HYBRID MACHINE TRANSLITERATION SYSTEM FOR ENGLISH TO KASHMIRI WITH DIACRITIC DEPENDENCY REGULATION

Sameer ul Rahman[1], Er. Shilpa[2]

[1]PG-Scholar, [2]Asst. Prof., Department of Computer Science and Engineering, RBU

Abstract

The maintenance of phonological characteristics of words during the script conversion is achieved via transliteration. Natural language processing can be used for transliteration to computationally create and speed up letter to letter conversion of words for people from different backgrounds to read any article in its own language. This can be used to transliterate whole of the document or any article on a click to any language whose transliteration is available.

This research is based on transliteration of pure English script into Hindi, Urdu and Kashmiri scripts, along with a detailed explanation of concepts and key processes. The method is not a direct transliteration but combination of 'phoneme' based and 'grapheme' based transliteration. That is, the word gets mapped to corresponding sound as well as graphed to its closest counterpart in other language.

Keywords: NLP, Grapheme, Phoneme, Transliteration, Diacritic,Yii, Apache, SQL.

1. Introduction

1.1 TRANSLITERATION:

Currently there are approximately 6500 in a different way languages spoken across the world. As a lot because it diversifies the splendor of the world, it receives similar struggle for the human beings from unique topographical areas to engage among each-other. Transliteration is a potent and effective mechanism to slim this gap.

Transliteration is letter to letter conversion from one language to another. Unlike translation, which tells you the meaning of a word, transliteration helps us pronounce as well as graph the alphabets of a word from one language to another. It puts up a word of similar sound of the two languages and helps to pronounce them amongst each other. For example, in Urdu language, we write "کل". This in English may be written as "kal". The letter "k" in English is synonymous in sound to as of the letter "ک" in Urdu and so on.

The unavoidable deprivation of the sense of a term in transliteration paves a manner for higher recognition of cross-ethnic translation ambience [1].

Fig 1: Transliteration.

1.2 NATURAL LANGUAGE PROCESSING (NLP):

Natural language processing (NLP) is a subfield of artificial intelligence and computational linguistics that concerns computer system's interaction, interpretation, understanding and management of human language. In its pursuit to narrow the gap between human conversations and machine understanding, NLP allures from vast disciplines, such as artificial intelligence, neural networks linguistic algorithms.

NLP specializes in the interplay among records technology and human language, and is scaling to masses of industries. Today NLP is booming way to the large upgrades with inside the get admission to records and the growth in computational power, which might be permitting practitioners to obtain significant consequences in regions like healthcare, media, finance and human resources, amongst others. The analysis and classification of data in emails with NLP is already in wield by the tech giants like Google, Yahoo and Outlook for spotting and eliminating spam before they flash in our inbox.

1.3 TRANSLITERATION USING NLP:

Natural language processing thus can be used as computational transliteration tool which will be a powerful way of interaction between humans and computers thus giving a new outlook to artificial intelligence. Depending how good any algorithm to map one language to other is, transliteration can go as close and appropriate to any work done manually by humans. The use of a Hybrid and Correspondence model for transliteration produces an apt outcome of target language transliterations as the source language phoneme and grapheme are utilized in there [2].

2. Literature Survey

The computational linguistics field is the source where use of latest approaches and advanced engineering seek to minimize the complexity and amplify the computer system's interaction, interpretation, understanding and management of human language. Machine Learning, Natural Language Processing Models, Data Mining etc. are the most common transliteration methods. An overview of transliteration models along with their highlights are mentioned here:

Noura Farra et al. [3], presented a universal inequitable miniature for spelling mistake rectification which aims at glyph- level conversions. They worked at the character level, while the system makes usage of term-level and contextual data. They implemented the system to correct fallacy in Egyptian Arabic dialect textbook, attaining 65% deduction in expression mistake rate over the input standard, and refining over the afore state- of- the- art model.

Ramy Eskander et al. [4], studied the conversion of unconstrained spelled Egyptian Arabic into a standard word-order. They flash that a two- phase operation can degrade bifurcations from this benchmark by 69%, making successive handling of Egyptian Arabic simple.Their methodology involves a blend of glyph transformations, entire- term transformations, and the application of a full syntactic tagger.

A Kumaran et al. [5], put forward a compositional machine transliteration model, where numerous transliteration elements may be drafted either to upgrade subsisting transliteration grade, or to empower transliteration performance between lingos in spite of lacking primary comparable nomenclatures corpora between them. Particularly, two distinctive configurations of constitution-Serial and Parallel are applied, utilizing a position of the craft machine transliteration structure in English and a block of Indian languages, specifically, Hindi, Marathi and Kannada. They show- cased that a CLIR network incorporated with compositional transliteration model performs invariably on grade than that incorporated with a straight transliteration model.

Soumyadeep Kundu et al. [6], put forward multiple frameworks for language self-supporting machine transliteration, implementing neural network grounded deep learning frameworks for the transliteration of titled objects. Their transliteration models adjust two distinctive Neural Machine Translation architectures: - Convolutional Sequence to Sequence based Neural Machine Translation and Recurrent Neural Network that dispenses relatively adequate outcomes when it comes to multi lingual machine transliteration.

Rama et al. [7], have treated the transliteration trouble as a translation challenge and have utilized expression based SMT methodology for English- Hindi language couplet. They accustomed SMT architecture, GIZA, beam search-based decoder for advancing the transliteration system and implemented English- Hindi aligned term corpus to train and try the model. Conclusions of offered model flaunt that these approaches can be successfully utilized for the assignment of machine transliteration at an attained preciseness of 46.3%.

Kumar et al. [8], have staged a Statistical Machine translating model to transliterate proper nouns penned in Punjabi vocabulary into its identical English dialect. The technique is to transliterate proper nouns of Gurumukhi script into its English counterpart nomenclatures. The model is sampled on varied names and is sampled on further 1000 names and system has generated a preciseness of 97%.

Dhore et al. [9], have proposed a phoneme-based system that transliterates Indian named objects into English implementing entirely consonant methodology and employing a hybrid (rule and metric grounded) pressure deconstruction technique for schwa omission. Their approach is direct without coaching any bilingual database and displays an in-depth knowledge of word setup in Devanagari script, focusing on Hindi and Marathi to English transliteration.

Malik et al. [10], have developed Punjabi Machine Transliteration System that's applied as to transliterate expressions from Shahmukhi dialect to Gurmukhi dialect. The elemental idea is glyph mappings and reliance regulations as only character mappings (applied at the beginning) aren't sufficient enough for the system. The developed model produces farther than 98% preciseness on traditional literature and 99% in case of modern literature.

Fehri et al. [11], have put forward a methodology for identification and translating of Arabic named entities based on a characterization model, a block of bilingual lexica and a block of transducers determining linguistic and dialectal prodigies related to the Arabic named entities. Their resources are reusable single-handedly as concluded by their trial and assessment validation of the model which is performed on the NooJ linguistic platform.

Jaleel et al. [12], presented an easy statistical approach to train an English to Arabic transliteration miniature from couplets of nomenclatures which they called an elected n- gram model because a two-phase routine operation. The model firstly learns which n- gram partitions should be appended to the unigram reservoir for the root language, and further in a successive phase, the translation model over that particular reservoir, without requiring any heuristics or lexical knowledge of either language. They assessed the statistically- trained model and a simpler hand- drafted model on a trial faction of named entities from the Arabic AFP corpus and substantiate that they pull off better than two online translation cradles and similarly researched the persuasion of these systems on the TREC 2002 cross language IR task. Conclusively they determined that transliteration either of OOV named entities or of all OOV expressions is a productive technique for cross language IR.

Table 1 Relative analysis of prevailing methodologies.

Ref.	Year	Dataset	Technique	Remarks
[19]	2017	English-Hindi dialect	Glyph- level conversions	Word accuracy of the proposed transliteration software has been found to be 70.22% as against 58.73% of Google Input tool as on Mar 04, 2017.
[3]	2014	Egyptian -Arabic dialect	Glyph- level conversions	Deduction in expression mistake rate over the input standard, and refining over the afore state- of- the- art model.
[4]	2013	Egyptian -Arabic scripts	Blend of glyph transformations, Full syntactic tagger.	A two- phase operation can degrade bifurcations from the benchmark by 69%, making successive handling of Egyptian Arabic sample.
[5]	2010	English- Hindi, Marathi and Kannada scripts	Serial and Parallel configurations, CLIR network.	Show- cased that a CLIR network incorporated with compositional transliteration model performs invariably on grade than that incorporated with a straight transliteration model.
[6]	2018	N/A	Neural networks	Convolutional Sequence to Sequence based Neural Machine Translation and Recurrent Neural Network that dispenses relatively adequate outcomes when it comes to multi lingual machine transliteration.

[7]	2009	English- Hindi language	SMT architecture, GIZA, beam search-based decoder	Offered model depicts that these approaches can be successfully utilized for the assignment of machine transliteration at an attained preciseness of 46.3%.
[8]	2013	Gurmukhi- English dialect	Statistical Machine Translation	The model is sampled on varied names and is sampled on further 1000 names and system has generated a preciseness of 97%.
[9]	2012	Hindi,Marathi-English scripts	Rule and Metric based hybrid model	The approach is direct without coaching any bilingual database and displays an in depth knowledge of word setup in Devanagari script.
[10]	2006	Shahmukhi- Gurmukhi dialect.	Glyph mappings, Reliance regulations	The developed model produces farther than 98% preciseness on traditional literature and 99% in case of modern literature.
[11]	2011	Arabic named entities	Bilingual lexica, transducers, NooJ	Their resources are reusable single-handedly as concluded by their trial and assessment validation of the model.
[12]	2003	English – Arabic scripts	Elected n- gram model	Assessed the statistically-trained model and a simpler hand- drafted model on a trial faction of named entities from the Arabic AFP corpus and substantiate that they pull off better than two online translation cradles.

3. Proposed Methodology

Data acquisition : The data is collected in the form of data sets to perform the experiment.

• Data Preprocessing : For applying the hybrid transliteration model to achieve preciseness in terms of graphemes and phonemes , preprocessing of the data is needed to reduce redundancy.

• Data Mapping : The fed data element is converted into the corresponding tokens respectively by the principal controller that consists of glyph mappings and diacritic dependency regulations.

• Data Output : The output submitted isn't bare a letter-to-letter transformation, but proper connotation of sounds correspondingly comes into effect. Kashmiri and Urdu and not major universally spoken languages. Therefore, their literature isn't that rich on online platforms in the configuration of NLP. Creating its transliteration meant to produce its database from scrape. employing Machine transliteration procedure of glyph mappings and diacritic dependency regulations , this application provides real time transliteration. This application has correspondingly bedded in it a third-party translator, which gives a coterminous restatement of each input term.

The application is a hybrid-based transliteration miniature where both phoneme and grapheme-based models are concerned. It focuses both on sounds and transformation of letters from root language to targeted language. applying the hybrid model gives big-time added leverage than operating only one of the two. The output submitted isn't bare a letter-to-letter transformation, but proper connotation of sounds correspondingly comes into effect.

3.1 ARCHITECTURE:

On server side, the model comprises of MySql and the language employed is PHP with Xampp as its development environment. On the client side, it has http + css technologies used.

All of the database is created using MySql (Xampp). The database consists of collection of all the letters of each language and their mappings with corresponding phonemes. PHP was used incorporation with JavaScript to create dynamic interactions with the databases.

Yii (yes, it is) framework of PHP is used.

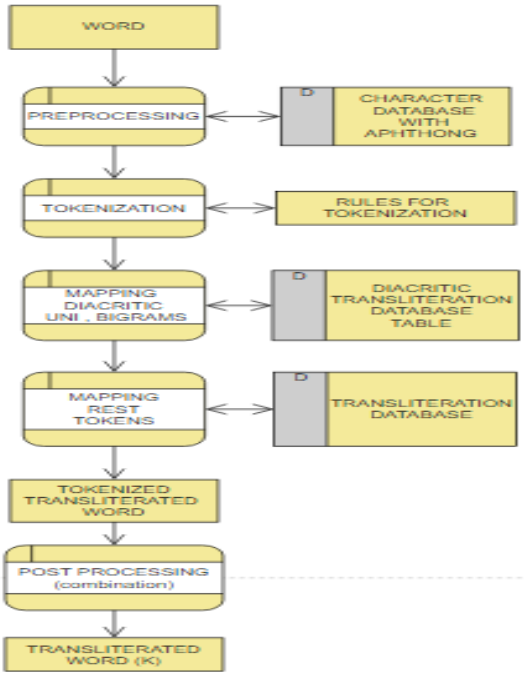

Fig 2: Architecture and Mapping

3.1.1 APACHE:

It's an open- source web server that intends to transfer web substances over the internet. It processes the queries and serves it via HTTP method. Apache is the most extensively utilized server with PHP. This server is embedded in XAMPP by default.

The data input by a client-side user in English language is broken down into alphabets for preprocessing. The resultant alphabets are then mapped to their respective tokens or ids which correspond to the residence of their transliterated alphabets.

The design also takes into account the phoneme of the terms so as to present a precise output. The resultant counter-plotted alphabets or terms are also blended to frame a comprehensive meaningful word in desired corresponding language to get the longed output.

3.1.2 XAMPP:

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It's an open- source bundle of web results that includes Apache dispensation for numerous servers and command- line executables along with modules similar as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and accounts via computers and laptops before releasing it to the main server. It's a platform that furnishes a capable atmosphere to sample and authenticate the working of blueprints based on Apache, Perl, MySQL database, and PHP through the network of the host itself.

3.1.3 Yii:

Yii is an acronym for “yes it is” and is and is an open-source OOP PHP framework. It is established around the composite MVC pattern and imparts pro and firm characteristics to projects. It is appropriate for framing all types of web applications.

3.1.4 PHP:

PHP is a recursive acronym for Hypertext Preprocessor. It is a scripting language, open source and extensively- applied in web development projects which can also be embedded into HTML.

3.1.5 Ajax:

Asynchronous JavaScript And XML is not a programming language but a blended technique of browser embedded XML Http request object and JavaScript and HTML DOM that allows asynchronous exchange of data behind the scenes with the server without reloading the whole webpage.

4. Implementation and Results

4.1 SYSTEM PROCESS

Unicode encoded English input text is collected by the input word parser which parses the input English text into English alphabets. These alphabets in the system are known as tokens. These tokens are then passed on to the transliteration element that converts it into the corresponding Hindi, Urdu and Kashmiri tokens respectively by the principal controller that consists of glyph mappings and diacritic dependency regulations. The output token generated is then passed back to the transliteration component. As soon as all the token conversions are completed, the respective tokens are combined to form the needed transliterated output text with diacritic implementation which is displayed by the output text generator.

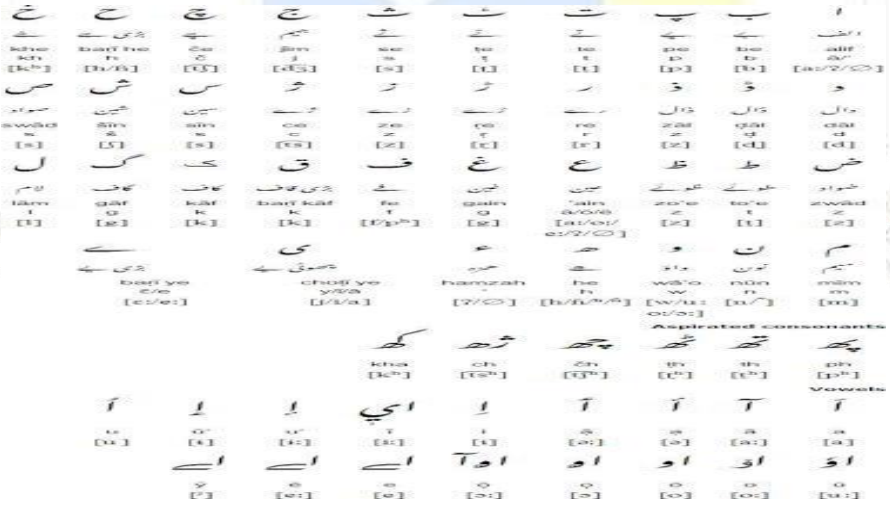

Fig 3: Word mapping of kashmiri glyphs with english language alphabets

The machine transliteration procedure is utilized in the token conversion and diacritic dependency regulating manager. While parsing the input tokens, their character dependencies are decided and then resolved in accordance with the contextual arrangement of the character in the token. There may or may not be character dependencies but in most cases the occurrence of character dependencies is high as there are diacritics in the Hindi, Urdu and Kashmiri scripts and to map these dependency embedded tokens in pure English script is very difficult. If the character in process carries a dependency, it is determined and fixed by dependency regulations.

If the character in process does not carry a character dependency, it is transliterated directly by tokenization and glyph mappings. In such manner all the English script tokens are transliterated into Hindi, Urdu and Kashmiri script tokens. A notable and necessary consideration in the process is that the scripts which contain the diacritical traces must be properly regulated and then mapped accordingly, failing of which will invalidate the actual pronunciation and meaning of the transliterated text.

The preciseness of this machine transliteration system depends upon the determined and properly regulated diacritical traces. Dearth of these important diacritical traces affects the preciseness to a greater extent.

4.2 RESULTS

Following the compilation of chosen input texts, which are then transliterated into Hindi, Urdu and Kashmiri texts by implementation the hybrid machine transliteration system, these output texts are tested for mistakes and inaccuracy. Testing of preciseness is done manually with aid from a dictionary and the persons knowing the respective scripts. The system was tested with a dataset of 1000 words and successfully achieved an accuracy of 90%. The only constraint to attain results this precise is the removal of ambiguity due to diacritical traces and resolving them through character dependency rules.

Fig 5: Interface to transliteration.

In fig 5, we can see that the user input is in English language given to be 'flip' which is transliterated to its corresponding transliterates in Urdu, Hindi and Kashmir as shown in the image above. The text box takes the input, and the corresponding three transliterations appear dynamically.

5. Conclusion and Future Directions

The axis of creating this application primarily is furnish a platform to transliterate English as input language to Urdu, Hindi and Kashmiri as desired output. Since the miniature is hybrid centred, the transliterations are considerably near to how they should be. Moreover, it created a route for attainability of data of Kashmiri and Urdu vocabulary which isn't verbalized by numerous and hence its data isn't handily set up. The PHP along with JavaScript has handed it with a certifiably user-friendly interface to apace with.

Some of the attainable advancements to it in hereafter are stated as:

• The improved model of the application will take in a voice to text transliteration.

• Scope of appending a camera scanner is also there, so that each word concentrated in it would be transliterated without possessing the needfulness to input it manually into the application text box.

• A calculator which measures the approximation of correctness of the transliterated terms is also a suggested characteristic.

REFERENCES

[1] "Understanding the Processes of Translation and Transliteration in Qualitative Research", 2010 by Krishna Regmi, Jennie Naidoo, Paul Pilkington.

[2] S. Karimi, F. Scholer, and A. Turpin, "Machine transliteration survey," *Computing Surveys (CSUR),*vol. 43, p. 17, 2011.

[3] Noura Farra, Nadi Tomeh, Alla Rozovskaya, and Nizar Habash. 2014. Generalized Character-Level Spelling Error Correction. In Proceedings of the Conference of the Association for Computational Linguistics (ACL), Baltimore, Maryland, USA.

[4] Ramy Eskander, Nizar Habash, Owen Rambow, and Nadi Tomeh. 2013. Processing Spontaneous Orthography. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Atlanta, GA.

[5] A KUMARAN, MITESH, M. KHAPRA and PUSHPAK BHATTACHARYYA, September 2010 Indian Institute of Technology Bombay.Compositional Machine Transliteration."work done during the author's internship at Microsoft Research India".

[6] Soumyadeep Kundu, Sayantan Paul, and Santanu Pal. 2018. A Deep Learning Based Approach to Transliteration. In *Proceedings of the Seventh Named Entities Workshop*, pages 79–83, Melbourne, Australia. Association for Computational Linguistics.

[7] Rama, Taraka & Gali, Karthik. (2009). Modeling Machine Transliteration as a Phrase Based Statistical Machine Translation Problem. 10.3115/1699705.1699737.

[8] Kumar, Pankaj. "Statistical Machine Translation Based Punjabi to English Transliteration System for Proper Nouns." (2013).

[9] Dhore, Manikrao & Dixit, Shantanu & Dhore, Ruchi. (2012). Hindi and Marathi to English NE Transliteration Tool using Phonology and Stress Analysis. 111-118.

[10] Malik, Muhammad Ghulam Abbas. (2006). Punjabi Machine Transliteration. 1. 10.3115/1220175.1220318.

[11] Fehri, Hela & Haddar, Kais & Ben Hamadou, Abdelmajid. (2011). Recognition and Translation of Arabic Named Entities with NooJ Using a New Representation Model. 134-142.

[12] Jaleel, Nasreen & Larkey, Leah. (2003). Statistical transliteration for english-arabic cross language information retrieval. 139-146. 10.1145/956863.956890.

[13] P. Bhattacharyya, M. M. Khapra, and A. Kunchukuttan, "Statistical Machine Translation between Related Languages," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, 2016, pp. 17–20, doi: 10.18653/v1/N16-4006.

[14] S. Kumar, S. Aggarwal, M. Sharma, and R. Mamidi, "How do different factors Impact the Inter-language Similarity? A Case Study on Indian languages."

[15] P. Agrawal and L. Jain, "English to Sanskrit Transliteration: an effective approach to design Natural Language Translation Tool Human Behaviour tracking system View project ARTICONF View project English to Sanskrit Transliteration: an effective approach to design Natural Language Translation Tool," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 1, [Online]. Available: https://www.researchgate.net/publication/334447849.

[16] J.-H. Oh, "A Comparison of Different Machine Transliteration Models," 2006. [Online]. Available: http://www.cs.cmu.edu/.

17] A. Diluni De Silva and A. R. Weerasinghe, "Masters Project Final Report (MCS) 2019 Project Title Singlish to Sinhala Converter using Machine Learning Student Name Supervisor's Name S E1 E2 For Office Use Only."

[18] J.-H. Oh and K.-S. Choi, "An English-Korean transliteration model using pronunciation and contextual rules," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1-7.

[19] Dhindsa, Baljeet. (2017). ENGLISH TO HINDI TRANSLITERATION SYSTEM USING COMBINATION-BASED APPROACH. International Journal of Advanced Research in Computer Science. 8. 609-613. 10.26483/ijarcs.v8i8.4801.