in a passport, matching information in legal incorporation, etc.)
2. Requiring **consistent transformation** (allowing use of such information to match other information provided in another context, e.g. to match address information of a registrant on a Google map, etc.)
3. Requiring **ad hoc transformation** (allowing informal or casual version of the information in another language to provide more general accessibility)

Each use imposes a different set of requirements on the transformation from source language to target language. Accurate transformation of at least some proper names may require manual process because even if transformation is 100% accurate, in many cases names have allonyms (or alternate names or spellings) and the selection from among them is normally an arbitrary choice and for that reason will need to be verified by individual registrant.

For example, when transforming names from Chinese into English, the same Chinese character 金 is transformed into different English letters according to the origin of the person (Huang 2005): **Jin** Renqing (China), **Kim** Dae-jung (Korea), Martin Luther **King** (USA), **Kane**maru Shin (Japan) and Jose Joa**quin** Brunner (Chile). Similarly due to inherent ambiguity in the writing system of Arabic, as the diacritical marks are not normally written, the word حسن as a person name, can be interpreted as حَسَن to

name a male or as حُسْن to name a female (Alkharashi 2009). This will result in different transformations, which include at least *Hasan* and *Hassan* for the male name and *Husn* and *Hosn* for the female name. Manual intervention is necessary to accurately capture such variations.

Further challenges may be introduced by having a mix of proper and common nouns within a name. In such cases, sometimes the generic terms are translated whereas in other cases they need to be transcribed. For example, when "port" is used in English as a source language, in the target language it is not translated but transcribed for *Newport* but may be translated for *Port of Houston*.

In summary, if an accurate transformation of the name is needed, a manual verification may be required, which may require knowledge of geography, registrant origin and gender, and similar other cultural conventions and other world knowledge. To some extent this can be done independently of the registrant. However, sometimes there are multiple allonyms and the exact choice is arbitrary and can only be determined after consultation of the registrant.

Transliteration or transcription processes would not be accurate but could give consistent transformation. Transcription is closer in pronunciation and understandable by human users but is generally not reversible, allowing inconsistent retransctiption back into the source language. On the other hand, transliteration is more difficult to process by humans as it is not directly based on pronunciation but on the writing system, however is generally considered more reversible, allowing more consistent retransliteration. Unicode Transliteration Guidelines (Unicode (no date)) also note similar reversibility challenges:

> The term transliteration is sometimes given a narrow meaning, implying that the transformation is reversible (sometimes called lossless). In CLDR this is not the case; the term transliteration is interpreted broadly to mean both reversible and non-reversible transforms of text. ... A non-reversible transliteration is often called a transcription, or called a lossy or ambiguous transcription.

Transliteration may not give accurate transforms, but can result in consistent conversions, with a particular character or sequence of characters always giving the same result, independent of the context. However, that is also dependent on a common standard being used for such transformations. Multiple standards or ad hoc mapping mechanisms used by different organizations can result in inconsistencies. Further, even if this can be managed, legacy data may still remain inconsistent.

3.3 Multilingual Availability for Global Use

As discussed, DNRD is collected, maintained and made available to inform global users about the domain name registrant and registrar, for a variety of purposes. For a true global access, this requires data acquired in any language and script to be available in all other languages and scripts, which is not easily possible. United Nations Group of Experts on Geographical Names faced a similar challenge. As UNGEGN suggests "Names of places and features – like Nairobi, Mumbai, Bandung, Nuuk, Sierra Nevada, Lake Taupo and IJsselmeer – are keys to accessing our digital world … Duplication of names and lack of clearly recorded names have resulted in confused instructions to emergency services and wasted time, which in turn have led to loss of life …" (UNGEGN 2007). To address this challenge of finding the right place consistently, across multiple languages, the practical solution recommended by United

Nations is that "**[t]he Roman script (also referred to as Latin script) has been adopted as a base for international use by the United Nations, and the Group of Experts strongly recommends the development of a single romanization (that is to say, transliteration) system for each non-Roman script"** (UNGEGN 2006, pg. 15; bold emphasis in original document).

> "Non-Roman scripts can then be converted via their romanization into other scripts for national and international use" (UNGEGN 2006, pg. 11).

It further notes that this is not possible without the national level initiatives.

> The method of conversion from one script to another is generally decided by the country concerned and then submitted for approval as the international system. The United Nations conferences over the last 30 years have agreed upon the romanization of some 30 non-Roman scripts. International toponymic usage still depends on the availability of official toponyms established within each country. The United Nations organization encourages each country to provide official national names, in a form suitable for use on maps, using its own standard writing script. It also urges all countries with non-Roman scripts to provide a single system of romanization (that is to say, conversion of its script into Roman script) (UNGEGN 2006, pg. 11).

Though using romanization as a pivot for transforming data from any language and script to any other does seem like a possible method to make the DNRD available to global users, it may still have practical limitations as it requires romanization tables for each language. These tables may not be available in all cases. To address this challenge, the Unicode Transliteration Guidelines (Unicode (no date)) propose a step-wise back-off implementation technique based on BCP 47 (Phillips and Davis 2009), which provides reprieve in such cases in a consistent manner. The guidelines suggest to "progressively handle the fallback among source, target, and variant, with priorities being the target, source, and variant, in that order."

As an example, if Russian language is to be transformed to English, the first preference would be to do it through a Russian-English (source-language-to-target-language) table provided by UNGEGN. If this romanization scheme is not available, an alternate language romanization table (published standard) may be used. If no language based tables are available for the source-target pair, the system should fallback to source-script-to-target-language table published by UNGEGN (or secondary published standard if UNGEGN table is not available). If these tables are also not available, the system should fallback to source-language-to-target-script tables and finally to source-script-to-target-script tables (if earlier options are not available). This fall back sequence is illustrated by Unicode Transliteration Guidelines for Russian-English language pair, as given below (Unicode (no date)).

1. Russian-English/UNGEGN
2. Russian-English [/alternate option]
3. Cyrillic-English/UNGEGN
4. Cyrillic-English [/alternate option]
5. Russian-Latin/UNGEGN
6. Russian-Latin[/alternate option]
7. Cyrillic-Latin/UNGEGN

8. Cyrillic-Latin[/alternate option]

For this to be possible, language and script information for each element which is to be transformed has to be specified when the data is provided. Further, during access of this information, user may also need the details of the level of transformation done on the data (i.e. which of the levels 1-8 above) and also the standard used to undertake this transformation (UNGEGN or alternate standard name). All relevant guidelines and tables need to be accessible to the relevant organization(s) to undertake the transformation.

4 Current Practice with Internationalized Contact Information

Before assessing the techniques and tools for transforming the contact data, the current report studies what are the current practices for collection and display of such information by (i) general e-commerce websites, and (ii) registries and registrars. In the latter case, any constraints due to current relevant protocols are also documented.

4.1 Internationalized Contact Data Support in E-Commerce Websites

E-commerce websites which operate globally have to deal with internationalized contact information as a critical piece of information as part of their business process. Operations of few large e-commerce websites are selected in countries which have one or more pre-dominant local language(s). Submission and display of localized contact information is observed and reported for these websites. The script and language information for the websites analyzed is given in Table 4.1 below.

Table 4.1. Language and scripts relevant for websites analyzed

Name	Country	Script	Language
Amazon	USA/ Global	All	All
Alibaba	China/ Global	All	All
Rakuten	Japan	Kanji, Hiragana, Katakana	Japanese
Homeshop18	India	Local Various	Local Various
LDLC	France	Latin	French
eMall	Saudi Arabia	Arabic	Arabic

Amazon is a company based in United States with subsidiaries in various countries worldwide. The websites of its operations in Spanish and Japanese languages are shown in Figure 4.1 below. Each localized version allows addresses in the local script (note the use of diacritic in the Spanish name *Castaño* and the name in Kanji script in Japanese is allowed by these websites)

(a) Spanish (b) Japanese

Figure 4.1. Websites and sign up web pages of Amazon in (a) Spanish and (b) Japanese

Interestingly, trying to enter an address in Japanese script on the Spanish Amazon (Japan can be selected as a country in the dropdown menu) does not trigger any error, but converts the contact information in html & equivalent form with no further validation process.

(a)

(b)

Figure 4.2. Address information on Amazon showing (a) Input before pressing Enter, and (b) Output after pressing Enter

Alibaba is a company based in China with subsidiaries in various countries worldwide. The website only allows English language characters for signing up, as shown by the error messages given in Figure 4.3.

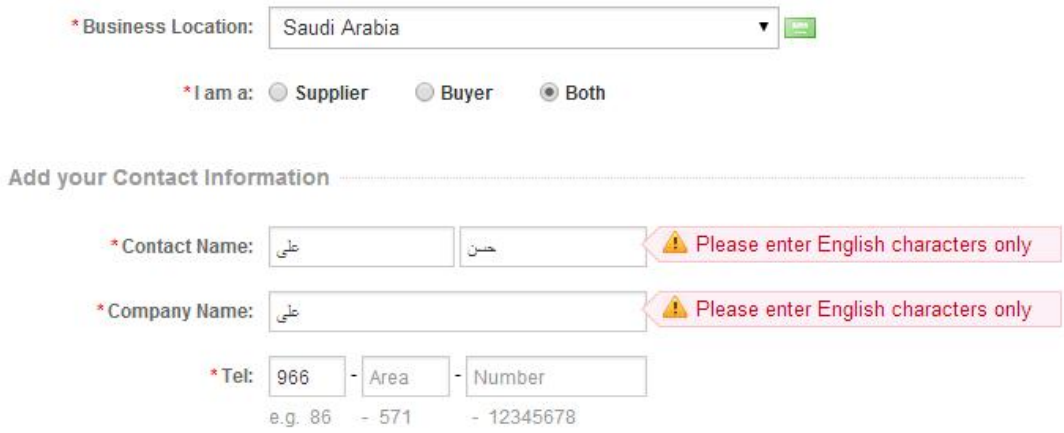


Figure 4.3. Restriction to English characters by Alibaba website

Rakuten, the leading Japanese shopping website, offers shipping abroad through its international website, but it only accepts ASCII within Latin script as shown in Figure 4.4.

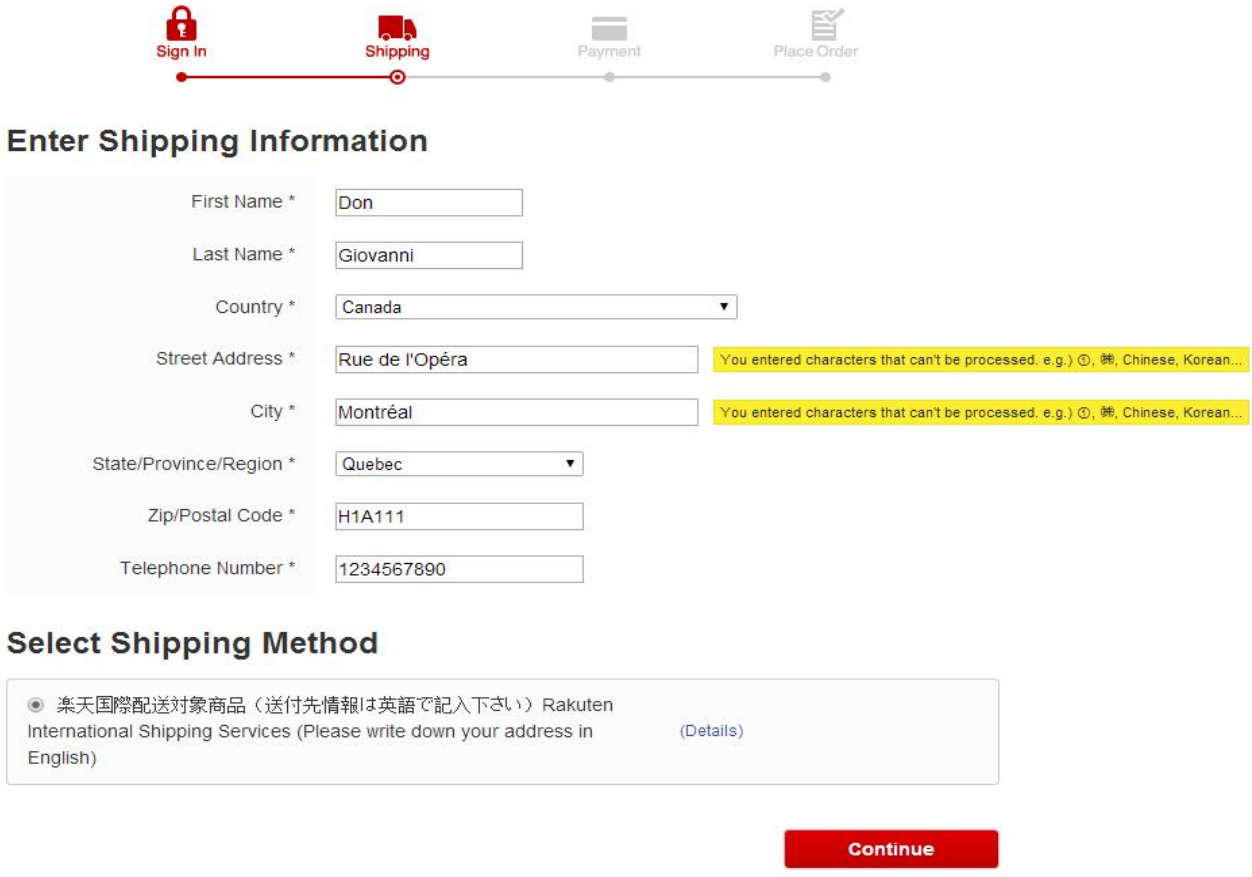


Figure 4.4. Restriction to ASCII characters by Rakuten website

A certain number of other e-commerce websites have been evaluated. None of them will allow shipping outside of their own country, and hence do not resort to transliteration. Here are two more minor examples to help understanding the usual behaviour.

The website for Hoemshop18 is targeted for shoppers from India (no option for country). It is in English only and there is no version in any of the local languages in any other script. It allows addresses in Latin as well as other scripts used locally. However, use of digits is limited to ASCII for the phone number and pincode, as shown in Figure 4.5.

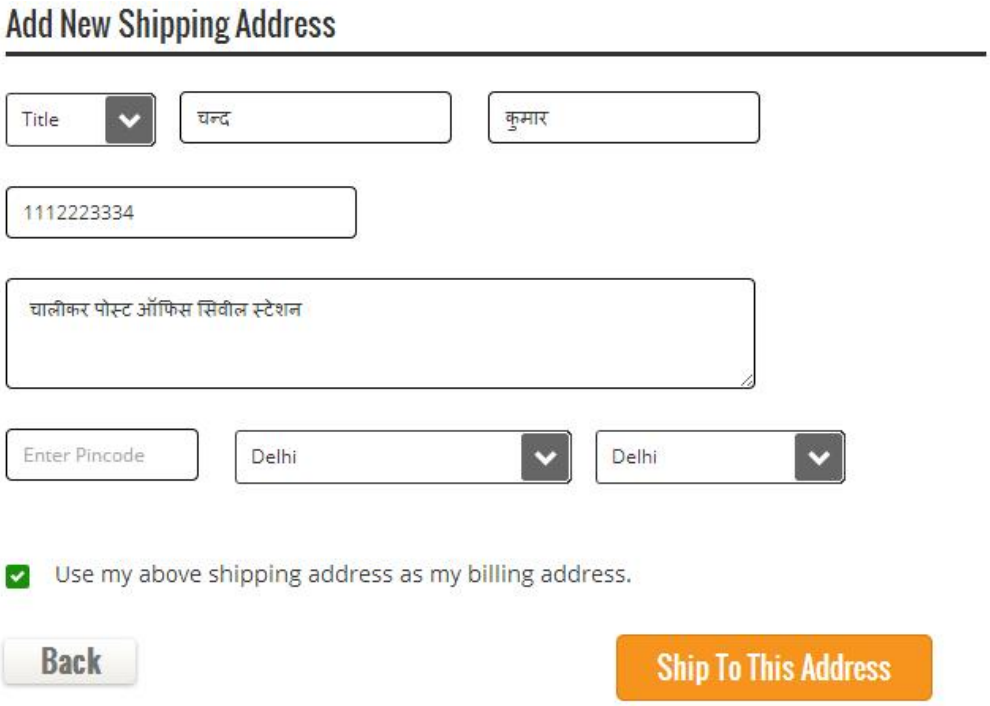
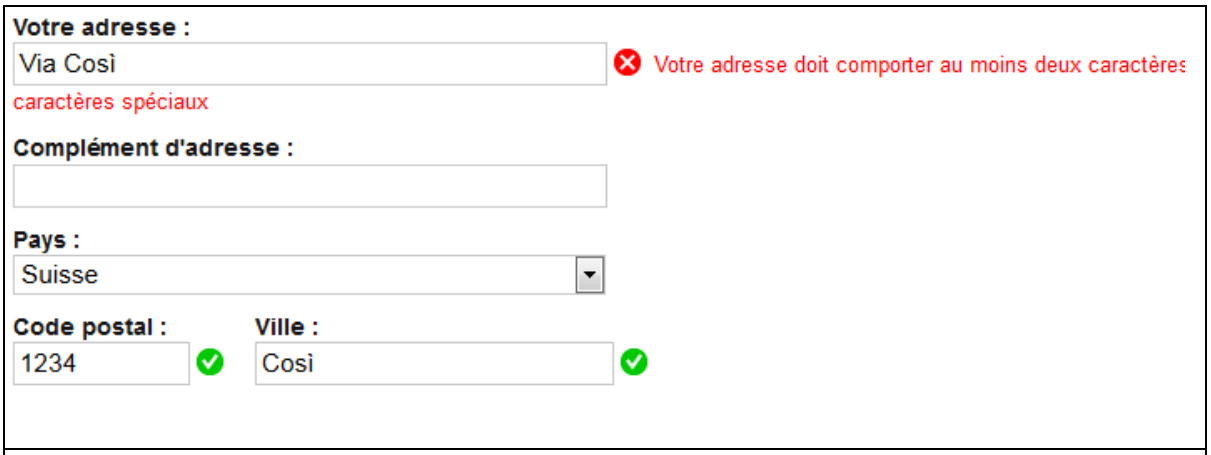


Figure 4.5. Homeshop18 website in India allows Hindi in Devanagari Script

LDLC is a French company allowing orders from Switzerland. Use of French decorated Latin characters on this website is compared with the use of German or Italian decorated characters. Figure 4.6 shows that LDLC allows the full French set of characters, but does not support extended character set valid for the other languages using Latin script in the region.

(a)

(b)



(c)

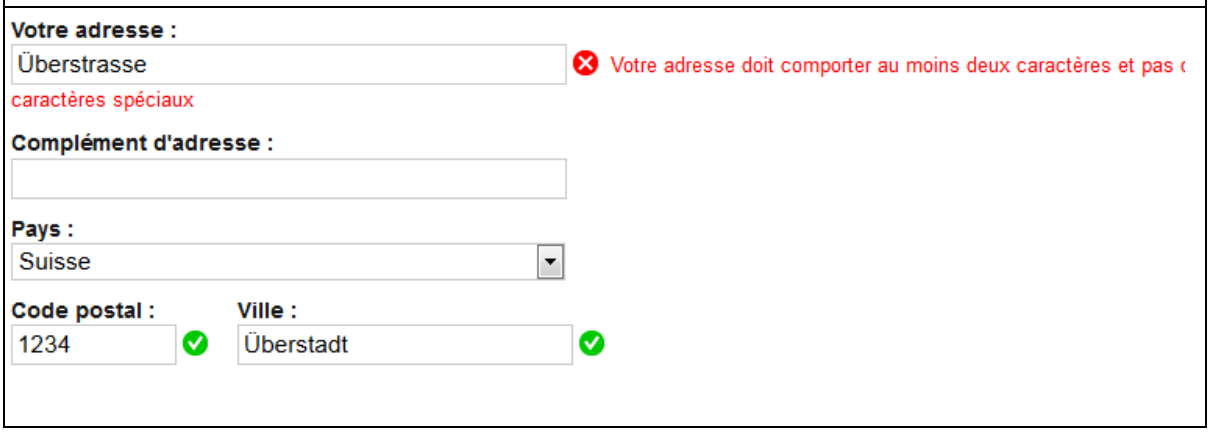


Figure 4.6. LDLC website in France allows extensions for (a) French, but not for (b) Italian, or (c) German, also spoken in the region

The error says "Your address must be at least two characters long and must not include special characters". The characters ì and Ü are rejected in the address but not in the city field.

The eMall website is for local audience in Saudi Arabia. It allows users to sign in using Arabic language. It allows extended Arabic script for username but limits other fields to only characters allowed in Arabic language, as shown in Figure 4.7. Password and phone fields are only allowed in ASCII. The first error message says that "The password should contain at least 8 characters and should only contain letters and numbers." The last error message says "Special symbols and numbers are not allowed." This is because the text boxes contains ى letter (U+06CC) which is not used in Arabic language.

Figure 4.7. eMall website in Saudi Arabia allows Arabic script but limits to ASCII or Arabic language characters in certain fields

This survey of e-commerce sites shows consistently that the websites allow data in local languages but verify the contact data provided by the user only to a limited extent and that too for only a subset of fields. Most just accept the user input, without dealing with the complexity of multi-scripts/language contexts, putting the onus of verification of addresses on the user. Some websites are even active in markets where they do not support the dominant script or language used. It should be understood that e-commerce include various parties involved such as the shipping partner. Therefore, the seller is a conduit of the contact data to the shipping partner. The latter is the one who really needs accuracy of the data to ship at the right physical destination.

4.2 Registry and Registrar Practices for Internationalized Contact Data

Many registries and registrars already serve registrants using extended Latin or other scripts. Separate surveys have been conducted to find the current practices of such registries and registrars. The registry survey has been responded by twelve registries representing large gTLDs and ccTLDs covering multiple languages and scripts, such as Arabic, Han, Cyrillic, Japanese, German, French and English. Those registries are spread over multiple continents and most are in markets where the primary language is not English.

The registrar survey has been responded by two registrars, of which one is a very large registrar, in the time frame of the study. As the survey has a limited number of respondents, conclusions should not be generalized, but may still provide insights into relevant operations. The summary of responses is provided by the registries is given in Table 4.2 and by the registrars is given in Table 4.3.

Table 4.2. Summary of responses to the survey by 12 registries

Question	Answer
Does the registry support EPP?	Yes : 75% No : 25%
Does the registry support other methods than EPP?	Yes : 66% No : 33%
In which language/script registration data submission is allowed?	Any language/script : 33% Specific language(s)/scripts(s) : 66%
Can a registrant submit the same data in multiple languages/scripts?	Yes : 58% No : 42%
Is the data tagged, identifying the language/script?	Yes : 25% No : 75%
If yes, is each data element tagged separately?	Yes : 1 No : 2
Is there a mandatory language/script?	Yes : 58% No : 42%
Is romanization required?	Yes : 42% No : 58%
If two versions of the same data are submitted, do the registry care of consistency?	No : 100%
If two versions of the same data are submitted, which one is considered primary?	Variance of responses : for some, it is the localized version (native script), others don't care.
Support of EPP <postalinfo type= 'loc'>	Allowed : 56% ; Required : 22%, Disallowed : 22%
Support of EPP <postalinfo type= 'int '>	Allowed : 56% ; Required : 22%, Disallowed : 22%
Using any non-standardized extensions of EPP?	Yes for 5 registries. None related to IRD
Percentage of registrants submit data in non-English?	45% of registries : 0% of non-English 45% : >95% 10% : ~15%
Other methods than EPP to submit registration data?	Web portal : 75%
How these methods handle internationalized data?	Majority : UTF-8; One is local charset
Does the registry transliterate or translate submitted registration data?	No : 100%

Does the registry alter data between submission and display?	No : 100% (except one case of converting between encodings)
Displaying all available data (in different scripts) or choose a specific one?	All : 66% Specific one : 33%
Is i18n data displayed over WHOIS protocol?	Yes : 66% No : 33%
Specific marking used for i18n data in WHOIS?	None
Web display of RD shows all versions?	Variance in responses : only ASCII, every version, depends on settings, two separate page, depends on use of 'loc' vs. 'int'
Additional remarks	Implementation varies across TLDs

This limited registry survey already shows that the support of multiple versions of the registrant data in different languages is not widely supported. Moreover, the infrastructure needed to support internationalized registration data, such as EPP support, language/script tagging is also not widely supported. Finally, none is currently doing transformation of registrant data.

Table 4.3. Summary of responses to the survey by 2 registrars

Question	**Answer**
Registrant allowed to submit data in any language/script?	Yes : 50% No : 50%
Registrant allowed to submit data in multiple language/script?	Yes : 50% No : 50%
Percentage of registrants submitting data in their own language/script?	0%, 5%
Any transliteration/translation/ romanization done?	No : 100%
Registration interface available in multiple languages?	Yes : 50% No : 50%
If two versions of the same data is submitted, do the registrar care of consistency?	No : 100%
Registrar support for submitting i18n RD to registries?	Yes : 50% No : 50%
Registrar support for submitting multiple versions of i18n RD to registries?	Yes : 50% No : 50%

The fact that only two registrars have responded limit the conclusions which can be drawn from this survey. At least one registrar interviewed said that accreditation process of ICANN required collection of data in English in parallel with local script, thus the registrar initially collected data in local script but then switched its practice to collect data in local script and English.

4.3 Review of Relevant Protocols

The goal of this section is to review the standard submission and display protocols in current use and identify the gaps, if any, that would prevent or otherwise negatively affect implementation of IRD. Only standard protocols are reviewed: proprietary protocols or proprietary extensions are out of scope.

4.3.1 WHOIS (RFC 3912)

It is well known that WHOIS does not support internationalization. The WHOIS protocol has no mechanism for indicating the character set in use. Originally, the predominant text encoding in use was US-ASCII. In practice, some WHOIS servers, particularly those outside the USA, might be using some other character set either for requests, replies, or both. This inability to predict or express text encoding has adversely impacted the interoperability (and, therefore, usefulness) of the WHOIS protocol. Therefore, WHOIS cannot be used for implementing IRD. Note also that ICANN's Expert Working Group on gTLD Directory recommended to abandon WHOIS in its initial report.

4.3.2 EPP (RFC 5730, RFC 5733)

EPP allows an UTF-8 and/or an ASCII representation of registration data. The following relevant excerpt from RFC 5733 says:

> Two elements are provided so that address information can be provided in both internationalized and localized forms; a "type" attribute is used to identify the two forms. If an internationalized form (type="int") is provided, element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8.

Thus, using “loc” other scripts can be supported. However, EPP still has the following possible gaps:

a. It is not possible to specify more than two versions of the same data.
b. It is not possible to specify more than one UTF-8 version of the same data, for example having traditional and simplified Chinese versions of the same data.
c. It is not possible to specify more than one ASCII version of the same data. For example having a romanized version and a translated to English version of Chinese data.
d. It is not possible to tag the language of the data, as UTF-8 encoding can be used to determine only the script.
e. There is no information on the provenance of the data. It is not possible to know who has created the data (registrant, registrar, or registry), whether the data is the result of translation/transliteration, and if so whether it has undergone an automatic (computer) or manual (human) transformation.

4.3.3 RDAP (draft-ietf-weirds-json-response-06, RFC 6350)

RDAP is the WHOIS replacement protocol being developed in the IETF by the WEIRDS working group. Its goal is to fully support internationalization. It relies on vCard 4.0 (RFC6350) for contact data. vCard fully

supports internationalization through the LANGUAGE and ALTID parameters. There are no limitations on the number of versions of the same data or on the combinations of languages that are supported.

However, one gap has been identified, that there is no information on the origin of the data. It is not possible to know who has created the data (registrant, registrar, or registry), whether the data is the result of translation/transliteration, and if so whether it has undergone an automatic (computer) or manual (human) transformation.

The original protocols have had limited support for internationalized data. However, the more recent work is allowing for supporting languages and scripts. However, these protocols are conspicuously missing any scheme to record the source of the data and if its transformation history.

5 Transformation Methods for Scripts and Languages

The main aim of this study is to study the maturity of tools for transforming contact data among different languages, including translation, transcription and transliteration processes, as already discussed. Four types of writing system are studied to cover the breadth of challenges in transforming these writing systems, understand the standards and solutions available for them, and gauge level of their accuracy. These writing systems include the Logographic (Han), Abugida (Devanagari), Abjad (Arabic) and Alphabet (Cyrillic) systems. Before embarking on a detailed analysis of the tools available for transforming contact data, it is important to understand the differences with which these writing systems encode information and the existing methods and standards available for their transformation. This section also summarizes the status of adoption of these available standards. The discussion is focused on a few languages for each of the scripts, which are being studied in the current work.

5.1 Han Script

Han script is used to write Chinese language and is also used in Japanese (Kanji). It represents a Logographic writing system, in which a character represents a word, morpheme or a semantic unit. As the writing system is not based on sound units, it is not straightforward to map its characters onto alphabetic writing systems like Latin (as the latter are sound based). This presents a unique challenge while romanizing the languages written using this script. This challenge is usually met by replacing the logogram by how it is pronounced in the source languages, like Chinese. However, this transformation requires: (i) to determine which accent of source language will be used, as different accents may pronounce the same source character differently resulting in different romanization, and (ii) a standard way of mapping the source language sounds to Latin characters. As this system must be based on how letters are pronounced, it is inherently a transcription scheme. Further, as more than one character may represent a sound in source or target languages, the transformation can be ambiguous from or into the Han script.

There have been multiple romanization systems used for Han script used to write Chinese language in China, including Zhuyin, Wade-Giles (1859; 1892) and Chinese Postal Map Romanization. However, Pinyin system has replaced these systems since it was formally adopted by China in 1958 and is now commonly used across China. A different scheme is used in Taiwan, Hong Kong and Singapore. Pinyin

system defines the initial (consonant) and the final (remaining syllable; vowel and any consonant) mapping. In addition, it allows for combining marks with the vowel to represent the tones in Chinese, for example á for rising tone, à for falling tone, ā for high tone, and ă for falling-rising tone. The pinyin system is based on Beijing accent of Chinese and uses a fixed mapping into Latin based on pronunciation only, therefore largely addressing the potential ambiguities in romanization. However, there are still ambiguities in retranscription. This system has been adopted by ISO as the standard romanization for modern Chinese (ISO 7098:1991, updated from ISO 7098:1982) and the United Nations (1977)[2]. Pinyin is also used by many other organizations, for example the American Library Association.

5.2 Devanagari Script

Devanagari script is used to write Hindi, Nepali, Marathi and a few other languages. It is an Abugida system of writing, deriving from Brahmic script and related closely to many other scripts used in South Asia, including Bangla, Gurmukhi, etc. The consonant-vowel pair is normally written, with consonant as the primary component and vowel as a secondary diacritical mark around it. However, the vowel mark is required, with an inherent (default) vowel if no vowel mark is explicitly written. Thus, for consonants which are to occur without a vowel (e.g. in coda position of a syllable), the vowel has to be explicitly suppressed using a special Halant combining mark. Consonant clusters (without vowels) are also possible and are normally graphemically (visually) fused to form conjuncts (though underlying encoding sequence remains unchanged). As both consonant and vowels are written, the writing system may be more easily mapped on to an alphabetic system and thus can give a reasonably accurate transliteration or transcription. One challenge faced in the process is that the Halant is optionally written, meaning that if it is not present, it is unclear if the inherent vowel is to be added.

The Hunterian system of transliteration, developed in late nineteenth century and more formally adopted in 1872, is generally used for Hindi in India. A slightly different system is proposed by UNGEGN but is not yet adopted (UNGEGN 2013). Other schemes include International Alphabet of Sanskrit Transliteration (IAST) developed in 1894, limited to academic use, and ISO 15919 developed in 2001 which includes transliteration scheme for related scripts and is similar to IAST for Devanagari. ALA-LC romanization is used by Library of Congress and American Library Association. In Nepal, a system developed by Nepal Survey Department is currently being used for Nepali language (UNGEGN 2013). Latin decorated with diacritics is normally used to represent letters and dependent vowels for many of these standards, giving accurate representation and reversibility.

5.3 Arabic Script

There is a diverse set of languages from different language families across the world using the Arabic script and its extensions, e.g. Achenese, Arabic, Fula, Malay, Persian, Pashto, Sindhi, Swahili, Urdu and many more. As these languages are very different in their phonetic systems, the same letter in Arabic script may represent different sounds. Therefore, the transformation to Latin or English would be dependent on source-language. For example, ض is pronounced as /dˤ/[3] in Arabic but /z/ in Urdu

[2] See http://www.eki.ee/wgrs/rom1_zh.htm.
[3] The pronunciation within // represents phonemic transcription using International Phonetic Alphabet (IPA).

resulting in different transformation into Latin script or English language. Further, Arabic is an Abjad writing system, where consonants are written and vowels are generally not written or under-specified. This is a significant issue in transforming Arabic script based languages into English language or Latin script because it means that transformation will only (or mostly) contain consonants without vowels, and therefore not pronounceable. For example, the word كتب for "books" is transformed at "ktb" instead of "kutb". Such transformations also create significant ambiguities (see Alkharashi (2009)). This needs a "vowelisation" or vowel insertion process which would need more significant language processing. Further, when vowels are specified, due to dialectical differences multiple transformations may be possible, e.g. giving up to forty different transliterations for the unique spelling of the Arabic name سليمان, including 'Salayman', 'Seleiman', 'Solomon', 'Suleiman' and 'Sylayman' (Pouliquen 2005).

Beesley (1998) summarizes the challenge in converting Arabic language from Arabic script as given below, which is equally applicable to other languages using the Arabic script.

> Both transcription and transliteration have their uses, and the two can seldom resemble each other for Arabic. Because of unwritten vowels and other diacritics, and because of ambiguous and silent letters, standard Arabic orthography is a poor clue to pronunciation, especially for non-Arabic speakers who can't reliably guess which reading of a word is appropriate in syntactic context; conversely, a good phonological transcription is often a poor clue to standard orthography. It's a serious mistake to try to do Arabic transcription and transliteration at the same time.

For Arabic script, UNGEGN has proposed transliteration recommendations for Arabic, Persian, Uighur and Urdu languages (UNGEGN 2013). International Standards Organization also proposes transliteration standard for Arabic language. The latest specification is ISO 233-2 (1993), based on earlier ISO 233 standard (developed in 1984, with an earlier version in 1961). Multiple other regional, national and ad hoc romanization schemes exist for Arabic language. A (non-exhaustive) list[4] includes BGN/PCGN (1956), ANSI Z39.12-1972 (R1984) and Buckwalter (1991) into English; IGN System (1973) into French; and DIN 31635 (1982) into German. Latin, decorated with diacritics, is used for representing the letters and diacritics of Arabic language, giving accurate representation of the written (mostly consonantal) form.

Based on the national cartographic products, UNGEGN (2003) reported that the use of their transliteration recommendations are current in Iraq, Kuwait, Libya, Saudi Arabia, United Arab Emirates and Yemen. There is partial usage in Syria, Egypt and Sudan, whereas Algeria, Djibouti, Mauritania, Morocco and Tunisia use traditional systems based on French orthography. The challenges in adoption of UNGEGN guidelines for Arabic language transliteration are summarized by Atoui (2012), which include coordination between the 22 Arab states, national focus on adopting cartographic standards, regional capacity and regional differences in Arabic language.

[4] See http://en.wikipedia.org/wiki/Romanization_of_Arabic for a more exhaustive list and a comparative chart.

The Persian[5] transliteration scheme is currently used in Iran and the Uighur[6] transliteration scheme is used in China and other international efforts. The scheme proposed for Urdu[7] is currently not being used in Pakistan, India or elsewhere. Other schemes are used, e.g. the American Library Association-Library of Congress Scheme (ALA-LC) for Urdu[8].

5.4 Cyrillic Script

Cyrillic script is used across Euro-Asian region for Slavic and other languages, including Russian, Ukrainian, Bulgarian, Kazakh, Mongolian and many other languages. It is an alphabetic system with letters having capital and normal forms, and with many letter shapes similar to Latin script. Due to similarity of the systems, a fairly accurate romanization is possible for Cyrillic script.

Scientific Transliteration scheme has been defined for romanization of many of the languages using Cyrillic script. Revised ISO 9 version developed in 1995 is based on Scientific Transliteration, except that it is language independent and codified for the Cyrillic script. In Russia, GOST standard is used, which was originally developed by National Administration for Geodesy and Cartography at the Council of Ministers in Soviet Union in early 1970s to cover multiple languages using the Cyrillic script. The standard has undergone multiple revisions in 1980, and then in 2000s, after which it is now similar to ISO 9. It has also been adopted by UNGEGN (2003) for Russian. GOST is used for romanization of information on passports in Russia. Ukraine uses Ukrainian National Transliteration scheme for romanization, approved in 2010, which is also employed for romanization of information on passports. This is also used by UNGEGN (2013). Other standards including the Scientific Transliteration and ISO 9 may also be used for Ukrainian language. BGN and PCGN standards also exist for these languages, as well as the American Library Association-Library of Congress (ALA-LC) romanization schemes. These schemes are mapped to Latin characters, without significant use of diacritics.

5.5 Standards and Resources for Other Scripts and Languages

In addition to the transliteration standards discussed, there is support for a variety of languages by many organizations. Some of these are summarized in Table 5.1 below.

Table 5.1. Standards for Transforming Scripts and Languages

Organization	Standards
International Organization for Standardization (ISO)[9]	• ISO 9 — Cyrillic • ISO 233 — Arabic • ISO 259 — Hebrew • ISO 843 — Greek • ISO 3602 — Japanese • ISO 7098 — Chinese • ISO 9984 — Georgian

[5] See http://www.eki.ee/wgrs/rom1_fa.htm.
[6] See http://www.eki.ee/wgrs/v2_2/rom1_ug.htm.
[7] See http://www.eki.ee/wgrs/rom1_ur.htm.
[8] See http://www.loc.gov/catdir/cpso/romanization/urdu.pdf
[9] Also see http://www.iso.org/iso/products/standards/catalogue_ics_browse.htm?ICS1=01&ICS2=140&ICS3=10.

	• ISO 9985 — Armenian • ISO 11940 — Thai • ISO 11940-2 — Thai (simplified) • ISO 11941 — Korean (different systems for North and South Korea) • ISO 15919 — Indic scripts
United Nations Group of Experts in Geographic Names (UNGEGN)	Numerous tables given at http://www.eki.ee/wgrs/
Universal Postal Union	S42 International Addressing Standard (see details in the text after this table)
Unicode Consortium	Numerous tables (and variations) given at http://unicode.org/repos/cldr/trunk/common/transforms/
United States Board on Geographic Names and the Permanent Committee on Geographical Names for British Official Use (BGN/PCGN)	29 languages published in BGN (1994) available at http://libraries.ucsd.edu/bib/fed/USBGN_romanization.pdf
American National Standards Institute (ANSI)	• ANSI Z39.12-1972 (R1984). System for the Romanization of Arabic. NISO. Paper (8 p.). • ANSI Z39.37-1979. System for the Romanization of Armenian. NISO. Paper (7 p.) • ANSI Z39.25-1975. Romanization of Hebrew. NISO. Paper (15 p.). • ANSI Z39.11-1972 (R1983). System for the Romanization of Japanese. NISO. Paper (11 p.). • ANSI Z39.35-1979. System for the Romanization of Lao, Khmer, and Pali. NISO. Paper (14 p.). • ANSI Z39.24-1976. System for the Romanization of Slavic Cyrillic Characters. NISO. Paper (10 p.).
American Library Association - Library of Congress (ALA-LC)	74 languages reported at http://www.loc.gov/catdir/cpso/roman.html

The standards given above are for transformation of content. Universal Postal Union also specifies structure of addressing elements and transformation of the structure for different countries, as explained below and shown in Figure 5.1.

> The S42 international addressing standard comprises of a generic list of address elements (used in all UPU member countries) and country-specific templates that tell users how to transform address elements into an accurately formatted address. In other words, a country defining its S42 template provides precise information about its address elements and formats. This can be incorporated into software programs to manage addresses.[10]

[10] Source: http://www.upu.int/uploads/tx_sbdownloader/sheetAddressingS42InternationalAddressingStandardsFactSheetEn.pdf

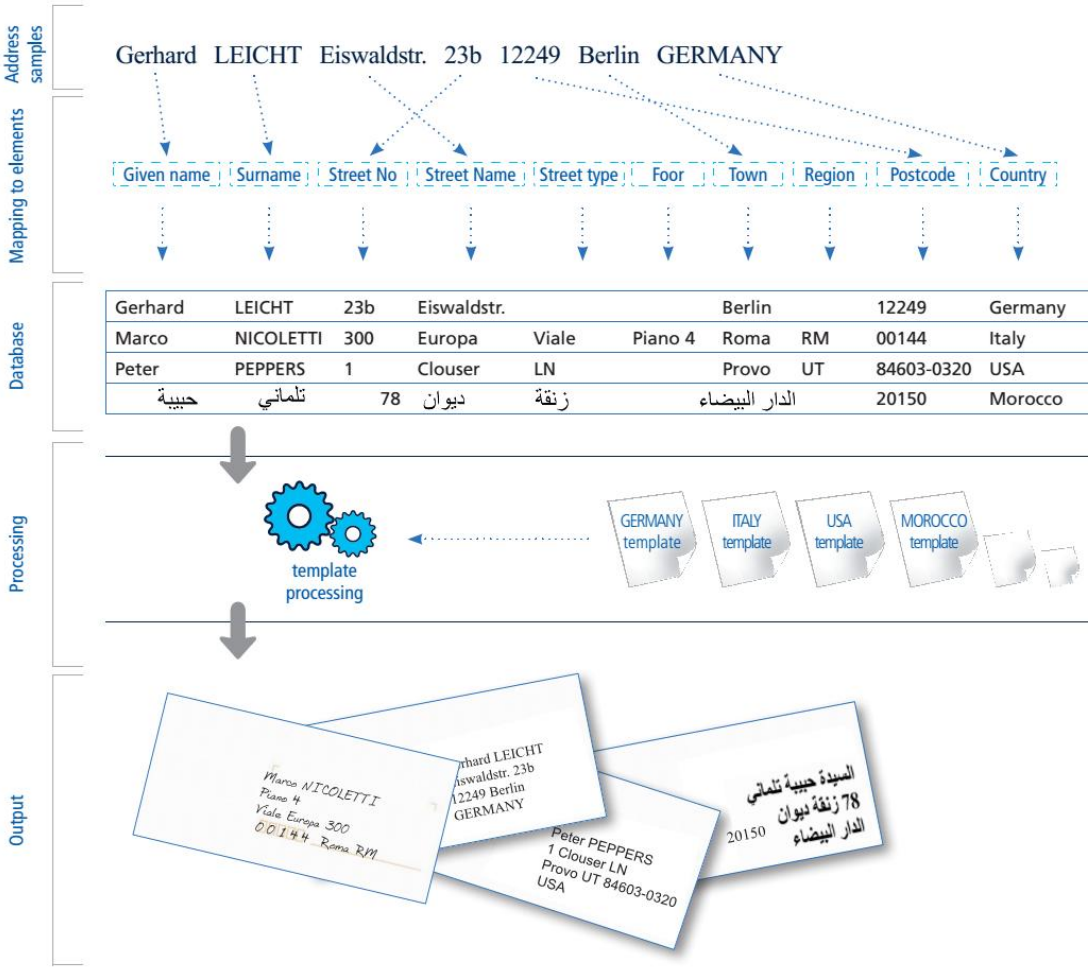


Figure 5.1. Illustration of S42 International Addressing Standard by Universal Postal Union[11]

UPU specifically asks for the country to be listed in the language of the dispatching country or in an internationally recognized language[12]. For detailed addressing formats, see http://www.upu.int/en/activities/addressing/postal-addressing-systems-in-member-countries.html.

In addition to these standards, the UN Gazatteer deserves a special mention in the context of contact information. It is a collection of over 8 million place names and a search engine which employs fuzzy logic to find location names worldwide by comparing phonetic romanized spellings. This is developed by UN Cartographic Section (UNCS) to serve the Security Council and the Secretariat including UN field missions. For details see http://dma.jrc.it/services/gazetteer/, and see http://ggim.un.org/projects.html for further information. Names of main cities and all the countries are also available through the website at http://unstats.un.org/unsd/geoinfo/geonames/, with formal full names of countries and their

[11] Source: http://www.upu.int/uploads/tx_sbdownloader/sheetAddressingS42InternationalAddressingStandardsFactSheetEn.pdf

[12] See http://www.upu.int/uploads/tx_sbdownloader/descriptionPostcodesAddressingAddressElementsEn.pdf

short forms in UNGEGN (2011). A longer list for similar initiatives at national levels in other countries is provided at http://unstats.un.org/unsd/geoinfo/UNGEGN/geonames.html.

6 Transformations Tools and their Analysis

There are many tools which transform between language and/or script pairs using the mechanisms and standards discussed. The current study tests some of these tools for a few languages across Han, Devanagari, Arabic and Cyrillic scripts.

6.1 Tools

There are many tools which can be used for transforming data, of different categories. These included general translation tools, general conversion tools (including those which just do transliteration), and some specialized tools for contact information, some focusing on names, others on addresses. A sample set of such tools are given below[13].

Some general translation tools are listed here (in alphabetical order):

- Ace Translator (http://www.acetools.biz/)
- Babylon (http://translation.babylon.com/)
- Google Translate (https://translate.google.com/)
- Microsoft Translate (http://www.microsoft.com/en-us/translator/)
- Power Translator (https://www.lec.com/power-translator-software.asp)
- Systrans (http://www.systransoft.com/)
- Translution (http://www.translution.com/default.asp)

Some general transliteration or transcription tools are listed here (in alphabetical order):

- Google Input Tools (http://www.google.com/inputtools/)
- IBM ICU Transliteration (http://demo.icu-project.org/icu-bin/translit; also see http://userguide.icu-project.org/transforms/general)
- JUnidecode (http://www.ippatsuman.com/projects/junidecode/index.html)
- Microsoft Transliteration Utility (http://msdn.microsoft.com/en-us/goglobal/bb688104.aspx)
- Ok-board.com (http://ok-board.com/)
- Unidecode (https://pypi.python.org/pypi/Unidecode)
- Yahoo Transliterate (https://transliteration.yahoo.com/)

Some tools focused on various parts of contact information transformation are listed here (in alphabetical order; many of these tools focus on address verification and not transformation):

- Address Doctor (http://www.addressdoctor.com/en/)
- Basis Technology Rosette Name Translator (http://www.basistech.com/text-analytics/rosette/name-translator/)
- Experian Data Quality (http://www.qas.com/contact-data-quality.htm)
- IBM Global Name Recognition (http://www-01.ibm.com/common/ssi/cgi-

[13] This list of tools is arbitrary and not comprehensive.

bin/ssialias?infotype=an&subtype=ca&appname=GPA&htmlfid=897/ENUS207-295)
- Loqate (http://www.loqate.com/technology/transliteration/)
- Trillium Software (http://www.trilliumsoftware.com/products/data-types/customer-data/)

Due to the limited scope of the study, it is not possible to test a comprehensive set of tools, however a few are tested to investigate the extent of support for translation and transliteration of relevant contact data. The shortlisted tools include those that cover a reasonably large number of languages and scripts. Both open source and proprietary tools are explored, to the extent that they are available for testing in the short time of the study. Based on these criteria, the following tools are tested on a limited set of data (as discussed in the next sections).

Two general translation tools (Translation1 and Translation2), two general transliteration tools (Transliteration1 and Transliteration2) and one specialized tool for contact information (Specialized1) are tested in the current study. More organizations with specialized tools have been contacted, however, their tools are not tested for one of the following reasons: no response is received, the company suggests that their focus is not on transliteration but on address verification, the organization does not provide their tool for independent testing and suggests that they do the testing for the study (a methodology which the study team does not follow for other tools), the testing process is not easily possible as the company requires signing an NDA and paying for the testing services.

6.2 Data

Though the Internationalized Registration Data may contain many fields, transformation of only contact information is needed. This information can be grouped into the following four general categories:

1. Individual or Entity names, including family and given names, organization names, etc.
2. Addresses, including proper names, generic terms (which should not be transformed), abbreviations (where applicable), punctuation, digits, etc.
3. City and state/province names
4. Country names, including full and short forms

Though in general contact information may appear in context of a larger sentence, in IRD there is no context available for such data, which may have adverse impact on the translation of this data. This has no bearing on its transliteration.

At least 50 cases for each script are tested, distributed in the four categories given above. Normal examples of names, addresses, cities/states and countries are tested, and only a few boundary cases (if any). The test cases are developed primarily for one major language using the script, though a few cases are included to cover a few additional languages, as per the following details:

1. Han (Chinese using Traditional and Simplified Chinese writing; Traditional Chinese examples are from Taiwan)
2. Devanagari (Hindi, Marathi)
3. Arabic (Arabic, Persian, and Urdu)
4. Cyrillic (Bulgarian, Russian and Ukrainian)

The Logographic, Abugida, Abjad and alphabetic systems are used to determine the script coverage, whereas language variation is used to find out support of language dependent variations within each script. The details of the test cases are given in Table 6.1.

Table 6.1. Details of Testing Data Used for Various Scripts

Script	Type	No. of Items	No. of Words	No. of Characters	Notes
Han	Name	12	27	136	Data covers Chinese language (both Traditional and Simplified)
	Address	12	129	818	
	City /State	5	5	38	
	Country	5	11	65	
Devanagari	Name	22	22	180	Data covers (mostly) Hindi and Marathi languages
	Address	12	73	430	
	City /State	26	37	295	
	Country	-	-	-	
Arabic	Name	20	20	115	Data covers (mostly) Arabic, Urdu and Persian languages
	Address	15	49	320	
	City /State	10	13	77	
	Country	10	14	100	
Cyrillic	Name	20	21	150	Data covers (mostly) Russian, Ukrainian and Bulgarian languages
	Address	14	30	216	
	City /State	11	19	174	
	Country	10	10	67	

For all the data points, language and script users are asked to develop the test data in local script and languages and provide the corresponding correct English representation. They are also requested to provide multiple acceptable English versions, where possible. For example, محمد in Arabic language may be represented as Muhammad, Mohammed, etc. and चौधरी in Hindi language as Chaudhary, Choudhry, etc. The English representations provided are used as the "gold reference" for comparison with output of the tools.

6.3 Criteria

The most straight-forward and the conservative criteria of transformation is to determine if the tool output has an exact match with the gold reference. It gives a simple TRUE or FALSE answer for each test. Due to limited number of test cases, these numbers may only be used as a comparative measure of accuracy between various tools, and not an absolute measure of accuracy for an individual tool. The transformation is done in both directions (source ←→ romanization) to determine the accuracy in each direction.

Though a strict accuracy measure is useful, a user may still be able to comprehend and use a transformation which is similar to, even if not exactly the same as, the gold reference. For example, the desired output of Cyrillic Russian Вельов is “Velyov” but “Viel'ov” may also be understandable. Therefore, the study also extends the accuracy analysis to include string similarity of the transformation with the gold reference. This is done by calculating the Levenshtein distance between the transformed output and the gold reference. Levenshtein distance calculates the number of edits (insertion, deletion and substitution) between two strings. If two strings are exactly the same, zero edits are required to change one into the other, so Levenshtein distance is also zero. The maximum distance is equal to the length of the longer string in the pair being compared. Thus, lower the Levenshtein distance, more similar are the strings to each other. There is a Levenshtein distance of two between the words “Velyov” and “Viel'ov” (the two edits needed in the second string to get the first one are: (i) delete “i" and, (ii) substitute “y” for “” in the second word).

In many cases, in the transliteration process, decorated Latin (i.e. with diacritics) is normally produced for faithful transformation and re-transliteration. It is not possible to represent the diacritics in ASCII and can even confuse a user who is expecting ASCII. For example القاهرة (Cairo) is transliterated as ạlqạhrẗ. Further, the gold reference provided by human informants does not have any diacritics. Therefore, for calculating the Levenshtein distance, the output string is first decomposed and diacritics are removed (ạlqạhrẗ reduced to alqahrt), for a more effective measure.

It is worth noting that Levenshtein distance is a purely computational measure and, though useful, it may not accurately map to the similarity between strings as perceived by humans. For example, (i) change in vowel vs. change in consonant may have different perceptual effect on comprehension, but will have same impact on the Levenshtein distance, and (ii) same distance may have different impact on comprehension based on the length of the label, i.e. small change in shorter labels may have more significant impact on comprehension than same change in longer labels, which is not captured by the Levenshtein distance, etc.

A complete round-trip transformation from source (to target language and back) is also done to determine the reversibility of the process. A tool should give exactly the source string after the round trip if it is reversible. This is also compared with the transformation of the reference to source language. Again an exact match metric is used in this case.

6.4 Limitations

With a variety of script systems, tens of scripts, thousands of languages, with each language abound with expression, it is hard to develop a study which can evaluate transformations for a representative set across a variety of tools. Therefore, this study can neither be comprehensive nor representative, but only indicative of the status of technology for the few languages using the scripts covered. It is important to note the limitations of this current study and the results should be interpreted in the context of these limitations.

Though the study has tried to cover a variety of scripts of different types, it has not covered all the script families. There are script types which are still not covered, e.g. syllabary used by Japanese Hiragana.

Even within script families, only a representative script has been covered, for example Devanagari from the Neo-Brahmic family is included, but many more like Bangla, Tamil, Sinhala, etc. are not covered.

A script can be used to write many languages, e.g. the Arabic Script Generation Panel[14] lists around fifty languages which use Arabic script. The current study has mostly looked at a single language per script. Even for the main language covered for each script, the coverage is very limited, with only 50 test cases divided across four categories of data: people names, addresses, cities/states and countries. Though this data gives an overview of the capability of a tool and highlights any challenges, it is insufficient to be considered an exhaustive measure. Further the choice of the strings to be tested is also arbitrary, based on the selection of the informant, even though instructions were given to cover a variety of letters and scenarios in a language for each category. Finally, the English equivalent provided by the informant is also arbitrary and has not been constrained to follow any standards or conventions in transliteration or translation.

It is still noteworthy that the gaps pointed out in the tools and techniques by even the small test set would still remain valid if the data for these languages is increased.

Regarding the selection of evaluative mechanisms, the limitations of Levenshtein distance as a computational similarity measure (and not a perceptual similarity measure) have been already discussed. Reduction of the converted string to its "ASCII equivalent" by de-composing and removing marks from the transformed strings is also a simplifying assumption which ignores the perceptual dissimilarity caused by these marks if they are not removed.

6.5 Results

The tables in this section give detailed results from the analysis of the data. Separate results are tabulated for each script for each of the four categories of data (name, address, city/state, and country) for the different tools which are tested.

The column for No. of Correct Items (or test cases) has the count of test cases whose transformation exactly matches the gold reference (out of total No. of Items or test cases).

- Therefore, the Accuracy (%) is calculated for each category (name, address, city/state, and country) by $\frac{No.\ of\ Correct\ Items}{No.\ of\ Items} \times 100$

- Overall Accuracy (%) is determined for each tool over all categories (name, address, city/state, and country), given by $\frac{\sum_{\text{over all categories}} No.\ of\ Correct\ Items}{\sum_{\text{over all categories}} No.\ of\ Items} \times 100$

String similarity is also computed at category (name, address, city/state, and country) and tool levels.

[14] See the appendix of the *Proposal for the Arabic Script Generation Panel* posted here: https://community.icann.org/download/attachments/43976436/Arabic%20Script%20Generation%20Panel%20Document.pdf?version=1&modificationDate=1390426534000&api=v2.

- Avg. Levenshtein Distance per Word is $\frac{\sum_{\text{All words in the category}} \text{Levenshtein Distance}}{\text{Total No.of Words in the Category}}$

- This is compared to the Avg. No. of Characters per Word to get Levenshtein Distance per Word (%), calculated by $\frac{\text{Avg. Levenshtein Distance per Word}}{\text{Avg. no. of Characters per Word}} \times 100$

- Overall Levenshtein Distance (%) is $\frac{\sum_{\text{All words in a category,All categories}} \text{Levenshtein Distance}}{\sum_{\text{All words in a category,All categories}} \text{Total no.of Characters in a Word}}$

See Table 6.1 for additional figures used in these calculations.

6.5.1 Han Results

Han script is very different from Latin writing system. This difference makes the Han (Chinese in Simplified and Traditional form) difficult to transform. Generally accuracy of translation systems is twice as much as transliteration systems. Looking at the data in more detail, the transliteration of Simplified Chinese is much better by the tools versus the transliteration of Traditional Chinese. For translation the accuracy of both writing systems is high for names, city/state and country. Address transliteration and translation accuracy is generally low across all tools, though slightly better for translation tools.

Levenshtein distance indicates that the transformations are not very accurate. The lowest overall distance is 42%, which is quite high. Address field is very inaccurate and give consistently high distance across tools. Transformations of Names are fairly accurate by comparison.

Table 6.2. Detailed Testing Results for Han Script

Language	Tool	Type	No. of Items	No. of Correct Items	Accuracy (%)	Overall Accuracy (%)	Avg. No. of Characters per Word	Avg. Levenshtein Distance per Word	Levenshtein Distance per Word (%)	Overall Levenshtein Distance (%)
Han	Specialized1	Name	12	6	50	26	5.0	1.0	21	59
Han	Specialized1	Address	12	0	0		6.3	4.1	65	
Han	Specialized1	city	5	3	60		7.6	1.8	24	
Han	Specialized1	country	5	0	0		5.9	5.3	89	
Han	Translation1	Name	12	9	75	59	5.0	1.4	29	49
Han	Translation1	Address	12	1	8		6.3	3.6	57	
Han	Translation1	city	5	5	100		7.6	0.2	3	

Han	Translation1	country	5	5	100		5.9	1.0	17	
Han	Transliteration1	Name	12	6	50	26	5.0	1.0	21	62
Han	Transliteration1	Address	12	0	0		6.3	4.3	67	
Han	Transliteration1	city	5	3	60		7.6	2.4	32	
Han	Transliteration1	country	5	0	0		5.9	5.6	95	
Han	Translation2	Name	12	5	42	53	5.0	1.7	33	42
Han	Translation2	Address	12	3	25		6.3	3.0	47	
Han	Translation2	city	5	5	100		7.6	0.2	3	
Han	Translation2	country	5	5	100		5.9	1.0	17	
Han	Transliteration2	Name	12	6	50	26	5.0	1.4	28	62
Han	Transliteration2	Address	12	0	0		6.3	4.2	67	
Han	Transliteration2	city	5	3	60		7.6	2.4	32	
Han	Transliteration2	country	5	0	0		5.9	5.6	95	

6.5.2 Devanagari Results

Devanagari writing system is similar to Latin, as both consonants and vowels are written out. However, the inherent vowel is not written and that can cause ambiguity and therefore loss in accuracy. Further vowel conventions in tools and writing conventions may also vary, e.g. English spelling may have “ee” instead of “i” causing mismatch. Devanagari script is tested using mostly Hindi data (and some Marathi data). Due to the reason discussed, transliteration tools give very poor accuracy. The translation tools fare better, with overall accuracy of 58%. Country data was not provided, so it is not reported in the table. As for other scripts, the address data is least accurately transformed, due to multiple issues, including word order, transformation of generic terms and others already discussed.

Even though translation tools give much better accuracy figures, the word level Levenshtein distances are comparable for translation and transliteration. Translation tools perform slightly better giving a low distance of 30% overall.

Table 6.3. Detailed Testing Results for Devanagari Script

Language	Tool	Type	No. of Items	No. of Correct Items	Accuracy (%)	Overall Accuracy (%)	Avg. No. of Characters per Word	Avg. Levenshtein Distance per Word	Levenshtein Distance per Word (%)	Overall Levenshtein Distance (%)
Devanagari	Specialized1	Name	22	7	32	18	8.2	2.2	27	48
Devanagari	Specialized1	Address	12	1	8		5.9	3.5	60	
Devanagari	Specialized1	city	26	3	12		8.0	3.4	43	
Devanagari	Specialized1	country								
Devanagari	Translation1	Name	22	10	45	58	8.2	2.8	34	30
Devanagari	Translation1	Address	12	2	17		5.9	2.3	39	
Devanagari	Translation1	city	26	23	88		8.0	1.2	15	
Devanagari	Translation1	country								
Devanagari	Transliteration1	Name	22	0	0	0	8.2	2.0	24	33
Devanagari	Transliteration1	Address	12	0	0		5.9	2.5	43	
Devanagari	Transliteration1	city	26	0	0		8.0	1.9	24	
Devanagari	Transliteration1	country								
Devanagari	Translation2	Name	22	15	68	58	8.2	2.0	24	44
Devanagari	Translation2	Address	12	2	17		5.9	3.7	63	
Devanagari	Translation2	city	26	18	69		8.0	2.2	27	
Devanagari	Translation2	country								
Devanagari	Transliteration2	Name	22	0	0	0	8.2	2.5	31	45
Devanagari	Transliteration2	Address	12	0	0		5.9	3.1	52	
Devanagari	Transliteration2	city	26	0	0		8.0	3.4	43	
Devanagari	Transliteration2	country								

6.5.3 Arabic Results

The data for Arabic script (mostly Arabic language) shows that Transliteration tools give 0% accuracy because vowels are not written in Arabic script but are expected in the gold reference. Translation tools are able to perform better, with best tools still making about 30% error. Though some tools perform better for Names and others for country or city, they all consistently perform worst for addresses, mostly because word omission, insertion and order issues discussed in more detail in the next section.

Levenshtein distance gives more detailed insight to accuracy values. Even though accuracy is 0% due to missing vowels, the transliteration tools still give partial matches based on consonants. Transliteration tools still give high distance values (due to vowel omissions) but translation tools give better results, with as low as 35% difference overall.

Table 6.4. Detailed Testing Results for Arabic Script

Language	Tool	Type	No. of Items	No. of Correct Items	Accuracy (%)	Overall Accuracy (%)	Avg. No. of Characters per Word	Avg. Levenshtein Distance per Word	Levenshtein Distance per Word (%)	Overall Levenshtein Distance (%)
Arabic	Specialized1	Name	20	14	70	40	5.8	1.0	17	61
Arabic	Specialized1	Address	15	0	0		6.5	4.7	73	
Arabic	Specialized1	city	10	7	70		5.9	1.8	31	
Arabic	Specialized1	country	10	1	10		7.1	7.0	98	
Arabic	Translation1	Name	20	13	65	69	5.8	1.7	30	35
Arabic	Translation1	Address	15	6	40		6.5	3.1	47	
Arabic	Translation1	city	10	9	90		5.9	0.6	10	
Arabic	Translation1	country	10	10	100		7.1	1.4	20	
Arabic	Transliteration1	Name	20	0	0	0	5.8	2.2	38	64
Arabic	Transliteration1	Address	15	0	0		6.5	4.4	68	
Arabic	Transliteration1	city	10	0	0		5.9	2.0	34	
Arabic	Transliteration1	country	10	0	0		7.1	7.6	106	
Arabic	Translation2	Name	20	10	50	64	5.8	3.7	64	40
Arabic	Translation2	Address	15	6	40		6.5	2.8	43	

Arabic	Translation2	city	10	9	90		5.9	0.8	14	
Arabic	Translation2	country	10	10	100		7.1	1.6	23	
Arabic	Transliteration2	Name	20	0	0	0	5.8	2.5	43	68
Arabic	Transliteration2	Address	15	0	0		6.5	4.4	68	
Arabic	Transliteration2	city	10	0	0		5.9	2.6	44	
Arabic	Transliteration2	country	10	0	0		7.1	8.2	115	

6.5.4 Cyrillic Results

The Transliteration tools perform low for Cyrillic script (mostly Russian language) as well. The disagreement is due to spelling conventions, even though the transliterated output may give similar pronunciation. Translation tools are able to perform better, with up to 86% accuracy on the data. Though some tools perform better for Names and others for country or city, they all consistently perform worst for addresses, mostly because of spelling conventions or decisions to translation of meaningful words (translating where it is not needed, or not translating where it is needed).

Levenshtein distance gives more detailed view of the comparison. Even though translation tools give much better accuracy figures, the word level Levenshtein distances are comparable to between different transformations. This is because both Cyrillic and Latin systems are alphabetic. Translation tools still perform slightly better giving a low distance of 31% overall.

Table 6.5. Detailed Testing Results for Cyrillic Script

Language	Tool	Type	No. of Items	No. of Correct Items	Accuracy (%)	Overall Accuracy (%)	Avg. No. of Characters per Word	Avg. Levenshtein Distance per Word	Levenshtein Distance per Word (%)	Overall Levenshtein Distance (%)
Cyrillic	Specialized1	Name	20	11	55	34	7.1	3.2	45	58
Cyrillic	Specialized1	Address	14	1	7		7.2	4.8	66	
Cyrillic	Specialized1	city	6	1	17		9.2	5.5	60	
Cyrillic	Specialized1	country	10	4	40		6.7	3.9	58	
Cyrillic	Translation1	Name	20	15	75	76	7.1	1.8	25	31
Cyrillic	Translation1	Address	14	8	57		7.2	2.5	35	
Cyrillic	Translation1	city	6	6	100		9.2	3.9	43	

Cyrillic	Translation1	country	10	9	90		6.7	0.2	3	
Cyrillic	Transliteration1	Name	20	6	30	22	7.1	1.1	16	38
Cyrillic	Transliteration1	Address	14	0	0		7.2	3.8	53	
Cyrillic	Transliteration1	city	6	4	67		9.2	3.8	41	
Cyrillic	Transliteration1	country	10	1	10		6.7	2.1	31	
Cyrillic	Translation2	Name	20	18	90	86	7.1	1.3	19	31
Cyrillic	Translation2	Address	14	10	71		7.2	2.5	35	
Cyrillic	Translation2	city	6	5	83		9.2	3.6	39	
Cyrillic	Translation2	country	10	10	100		6.7	1.4	21	
Cyrillic	Transliteration2	Name	20	5	25	18	7.1	1.3	18	43
Cyrillic	Transliteration2	Address	14	0	0		7.2	4.1	56	
Cyrillic	Transliteration2	city	6	3	50		9.2	3.9	43	
Cyrillic	Transliteration2	country	10	1	10		6.7	3.5	52	

6.5.5 Cumulative Results

The data shows that the accuracy of transformations differs across the categories (name, address, city/state, country). Table 6.6 shows a summary of the Levenshtein Distance accumulated for each category and script over all tools tested. As has been discussed transformations are the most inaccurate for addresses, due to ambiguity with translation/transliteration of generic terms and meaningful words and also due to word order differences, beyond the other character level issues discussed for transliteration. There are multiple forms in which country-names can be written, including acronyms (US or USA), short form (United States) or long form (United States of America). A main reason that the country category shows high level of inaccuracy is because the desired form is not specified for the tools and for the informants, causing arbitrary differences. Thus, beyond other errors discussed, this additional variation contributes to the error. The transformation of names and city/state is almost twice as accurate, overall.

Table 6.6. Levenshtein Distance (%) for all Categories and Scripts

Levenshtein Distance (%)	Han	Devanagari	Arabic	Cyrillic	Average
Name	26	28	38	25	29
Address	61	51	60	49	55
City/State	19	30	27	45	30
Country	63	-	72	33	56
Average	42	36	49	38	

Table 6.7 accumulates the accuracy and Levenshtein distance across translation and transliteration tools (over all categories). It is evident that translation tools (66% accuracy; 38.5% distance) are much more accurate than transliteration tools (16.3% accuracy; 53.7% distance).

Table 6.7. Summary of Tool and Type of Transformation

	Type	% Overall Accuracy	Average of % Accuracy	% Over all Lev. Dist.	Average of % Lev. Dist
Transliteration1	Transliteration	10	16.3	50	53.7
Transliteration2		9		55	
Specialized1		30		56	
Translation1	Translation	66	66	37	38.5
Translation2		66		40	

Table 6.8 gives a summary of percent accuracy of each tool across all categories of data for reverse transformation for each language. This process is not supported for all language and for all tools, as indicated in the table (missing data indicated by '-'). The Round-Trip transformation gives the accuracy of source-to-target-to-source-language transformation by the same tool. The Reference transformation gives the accuracy of gold reference to source language transformation. The latter resembles the cases when either the user types in the data or the data is provided by a third language data transformed into Latin English (for eventual transformation to source language, using Latin English as a pivot, as discussed earlier in the report). The results show that the re-transliteration accuracy can be fairly accurate if the transliteration scheme is followed accurately (as shown by Round-Trip results), which is not easily possible for human users, as it is normally highly decorated Latin. Otherwise, (as shown by Reference transliteration) the results are not accurate. For translation, the Round Trip accuracy and the Reference

accuracy are comparable, though generally latter is more accurate, because in source-target-source language translation the errors accumulate but not in the case of translation from the reference data.

Table 6.8. Summary of Accuracy of Reversibility of Transformation for Round Trip and Reference

Accuracy %		Han	Devanagari	Arabic	Cyrillic
Transliteration1	Round Trip	-	84	100	89
	Reference	-	0	0	16
Transliteration2	Round Trip	-	-	-	-
	Reference	-	-	-	-
Specialized1	Round Trip	0	-	0	-
	Reference	3	-	0	-
Translation1	Round Trip	56	52	75	65
	Reference	50	58	84	84
Translation2	Round Trip	41	44	65	78
	Reference	38	28	78	82

7 Analysis of Transliteration and Translation

The transliteration results show that there are real gaps between what the tools output and what is desired by the human user (as the gold reference). On the other hand, the translation results are closer to what is desired by the human user (as the gold reference). This section looks into details of the processes and the errors in these processes, highlighting what works well in each case and what are the challenges faced in these transformations.

7.1 What Works for Transliteration

7.1.1 Consistency of transformation

Transliteration is consistent, because data is always transformed the same way. Any letter or word coming in any context will always give the same result. Though this is not good for transforming general language, this consistency is very useful for contact information which largely comprises of proper nouns.

Further, as the mapping at character level is pre-defined, the accuracy of the system for a language pair can be gauged (and adjusted). This assessment can be done reasonably deterministically and is not dependent on unseen or new words which may come up in the data, as the latter will also predictably follow the same transliteration mechanism. This is contrary to the translation process, which is not predictable on new or unseen data.

For example, the character 李 always maps to “li” in Han Chinese, e.g. 李肃 as “li su” and in 李克强 as “li ke qiang”. This mapping will not change in context. The letter र always maps to ‘r’ in Latin English, e.g. both times it occurs in रणथम्भौर, which transliterates to “raṇathambhaura”. In Arabic Urdu the و transliterates to ‘w’ both times it occurs in مولوی giving the output “mwlwy̰” even though it should transliterate first occurrence to a vowel and only second to a consonant (note that even though the output is incorrect, the system behaves predictably and consistently). In Cyrillic Russian the letter ‘С’ would always map to the letter ‘S’ in Latin English, independent of any context.

This consistency is not only within words, but also in phrases, where the number of words in the transformed output always matches the number of words in the source language. Again, for proper nouns, this regularity is a useful property for a transformation process.

7.1.2 Fall-back Options

One of the reasons transliteration is preferred over translation is because it allows for fall back options, in case source language to target language mapping is not available, with multiple options, eventually falling back to source script to target script option, as discussed earlier in this report. The fall back options may not be as accurate but are better than getting no output at all.

For example, if transliteration for Sindhi is not available in Arabic script, it may be possible to fall back to Urdu or Persian language mapping or even to a generic Arabic script mapping (if one exists). Such mapping may still give usable results versus no output at all. However, in such a case, some letters may get mapped wrongly or remain unmapped, as the character set for a language may contain extra letters not covered in fall back options. So Sindhi language letters ت ،پ ،ب will map to ‘t̲’, ‘p’ and ‘b’ even based on mapping from Urdu or Persian languages. However, the letter ٻ in Sindhi would not be mapped as it does not exist in either Urdu or Persian. Similar schemes for Cyrillic or Devanagari scripts may be designed in case a particular language mapping scheme using these scripts is not available.

7.1.3 Extensibility to Languages within Scripts

Based on similarity across languages discussed, transliteration systems provide easier scalability across languages within a script. If a mapping scheme between a pair of scripts and for certain languages within these scripts already exists, it would be possible for a community to work to extend such systems to new languages within these scripts. The effort may vary across languages, by such effort will likely be much easier than developing a translation system between such languages.

For example, as the transliteration system is available between Arabic, Persian or Urdu languages using Arabic script into Latin English, Sindhi language community can work to extend these mappings to develop a system for Sindhi to English. This would require doing a character by character analysis and determining the equivalent Latin characters, and existing transliteration tables can be used to hint the process. This can be done in a fairly manageable time for getting a functional solution (though standardization of such mappings may require considerable time at national and/or international levels).

7.2 What is Challenging for Transliteration

7.2.1 Diversity in Writing Systems

As has been discussed in detail earlier, there are inherent differences in writing systems, which create challenges for transliteration across them.

For Han script, the pinyin system is used which maps each character of Chinese to an "equivalent" sounding syllable in Latin English. There is a fairly regular mapping defined for each character, making the process consistent. This mapping is done for both simplified and traditional Chinese. However, the pinyin systems in mainland China and Taiwan differ, making the transcription scheme depending on geographical region. The decisions for capitalization and space insertion for the pinyin system are quite complex and different from Latin English conventions. Finally, the tonal system in Chinese language is represented with diacritics, which would not be easy to understand by non-Chinese speakers, especially those speaking non-tonal languages.

Devanagari writing system writes out vowels and consonants and is regular in doing so, making it easy to map on an alphabetic system. However, a complication arises because each consonant has an inherent vowel if a vowel is not written explicitly, but in some cases inherent vowels is suppressed and not spoken (e.g. for coda consonants). In Hindi the inherent vowel is suppressed using the Halant combining mark, but this mark is normally not written and readers use their knowledge to decide whether the inherent vowel should be pronounced. However, a one to one transliteration system is configured to write out all such vowels if the Halant is not written out. So प्रसाद is transliterated as "prasada" instead of the reference "prasad".

In Arabic writing system only consonants are specified and vowels are optional diacritics on these consonants and generally not written. So any transliteration tool will only give romanized form of these consonants, for example, محمد (Muhammad) transliterates to "mhmd. This output is unreadable for a general user, especially if s/he is not familiar with the source language. Even people familiar with the source language may find it hard to guess unfamiliar names. Further, many languages using Arabic writing system normally use the three letters ی ،و ،ا as both consonants and vowels in different contexts. As it is a consonantal writing system, transliteration systems, which do not take context into account, map these letters to consonants (generally /', v or w, y/), thus generating extra consonants when these letters occur in vocalic contexts, e.g. the و in مولوی is vowel in the first instance and consonant in the second instance (Maulwi) but transliterates to consonantal version in both cases: giving "mwlwy".

Cyrillic script is an alphabetic script. Therefore, it has an easier mapping on Latin alphabetic system.

A common challenge in many transliteration schemes across many scripts is that decorated Latin (i.e. with diacritical marks) is used to accurately capture the variety in speaking and diction conventions. Transliterating to Latin with diacritics has two significant implications. First, the transliteration becomes less readable by end users as it requires training to understand the diacritical system (and also because they are used in a very ad hoc manner and their used differs greatly across languages and scripts). Second, it is not possible to represent such transformed data in ASCII, which limits its use, e.g. it cannot

be represented within current WHOIS data specifications even after being transformed to Latin. Some transliteration schemes try using either capitalization or other non-letter marks to address this challenge. However, in this case, capitalization only provides with two alternatives, whereas in scripts and languages more than two possibilities may exist. For example Arabic Urdu the sound /z/ (in English 'z' or 'Z') is represented with four letters: ز، ذ، ظ، ض. Using capitalization also confuses the Latin readers as it produces capitalization in the middle of words, and capitalization of name, etc. can cause representation errors. Other option is to use alternative characters like ~ or @, but that makes the output less readable.

7.2.2 Variation across Languages

Transliteration is also dependent on the language. Same letters in source script can map to different Latin letters as they are pronounced differently across languages. Thus, it is important to know the source language for accurate transliteration. General script level transliteration systems will be inaccurate across languages. For example, the letter 'j' is pronounced as /dʒ/ in English and /j/ in French. Letterض is pronounced similar to /dˤ/ in Arabic language but /z/ in Urdu. Therefore, language dependent transliteration tables are required for accurate transliteration and general script level tables will be inaccurate, but in many cases, such tables are not available.

The character set of languages within a script may also significantly differ. In such cases, if a language does not have a transliteration scheme available, and has to fall back on the script level transliteration scheme, the latter would not know how to handle extra characters in a specific language, causing untransliterated or wrongly transliterated content. For example, Heh Dochashmee used in Urdu, but not in Arabic language, is not transliterated in the name چودھری giving "cẖwdھryۍ" which is a mixed script output. Such output allows for accurate re-transliteration, but is neither understandable by the target users nor can be represented in ASCII format.

Further, multiple transliteration schemes are available for many languages. Some are historic and widely used, but are not adopted as clear standards. Alternatively, in many cases the standard transliteration schemes are different from those which widely used in practice. The accuracy of these transliteration schemes can also vary. The current results are based on arbitrarily chosen schemes, where multiple schemes were available.

7.2.3 Inability to Capture Necessary Translation

When contact information is transformed into another script and a language, much of the information is transcribed, that is it has a similar sounding representation. Thus, transliteration schemes (which are generally motivated by sound) can represent proper names. However, in some cases, the sound based representation does not work. This is true especially in the case of some proper names (including country names, some city names) and for generic terms in all categories of contact information. For example, even if the transliteration errors are rectified for country name مصر by adding the intended vowel to change the transliterated form from "msr" to "misr", it would still require native Arabic language knowledge to understand that "misr" represents Egypt. In Han Chinese, Korea is written as 韩国 in simplified form and 韓國 in traditional form. However, transliteration tools output "Han Guo" in

both cases. In Cyrillic Russian, China is written as Китай which the transliteration systems convert to "kitaj" or "kitai".

This is especially true for generic terms in names (titles), organization names and addresses. A transliteration system would not be able to translate, causing errors. As an example from Simplified Chinese, 北京市海淀区学院路37号 which is transliterated to "běi jīng shì hǎi diàn qū xué yuàn lù37hào" whereas the gold reference is "37 Xueyuan Road, Haidian District, Beijing". "Road" and "District" are not translated in this case. A similar example in Devanagari Hindi transliterates "अलेप्पी शॉप - आयुर्वेद होस्पिटल के सामने" to "alēppī śŏpa - āyurvēda hōspiṭala kē sāmanē" whereas the reference is "Alleppey shop - opposite ayurveda hospital". The phrase "kē sāmanē" translates to "opposite" in the reference. In Cyrillic Russian проспект Ленина transliterates to "prospekt Lenina" instead of "Lenin Avenue". As a more extreme case, the complete address in Arabic language شارع المركز المالي is changed in reference to "Financial Center Road" but is transliterated to "sẖạrʿ ạlmrḵz ạlmạly" which is not comparable in understanding by the end user.

7.2.4 Difference in Word Order

The word order in contact information is different due to linguistic and cultural differences. This is especially true for addresses, and has been one of the main reasons address transliteration has be so inaccurate. For example, in Chinese high level information precedes detailed information in addresses, but is vice versa in English. This mis-alignment also causes transliteration errors (when compared to the gold reference). For example, 北京市朝阳区西坝河光熙门北里甲31号 is transliterated to "bei jing shi chao yang qu xi ba he guang xi men bei li jia 31hao" but the gold reference is in the reverse order as "jia 31north guangximen xibahe chaoyang district beijing 100028 china". This is also true in many other languages. Taking the previous example from Devanagari Hindi "अलेप्पी शॉप - आयुर्वेद होस्पिटल के सामने" transliterates to "alēppī śŏpa - āyurvēda hōspiṭala kē sāmanē" whereas the reference is "Alleppey shop - opposite ayurveda hospital". The phrase "kē sāmanē" comes at the end of the address in the transliteration but "opposite" comes in the beginning of the second phrase in the reference. Similar example in Arabic language is أبراج الإمارات for which the reference is "Emirates Towers" but the transliteration gives the word order in the Arabic language, "ậbrạj ạlậmạrạt". Even though Cyrillic Russian has similar word order in many cases, it also has differences in some cases, e.g. булевард Тодор Александров transliterates to "bulevard Todor Aleksandrov" whereas the reference is "Todor Alexandrov Boulevard".

However, this fixed nature of the transliteration system has a strength that it would not skip any words, and would always give an exact word to word transformation.

7.2.5 Variation in Romanization

Another significant source of error is that romanizaiton from the source language may have multiple ways of representing certain sounds in Latin, which may be context dependent or even subjectively different, i.e. arbitrarily or predictably one spelling may be preferred over another, even if the output sounds the same. A transliteration scheme may not be able to match the desired result in all cases. For

example in Han Chinese 謝長廷 is romanized to “Xie Zhang ting” where as gold reference is “Hsieh Chang-ting”. In Devanagari Hindi, सिकंदराबाद is transliterated to “sikandarabada” where as the gold reference provided is “secunderabad” which differs in both consonants and vowels, even though it represents the similar pronunciation. Example from Cyrillic Ukranian is Андрій which transliterates to “Andrii” or “Andrij” but the gold reference provided is “Andriy”.

7.2.6 Re-transliteration of General Form

The accuracy of re-transliteration is dependent on the romanized form being given to the system. The options are either (i) the romanized form conforming to the formal transliteration scheme being used by the tool, or (ii) general romanization using some ad hoc scheme as input by users or romanized from from a third language. In the first case the re-transliteration is very accurate. However, unfortunately such formal input is generally not available.

For example, in the case of Arabic script, for محمد a desired input scheme requires user to enter “mḥmd” but the user would generally enter “Muhammad”; the latter would get converted back to مُهَمَد which is inaccurate representation in Arabic language.

These errors are caused due to multiple factors. First, the consonants in Latin may be under-specified, without diacritics, so they get mapped to different consonants in Arabic script. ‘ḥ’ is different from ‘h’ as the former is transformed into ح whereas the latter is transformed into ه. Second, the digraph representation for some sounds in Latin script, e.g. “kh” for the velar fricative /x/ or خ transliterates into two different consonants representing ک (from ‘k’) and ه (from ‘h’) instead. Second, the same English vowels may represent short or long vowels in languages using Arabic script. Transliteration is not sensitive to such differences and will transliterate both cases to the same short vowel, making an error if the long vowel was desired. If a vowel in romanized form is repeated to indicate length, It may also produce erroneous re-transliteration by generating two consecutive combining marks, which is not possible (as a consonant cannot be associated with two vowel combining marks). Two consecutive (and erroneous) marks may also be created if two consecutive vowels are written out, e.g., the name “Saim” transliterates to سَِم, which is an ill-formed sequence.

Another significant challenge with re-transcription of general form, as provided in the reference data, is that it contains translated terms (as discussed earlier an earlier section). It is not possible to transliterate such terms back to the original. For example, it is not possible to recreate Arabic language words نفق “nfq” from “tunnel”, شارع “shar’” from “road”, المركز “ạlmrḵz” from “center”, etc., found in the gold reference through the re-transliteration process. Similar issues would exist in other languages being re-transliterated into other scripts.

7.3 What Works for Translation

7.3.1 Independence from Writing System

As translation system allows for mapping multiple characters from the source to multiple characters to the target, it is not as dependent on the writing system as much as a transliteration system would be(latter normally based on character to character mapping).

In Han script translation gives proper names with appropriate spacing, instead of code point level transliteration. For example, partial address 北京中关村 would translate correctly to “Beijing Zhongguancun” instead of “běi jīng zhōng guān cūn” which are individual pieces given by the transliteration process, latter requiring the user to guess what is the right combinations. Transliteration can result in ambiguity and eventually wrong interpretation of longer addresses.

For Devanagari, this allows for the correctly identifying where to suppress inherent vowel ‘a’. So for the words like जसवंतसिंह, the translation process correctly gives “Jaswant Singh” where the transliteration either cannot suppress any inherent vowels giving “jasavantasinha” or suppresses all inherent vowels giving “jsvNtsiNh”, both of which are incorrect.

Translation is a more accurate way of transforming Arabic script, as it allows for full consonantal and vocalic representation in Latin English, making the transformation more readable compared to the consonant-only transliteration. The name محمد translates to “Mohammed” instead of the transliteration version “mhmd”.

7.3.2 Meaningful Transformation

Translation is more meaningful in target language and therefore gives a better option for end user. It allows for both options, to convert to a complete “equivalent” transliterated version (as discussed above) and to convert to target language. This mix allows for getting the best results.

For example, in Han Chinese is translated to “United States” instead of its transliteration “Mei Guo” but is translated to “Beijing” which is same as its transliteration “běi jīng”. This process is equivalent for both traditional and simplified Chinese (for translation systems trained on both types writing). This is also true for generic terms in addresses, for example translates to “Guangdong Province” whereas the transliteration gives “guǎng dōng shěng”, latter not understandable by end users. Similarly, ब्लड बैंक के निकट in Devanagari Hindi translates to “blood bank near” instead of “blaḍa baiṅka kē nikaṭa” (the translation is correct, except the difference in word order). For Arabic Urdu address رضا بلاک، جوہر ٹاؤن، ملتان روڈ translates correctly to “Raza Block, Johar Town, Multan Road” but the transliteration “rD blkh, jwhr ttw'n, mltn rwdd” is not readable. In Cyrillic Russian, Китай translates to “China” but transliterates to “kitaj” or “kitai”. Similarly, 5-й этаж translates to “5th Floor” but transliterates as “5-j étaž”.

7.3.3 Re-ordering of Words

Translation process allows for the possibility of re-ordering the words from source language, giving a more compatible word order in the target language. Transliteration, on the other hand strictly follows the word sequence in source language. For example, the transliteration of Han Chinese for the address 北京市海淀区学院路37号 is “běi jīng shì hǎi diàn qū xué yuàn lù37hào” which is not the convention used in Latin English. The translation is able to reverse the order giving “No. 37 Xueyuan Road, Haidian District, Beijing” to meet the expectation of Latin English user. As discussed earlier, Devanagari Hindi “अलेप्पी शॉप - आयुर्वेद होस्पिटल के सामने” transliterates to “alēppī śŏpa - āyurvēda hōspiṭala kē sāmanē”. The phrase “kē sāmanē” (meaning in front of) comes at the end of the address in the transliteration but

comes correctly in the beginning in the translated version “Allepey shop - in front of Ayurveda Hospital”. Similar example in Arabic language is أبراج الإمارات translation correctly reverses the word order to “Emirates Towers” but the transliteration gives the word order in the Arabic language, “ạ̈brạj ạlạmạrạt”. In Cyrillic Russian проспект Ленина transliterates to “prospekt Lenina” but translation gives the right word order “Lenin Avenue”.

7.4 What is Challenging for Translation

7.4.1 Language Dependency

One of the main constrains of the translation technology is that it is language dependent and cannot work at the level of script. Thus, it is not possible to have Han to Latin or Cyrillic to Latin translation, severely constraining any back off process in case the translation technology for a particular language pair is not available. Though translation is available for more popular languages, it is not available for a variety of languages across scripts.

Further, the accuracy of translation may vary greatly across languages as well, even when it is available, and may perform poorly compared to the transliteration process.

To improve accuracy of existing language pair or to add another language pair, the effort needed for transliteration is to define a mapping between letters, which is far less involved process compared to developing a translation system between two languages, latter requiring millions of words of parallel language corpus and follow up processing for accurate results. The effort for developing the transliteration system may be undertaken by a community based working group, however the work for developing a translation system requires specialized effort.

Another challenge is that translation it is not a deterministic process like transliteration (which can do letter by letter transformation), so it will only work well on words familiar to the system, and not on unseen or new words, e.g. foreign names and addresses. Therefore, in such cases the translation will have to fall back on transliteration and therefore will not be any more accurate. However, the fall back will only be possible if the transliteration system is integrated in the translation system.

This is indicated by examples from the data tested for some of the languages. One of the translation tool tested does not support Marathi Language in Devanagari script and thus no output is received. The Arabic Urdu city name ڈیرہ اسمائیل خان is not completely recognized by one of the translation systems and is left partially unconverted to “DERA اسمائیل Khan” (the fall back transliteration option is not invoked). Similarly, a tool is not able to translate Ukranian name Матюх and outputs “Mat&#xFFFD;h”.

7.4.2 Context Dependent Translation of Meaningful Words

Same word may be translated differently in different contexts. Though that is generally preferred in general language translation, it may not be the best mechanism in translation of contact information, which would be better served if the translation of a word is stable and predictable across different contexts. Further, some meaningful proper names are translated instead of transcribed making the translation incomprehensible. This is especially true for the context of IRD which mostly contains proper names.

For example, in Han Chinese the reference provided for 高雄市新興區七賢一路1巷1弄1號 is “No.1, Aly. 1, Ln. 1, Qixian 1st Rd., Xinxing Dist., Kaohsiung City 800, Taiwan (R.O.C.)” however, the translation system gives “Seven Sages of the way an emerging area of Kaohsiung Lane 1, Lane 1”, not being able to correctly ascertain that the data should be transliterated. Similarly, Devanagari Hindi name चन्द is translated to “one of two” (few) instead of the tranliteration “chand” given as reference. The name of ام city should be transliterated to “Umm” but is translated to “or” in Arabic language. Same is observed in Cyrillic Russian where Железнодорожный переулок is translated to “Railway Lane” instead of the gold reference “Zheleznodorozhny by-street” in which the first word is not to be translated.

7.4.3 Lack of One to One Mapping from Source to Target

Sometimes the translation does not result in one to one word translation and either some words are not translated or sometimes extra words are inserted. This is based on the learning model for the translation and, though it may work in some cases, generally causes inaccuracy for contact information. The testing results from Han Chinese shows that the name 王文志 is translated as “Wang Zhi” instead of “wang zhi wen”. Similarly the address in Arabic طريق القاهرة - الإسكندرية الصحراوي is translated to “Cairo - Alexandria Desert” instead of “Cairo - Alexandria Desert Road.”

7.4.4 Reversibility

Though re-transliteration process is accurate if the input is in the prescribed scheme, translating a label back to source language based on target language is generally less accurate, if the roman form is already created through a transformation process and is not manually generated. This is because if this is through a transliteration process, the retranslation is not trained to handle such input; or if this is through translation then the translation error in the initial translation may get compounded in the reverse process. For example, in a translation of Devanagari Hindi the name चन्द is interpreted as “one or two” instead of the name “Chand”. The reverse translation make the literal translation into एक या दो. Similarly, for Arabic script the name صائم is translated to “fasting” instead of being interpreted as a name. The translation back gives a different morphology of the work صيام. In Cyrillic, the name Петровская is translated into a short form “Peter” instead of the correct representation “Petrovskaya” and therefore reverse translation gives Питер. However, for manually generated data, reverse translation may be reasonably accurate.

Further, it is not necessary that if translation into Latin is supported, translation back is also supported by a tool. Even if such support is available, the accuracy in the two directions may not be the same.

8 Summary of Findings

The analysis of the transformation of contact information, including person names, addresses, city/state and country, show variability in output on multiple factors, including script, language, category of information, type of transformation and the tool used. The current section aims to extract some high level finding, sifting through the data presented earlier.

8.1 Existing Practices and Protocols

1. A few e-commerce sites allow users to input some contact information in different languages and scripts. Sometimes they limit input to the character set used by the target user population, though such limitations are not consistent even within the same website, e.g. password field may require ASCII even if other fields allow other scripts, or phone numbers do not allow local language digits.
2. E-commerce sites do not transform contact data information provided by the users across languages or scripts (except in its encoding), and expect the users to input correct information in the script and language in which the data will be understood by those responsible for the product delivery channel.
3. Registries and registrars are collecting information in local languages and sometimes in both local language and its romanized form (romanized form is required by the Registrar Accreditation Agreement (RAA) even for IDN registrations). Consistency between the two versions is not verified.
4. None of the registry and registrar who responded to the survey is transforming the contact information in the registration data. Where multiple language data is collected, it is provided directly by the registrant.
5. Support of internationalized registration data is variable across the processes and systems used by registries and registrars.
6. WHOIS only supports ASCII and does not support internationalized contact data. EPP supports UTF-8 encoding, through which internationalized contact data may be transmitted and received, but without specification of the language. Moreover, EPP does not seem to be able to record multiple linguistic versions of the same data. RDAP can also encode language information and can handle multiple versions in parallel.
7. These protocols cannot document the method and language/script with which data has been acquired and history of (any) transformation(s) it has undergone to get to its current form.

8.2 Transformations and Tools

8. Transliteration works better for proper names (e.g. person or organization names, cities) but not for items which have common nouns (addresses) or alternate names in other languages (e.g. country names). Translation works for all four categories, but is highly dependent on maturity of the tool being used. More structured data, e.g. as per UPU guidelines, improves transformation accuracy and ordering.
9. Translation is a better option than transliteration for the language pairs which have well trained systems as it can handle word order difference and translation of common nouns

10. The following information for the contact data is needed for conducting transformation or re-transformation:
    a. Current language and script
    b. Method of obtaining current data (manual or transformed)

    If the data has been transformed already, the following additional information needs to be recorded:
    c. Source language and script
    d. Type of transformation (translation or transliteration)
    e. Mechanism of transformation (manual or automated)
    f. Standard used for the transformation (for transliteration)
11. One single tool may not work for all contact information, because the accuracy of tools varies with different types of contact information: Those working well for proper names may not necessarily work for addresses; those specialized for addresses may not work well for proper names; etc.
12. Transliteration is usable for scripts which fully specify consonants and vowels. It does not work for scripts where consonants or vowels are either not given or under-specified.
13. Ad hoc transformation using translation systems give an arbitrary output. However, results show that even though it does not predictably convert character by character, its overall output is much more readable and independent of the scripts of the language pair involved. Therefore, they perform better from an end-user perspective. However, this is only true for a limited set of language pairs which have mature automatic translation systems. Currently a limited set of such systems exist (for tens of different languages, with varying accuracy). Making a new translation system for a language pair is very challenging.
14. Consistent transformation is possible through transliteration but compromises the comprehensibility of the information; especially between scripts which encode information differently. Transliteration can still be inconsistent if different standards are used or if different tools are used which do not fully conform to such standards.
15. Though pivoting through romanization presents an interesting possibility to provide local language to local language transformation, the two levels of transformation involved may make the output very inaccurate for effective use, given the variation in transformation techniques and tools.
16. Accurate transformation is not possible through automated processes and would require manual effort, including registrant verification in at least some cases (where spelling variation is possible). Therefore, if the use of the registration data requires precise and accurate transformation, such as for trademark and legal enforcement cases, then no automated tools can be satisfactorily used.

9 Conclusions

This report documents and assesses the current solutions for submitting or displaying internationalized contact data with the aim to help the ICANN community to examine the possibility of transformation of such data for its broader accessibility. The study looks at the current practices of handling internationalized contact data by e-merchants, registries and registrars. It also determines the support of such data by relevant protocols. Finally it evaluates the accuracy of transforming internationalized contact data, including names, addresses, cities/states and countries. The study concludes by presenting the findings around the practice of collection and documentation of contact data, the readiness of protocols and the accuracy of transformation tools for further consideration. The study has found that provisioning and querying protocols are lacking either support or deployment for internationalized registration data, and that the tools tested are not providing a high level of transformation accuracy and consistency of internationalized registration data.

Acknowledgements (in alphabetical order by first name)

The authors would like to recognize the contribution of following in this study:

Akshat Joshi
Abdulaziz H. Al-Zoman
Adel Riyad
Bashar Al-Abdulhadi
David Hall
Dennis Tan Tanaka
Dmitry Belyavskiy
James Galvin
Jody Kolker
Michael Yakushev
Naoki Kambe
Neha Gupta
Nishit Jain
Olga Baskakova
Peter Koch
Qichao
Stéphane Bortzmeyer
Stuart Olmstead-Wilcox
Zain Al Abdeen Baig

Authors (in alphabetical order by first name)

Guillaume Leclanche, Viagénie, Canada
Marc Blanchet, Viagénie, Canada
Sarmad Hussain, Al-Khawarizmi Institute of Computer Science, UET, Pakistan
Simon Perreault, Viagénie, Canada
Steve Sheng, ICANN Staff

References

Alkharashi, I. A. (2009). "Person Named Entity Generation and Recognition for Arabic Language," in the *Proceedings of 2nd International Conference on Arabic Language Resources and Tools,* Cairo, Egypt.

Atoui, B. (2012). "The Issue of the Romanisation System for the Arab Countries: Between Legitimacy and Parctices. Which Solutions?" in the *Tenth United Nations Conference on the Standardization of Geographical Names.* Economic and Social Council, United Nations, New York. Accessed on 15th Feb. 2014 from http://unstats.un.org/unsd/geoinfo/UNGEGN/docs/10th-uncsgn-docs/econf/E_CONF.101_96_The%20issue%20of%20Romanization.pdf.

Beesley, K. (1998). "Arabic Morphological Analyzer – Romanization, Transcription and Transliteration." Xerox Inc. Accessed on 15th Feb. 2014 from http://open.xerox.com/Services/arabic-morphology/Pages/romanization.

BGN (1994). *Romanization and Roman Script Spelling Conventions,* US Board on Geographic Names, Defense Mapping Agency, Fairfax, Virginia, USA.

http://libraries.ucsd.edu/bib/fed/USBGN_romanization.pdf

Huang, F. (2005). "Cluster-specific Named Entity Transliteration," in the *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing,* pp. 435-442, Association for Computational Linguistics, Stroudsburg, PA, USA.

Phillips, A. and Davis, M. (2009). "Tags for Identifying Languages," IETF. Accessed on 15 Feb. 2014 from http://tools.ietf.org/html/bcp47.

Pouliquen, B., Steinberger, R., Ignat, C., Temnikova, I., Widiger, A., Zaghouani, W., Žižka J. (2005). "Multilingual person name recognition and transliteration," in *Journal CORELA - Cognition, Représentation, Langage. Numéros spéciaux, Le traitement lexicographique des noms propres.*

SAC054 (2012). "SSAC Report on the Domain Name Registration Data Model ," published by the Security and Stability Advisory Committee, ICANN. Accessed on 7th May 2014 from http://www.icann.org/en/groups/ssac/documents/sac-054-en.pdf.

UNGEGN (2002). *Glossary of Terms for the Standardization of Geographical Names.* United Nations Group of Experts on Geographical Names, Department of Economic and Social Affairs, Statistics Division, United Nations. Accessed on 18th Jan. 2014 from http://unstats.un.org/unsd/geoinfo/ungegn/docs/pdf/Glossary%20of%20terms_revised.pdf.

UNGEGN (2003). "Report on the Current Status of United Nations Romanization Systems for Geographical Names," compiled by the UNGEGN Working Group on Romanization Systems

UNGEGN (2006). *Manual for the National Standardization of Geographical Names.* United Nations Group of Experts on Geographical Names, Department of Economic and Social Affairs, Statistics Division, United Nations. Accessed on 18th Jan. 2014 from http://unstats.un.org/unsd/publication/seriesm/seriesm_88e.pdf.

UNGEGN (2007). *Geographical names as vital keys for accessing information in our globalized and digital world.* United Nations Group of Experts on Geographical Names, Department of Economic and Social Affairs, Statistics Division, United Nations.

UNGEGN (2011). "UNGEGN List of Country Names," by UNGEGN. Accessed on 23rd Feb. 2014 from http://unstats.un.org/unsd/geoinfo/UNGEGN/docs/26th-gegn-docs/WP/WP54_UNGEGN%20WG%20Country%20Names%20Document%202011.pdf.

UNGEGN (2013). "Report on the Current Status of United Nations Romanization Systems for Geographical Names," by Working Group on Romanization Systems, UNGEGN. Accessed on 16th Feb. 2014 from http://www.eki.ee/wgrs/ (also see individual language pages available through further links).

Unicode. "Unicode Transliteration Guidelines," available at http://cldr.unicode.org/index/cldr-spec/transliteration-guidelines.