A Rule-Based Kurdish Text Transliteration System

SINA AHMADI, Paris Descartes University

In this article, we present a rule-based approach for transliterating two of the most used orthographies in Sorani Kurdish. Our work consists of detecting a character in a word by removing the possible ambiguities and mapping it into the target orthography. We describe different challenges in Kurdish text mining and propose novel ideas concerning the transliteration task for Sorani Kurdish. Our transliteration system, named *Wergor*, achieves 82.79% overall precision and more than 99% in detecting the double-usage characters. We also present a manually transliterated corpus for Kurdish.

CCS Concepts: • **Computing methodologies** → *Natural language processing*; *Information extraction*; *Language resources*;

Additional Key Words and Phrases: Transliteration, rule-based approach, Kurdish, less-resourced language processing

ACM Reference format:
Sina Ahmadi. 2019. A Rule-Based Kurdish Text Transliteration System. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 18, 2, Article 18 (January 2019), 8 pages.
https://doi.org/10.1145/3278623

INTRODUCTION

Kurdish is an Indo-European language with a majority of speakers in the Kurdish regions of Iran, Iraq, Turkey, and Syria. Although it is spoken by 20 to 30 million people [1, 2], the Kurdish language is considered as a less-resourced language. In 2016, Google added 13 new languages to its online automated translation tool, Google Translate, among them Kurdish (for the time being, only Kurmanji dialect). One of the main reasons of this delay, in comparison to some other languages with less users for whom the same service was provided earlier, is the lack of parallel corpora, online resources, and language processing tools [3].

Regarding the area and the extent to which Kurdish orthographies are applied, one should confess that integrity in writing Kurdish has not been achieved. The difference of orthographies naturally results in the distinction of produced textual sources and adds to the gap between the dialects and, thus, scatters readers. Despite the fact that the Kurdish Academy of Language introduced the Unified Kurdish Alphabet *Yekgirtú* in response to this problem [4], no standard orthography is popularly accepted considering all the challenges and the diversity of the dialects. Aware of this problem, Kurdish intellectuals have emphasized on the unification of the orthographies [5].

In this article, we are focusing on the challenges of transliteration of the two most used orthographies, Arabic-based and Latin-based, for Sorani Kurdish. Transliteration is a mapping from one

Authors' addresses: S. Ahmadi, U. F. R. Mathématiques et Informatique, Université Paris Descartes, 45, rue des Saints-Pères, 75006 Paris; email: ahmadi.sina@outlook.com.

2375-4699/2019/01-ART18 $15.00
https://doi.org/10.1145/3278623

system of writing into another, typically grapheme to grapheme [6]. Given $w_{input} = c_1, c_2, \ldots, c_n$ in the orthography A, a transliteration task consists of mapping each character of the word to an equivalent character in the orthography B, which yields $w_{output} = c_1, c_2, \ldots, c_m$. This juxtaposition is not always straightforward. In the case of Sorani Kurdish, the Latin-based and Arabic-based orthographies are not completely identical in terms of character representation. Although confronting the problem of normalization in Kurdish seems to be addressed already in some of the previous research, such as in Refs [7, 8, 9], as a partial task, a solution has not been proposed for the transliteration task so far. For instance, in a recent work by Hassani [10], transliteration has been mentioned implicitly as one of the tasks, but no detail has been reported concretely.

The task of transliteration is one of the fundamental elements in many natural language processing (NLP) applications such as statistical machine translation, terminology extraction, cross-lingual data linking, and so forth. Transliteration can be done with phoneme-based or grapheme-based models for which the latter has been shown to perform better than the first one [11]. Kashani et al. [12] and Al-Onaizan and Knight [11] use a grapheme-based model, and Stalls and Knight [13] and Pervouchine et al. [14] use the phoneme-based approach. Since there are a few languages with manually labeled transliteration pairs (a word and its transliteration), some studies such as those in Refs [15–17] have been focused on transliteration mining, which consists of automatically extracting transliteration pairs from a noisy list of transliteration candidates.

The rest of the article is organized as follows: First, we provide a description about Kurdish writing systems in Section 1. In Section 2, we focus on the challenges of Sorani Kurdish transliteration in the Arabic-based (also referred to as "Persian-Arabic") and Latin-based orthographies. In Section 3, we present the rule-based techniques used in Wergor.[1] This section includes our rule-based methods to solve the present challenges. Section 4 is devoted to the tests and experiments on the algorithms. In this section, we describe our manually transliterated dataset. Finally, in Section 5, our work is concluded and some ideas are proposed for future works.

1 KURDISH WRITING SYSTEMS

Nowadays, Kurdish is written in several orthographies adopted from other languages and, thus, applied to it [18]. Although debate on what orthography to apply yet remains, Latin-based orthography (henceforth referred to as *LbO*) and Arabic-based orthography (henceforth referred to as *AbO*) are among the most popular ones that are, respectively, mostly used for the *Kurmanji* dialect and the *Sorani* dialect of Kurdish. In addition to these two main dialects, *Hawrami* and *Kalhor* are also written in the AbO. These orthographies are based on the phonetics of the language [19].

In order to provide a common description about Kurdish orthographies and to avoid inconsistent descriptions, mainly in Refs [20–23], we have used the description in Ref. [24] for the LbO and the presented characters in Ref. [25] for the AbO. Although some of the characters may have other usages in other descriptions, these two references are the most well-known for Kurdish writers. Table 1 shows the characters in these orthographies in comparison to one another. In case a character does not exist for a given phoneme, the case is colored in gray. We encourage future researchers to use the selected Latin-based orthography as it does not have any ambiguity.

In the early stages of development of text processing tools for Kurdish, some fonts have been introduced to Kurdish users. *Dilan fonts*, *Ali fonts*, *Zanest fonts*, and *Rebaz fonts* were among the most well-known fonts. These fonts were mainly based on the Persian and the Arabic keyboards and did not support Unicode. Fortunately, the existing characters in the Kurdish orthographies are

[1] "Wergor", pronounced as "wargor", is composed of "wer"—a Kurdish prefix related to *transformation*, and "gor"—the stem of "goran" meaning *to change*. We coined this word for "transliterater" similar to the Kurdish word "wergêr", meaning *translator*.

Table 1. Comparison of the Latin-based and the Arabic-based Orthographies

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Arabic-based	Initial	ئا	بـ	جـ	چـ	دـ	ئه	ئێ	فـ	گـ	هـ	حـ		ێ	ز	کـ	لـ	ڵ	صـ	نـ	ۆ	پـ	قـ	رـ	ڕ	سـ	شـ	تـ	ئو	ئوو	ڤـ	وـ	خـ	یـ	ژ	عـ	غـ	ڵ
	Middle	ا	بـ	جـ	چـ	دـ	ه	ێ	فـ	گـ	هـ	حـ		ێ	ز	کـ	لـ	ڵ	صـ	نـ	ۆ	پـ	قـ	رـ	ڕ	سـ	شـ	تـ	و	وو	ڤـ	وـ	خـ	یـ	ژ	عـ	غـ	ڵ
	End	ا	ب	ج	چ	د	ه	ێ	ف	گ		ح		ێ	ز	ک	ل	ڵ	م	ن	ۆ	پ	ق	ر	ڕ	س	ش	ت	و	وو	ڤ	و	خ	ی	ژ	ع	غ	
	Single	ا	ب	ج	چ	د	ه	ێ	ف	گ	ه	ح		ێ	ز	ک	ل	ڵ	م	ن	ۆ	پ	ق	ر	ڕ	س	ش	ت	و	وو	ڤ	و	خ	ی	ژ	ع	غ	
Latin-based		A a	B b	C c	Ç ç	D d	E e	Ê ê	F f	G g	H h	Ḥ ḥ	I ı	İ i	J j	K k	L l	Ł ł	M m	N n	O o	P p	Q q	R r	Ř ř	S s	Ş ş	T t	U u	Û û	V v	W w	X x	Y y	Z z	Ë ë	Ẍ ẍ	

completely supported by the Unicode standard. In the most recent development, the *Kurditgroup* keyboard is proposed based on the Unicode characters, which is widely used by most of Kurdish users.[2] We have also used this keyboard in our study.

2 KURDISH TEXT NORMALIZATION CHALLENGES

For the current Arabic-based and Latin-based orthographies, we can classify the normalization challenges in three categories:

2.1 Characters Used to Represent More Than One Phoneme

This is the case of "و" and "ی" in the AbO, which may be transliterated, respectively, as {"w" or "u"} and {"y" or "î"} in the LbO. For instance, the word "هاوین" could have four possible transliterations considering different mappings "یـ" → {"y", "î"} and "و" → {"w", "u"}: "hauîn", "hauyn", "hawîn", "hawyn", for which "hawîn" is the correct form. Despite the visual similarity of "ه" as the equivalent of "h" and "e" in LbO, this character is not in the same category with "و" and "ی" having different codes in Unicode.

2.2 Characters with no Equivalent in the Other Orthography

This is the case of "ح", "ع", "غ", "ڵ", and "ڕ" characters in the AbO for which there is no equivalent in the LbO. A specific case, however, is the case of *Bizroke*. Bizroke (which literally means "the little furtive") is represented by "i" in the LbO while it is totally ignored in the AbO. For example, the word "ئاگر" may be transliterated as "agr", which is not correct since the Bizroke between "g" and "r" can not be represented in the AbO. The correct form is "agir". Having said that, native speakers pronounce Bizroke while speaking, even if it does not exist in the Arabic-based orthography [26].

2.3 Unicode Assignments of the Arabic-Based Kurdish Alphabet

The potential sources of ambiguity in the assignment of the characters of the current Kurditgroup keyboard is as follows:

- —Some of the Arabic characters have similarities in form, but they have different Unicodes, e.g., "ي" (U+064A) instead of "ی" (U+06CC) for {"î", "y"} and "ك" (U+0643) instead of "ک" (U+06A9) for "k" in the LbO.
- —Although "ه"(U+0647) as "h" is a connecting character when placed at the end of a word, it seems visually identical to "ە"(U+06d5) that represents "e". For instance, the final "ه" in "بەهبەه" ("behbeh") is not connected to the previous character, which shows that the final "ه" is "h". This is not a source of ambiguity in terms of normalization since the two possible forms of "ه" have different Unicodes. Some suggest that "ه" as "h" be marked using a zero-width non-joiner character (U+200C) or an *en dash* (U+2013). Such words ending with the "h" phoneme are quite rare in Sorani Kurdish.

[2] https://kurditgroup.org/downloads.

Table 2. Examples of Different Challenging Categories in Sorani Kurdish Text Normalization

Word	Possible transliterations	Correct form	Challenge category
بیور	bîwr bywr bîur byur	bîwir	"ر" → {"w", "u"} "ی" → {"y", "î"} *Bizroke*, i.e., "i", not recognizable
هەپەسان	hepesan	ẖepesan	No character for "ح" in the LbO
بەنا**وو**دەنگ	benaûdeng	benawûdeng	Double character for one character

Challenging characters, if available, are made bold.

—Although "û" in the LbO is a single character with a unique Unicode (U+00FB), the equivalent character "وو" in the AbO is created by a double "و". The usage of two characters to represent another character is far problematic than a simple replacement since some of the words are preceded or succeeded by a similar character. For instance, the double "و" in words like "هاوولّاتی" and "وتوویژ" may be transliterated, respectively, as "haûłatî" instead of its correct form "hawwiłatî" and "witûej" instead of its correct form "witûwêj". In a similar way, some have proposed using "ll" and "rr" to represent "ڵ" and "ڕ" in the LbO [27]. Consequently, it would be the same case for such usages.

Table 2 shows some words in the AbO with the possible transliterated forms in LbO, the correct form for each word based on the reference orthography, and the challenge category. Note that the possible transliterations are not essentially correct since they represent the possible mapping of the characters of one orthography to the other.

3 WERGOR SYSTEM

Figure 1 illustrates Wergor transliteration system architecture. The system normalizes a given text by preprocessing and unifying different forms of a character discussed in Section 2.3. In this stage, Wergor yields the corresponding characters of the double-usage characters such as "و" and "ی" and detects the possible presence of Bizroke in the AbO. Finally, the characters are mapped to the other orthography characters. According to this architecture, the system transliterates "بزگوڕ" from AbO into "bizguř" in the LbO by detecting the correct equivalent of "و" as "u" and the correct position of Bizroke.

Our method to solve the aforementioned challenges in Sorani Kurdish text processing follows the rules based on the phonological characteristics and the writing tradition. Some of the essential rules based on Ref. [22] that are applied in Wergor are as follow:

—If a word begins with a vowel, i.e., {"ا", "ێ", "ی", "ۆ", "وو", "و", "ە"}, it is always preceded by "ئ" in the AbO. This is the only usage of "ئ" (called *Hamza*) as an auxiliary character and is only used in the AbO.
—Although "r" as the first phoneme in every word in the Sorani Kurdish is trilled, thus, pronounced "ř", traditionally the non-trilled form "r" is used [22]. This rule is applied in the two orthographies. For instance, "ڕۆژ", "ڕاوێژ", and "ڕێگا" are to be transliterated as "roj", "rawêj", and "rêga", respectively.
—No Sorani Kurdish word begins with "ڵ / ɬ" [22].
—Since, in Sorani Kurdish, a word has as many syllables as it has vowels, no two vowels can be in one syllable. Some of the frequent syllable structures in Sorani Kurdish are: V, VC, VCC, CV, CVC, CVCC, where V stands for vowel and C stands for consonant. In no syllable structure is a vowel preceded or succeeded by another vowel [26].

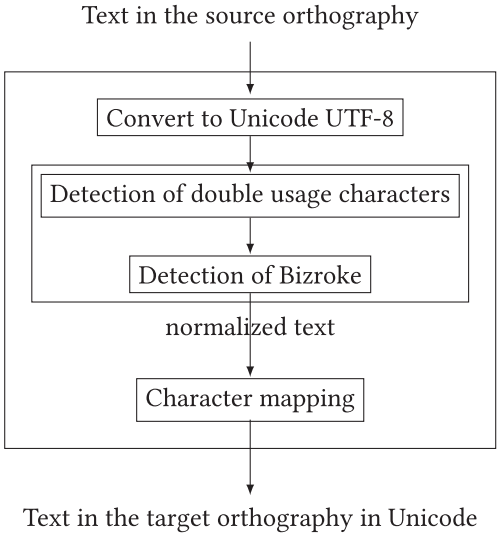

Fig. 1. Wergor system architecture.

Using syllable structure patterns in Kurdish, we propose Algorithm 1 to detect double-usage characters "و" and "ی". A character in its single form is considered consonant by default. The algorithm follows the same procedure for any of the target characters.

Although the transliteration of Bizroke (i.e., "i") from the LbO to the AbO is by omitting it, it is challenging to find Bizroke in the inverse direction. Analyzing syllable structures, the only rule that we could rely on is that in the CVC structure, if positioned as the first syllable, V is always Bizroke, e.g., "bira", "wirya", except the cases that the second consonant is "y" or "w", e.g., "kwêr", "dyar". Although it seems to be frequent to see Bizroke in the same pattern in the last syllables, e.g., "çirij", "kirdin", we could not use it as a rule.

4 EXPERIMENTS

4.1 Dataset

Among the 36 top ranked Kurdish websites, including news and media services, we have found only one site that uses AbO for both Sorani and Kurmanji dialects.[3] Eighteen websites use only LbO for Kurmanji and 29 websites use only AbO for Sorani. We found no Sorani website that uses LbO.

In order to provide a resource for Kurdish transliteration, we propose Wergor corpus, to the best of our knowledge, as the first transliteration corpus for Kurdish. Our corpus consists of parallel transliterated texts from the two orthographies. This corpus can be used for other tasks in machine translation as well.

4.2 Results and Discussion

Table 3 shows the results of Wergor in transliterating our dataset from the AbO to the LbO. Results of different tests are presented based on the correct and incorrect transliterations and the precision of the system is calculated as the the percentage of the correct transliterations.

[3]Ranking based on Alexa, retrieved from http://www.alexa.com.

Table 3. Arabic to Latin Transliteration Results

		Bizroke detection		w/u detection	y/î detection	whole test set
Prediction	Correct	721/1,861		2,472/2,480	4,808/4,850	5,779/6,980
	Incorrect	last syllable	other syllables	8/2,480	48/4,850	1,201/6,980
		286/1,140	854/1,140			
Precision		38.74%		99.67%	99.13%	**82.79%**

ALGORITHM 1: Detection of "w/u" and "y/î" equivalents in the Arabic-based orthography

Input: Word W containing the target char ("و", "ی")
Output: Detected forms of "و" as "w" or "u" and "ی" as "y" or "î" in W.

```
1:  procedure TargetCharacterDetector(W, TargetChar)
2:      length ← length of W
3:      vowels ← ["i", "î", "u", "û", "ه", "ا", "و", "ئ"]
4:      Hamza ← "ئ"
5:      target_char_vowel ← the vowel form of TargetChar
6:      target_char_consonant ← the consonant form of TargetChar
7:      if W = TargetChar then
8:          return target_char_consonant
9:      for index ← 0 to length do
10:         if W[index] = Hamza & W[index + 1] = TargetChar then
11:             W[index + 1] ← target_char_vowel
12:             index ← index + 1
13:         else
14:             if W[index] = TargetChar then
15:                 if index = 0 then
16:                     W[index] ← target_char_consonant
17:                 else
18:                     if W[index − 1] is in vowels then
19:                         W[index] ← target_char_consonant
20:                     else
21:                         if index + 1 < length then
22:                             if W[index + 1] is in vowels then
23:                                 W[index] ← target_char_consonant
24:                             else
25:                                 W[index] ← target_char_vowel
26:                         else
27:                             W[index] ← target_char_vowel
28:     Remove Hamza in W
29:     return W
```

In detecting the possible position of Bizroke, Wergor achieves 38.74% precision and 100% recall. Since the rule that we could apply in the current version of the system for detecting Bizroke only considers the first syllables, Wergor is not able to correctly find the position of Bizroke in the 1,140 cases among 1,861. In other words, the correct prediction refers to those words that have only one Bizroke, and it is positioned in the first syllable. In the incorrect transliterations, in 286 cases, Bizroke is in the last syllable, and in 854 cases, it is in other syllables.

Evaluating the system on the double-usage characters, i.e., "و" and "ی", shows a high precision of more than 99% and a recall of 100% since all relevant words were retrieved. Incorrectly transliterated words are mostly non-Kurdish words, e.g., "Claud" that are used in the original form in the manually transliterated dataset, and proper nouns such as "Kurdistan," which are capitalized in the LbO. The AbO does not have capital letters.

On the other hand, the Wergor system achieves almost 100% precision in transliterating the LbO into the AbO. Since the mapping of the LbO characters into the AbO ones is straightforward with no challenging characters, this precision is justifiable.

Figures A.1 and A.2 in Appendix A show two transliteration texts using Wergor.

5 CONCLUSIONS AND FUTURE WORK

In this article, we propose a rule-based technique for Kurdish text transliteration. Kurdish confronts various challenges in transliterating its two popular orthographies, Arabic-based and Latin-based. In this article, we described a method to solve these challenges using the Wergor transliteration system. Although our system achieves 99% precision in transliterating double-usage characters ("و", "ی"), it is less efficient in transliterating Bizroke, i.e., "i." In order to improve the current results, a bigger transliteration dataset is required. We also believe that the phonological aspects of the language can be of help, which are not enough studied yet. Having the Wergor transliteration dataset, we are currently interested in applying statistical methods for detecting Bizroke more efficiently.

Our codes and corpus are available at https://github.com/sinaahmadi/wergor.

A APPENDIX

بە دەگمەن لە رۆژاوای کوردستاندا وەرگێڕاو لە چاپ دراوە. نووسەری ژن، کەچا کورد، زۆرترین ژمارەی وەرگێڕاوی هەیە. هەروەها چوونکینێ هیچ لەچاپدرێک خۆی بۆ وەرگێڕاوەکان شڵۆێ ناکا، زۆربەی وەها بەرهەمانێک لە سەر هەست و وێستی نووسەرەوە بەخۆڕایی وەردەگێڕدرێنەوە.
Be degmen le rojaway **kurdst**anda wergêřaw le çap dirawe. Nûserî jin, **k**eça **k**urd, zo**rtr**în jimarî wergêřawî heye. Herweha çûnkînê hîç leçapderêk xoy bo wergêřawekan şiłwê naka, zorbey weha berhemanêk le ser hest û wîstî nûserewe bexořayî werdegêř**dr**ênewe .
Be degmen le rojaway Kurdistanda wergêřaw le çap dirawe. Nûserî jin, Keça Kurd, zortirîn jimarî wergêřawî heye. Herweha çûnkînê hîç leçapderêk xoy bo wergêřawekan şiłwê naka, zorbey weha berhemanêk le ser hest û wîstî nûserewe bexořayî werdegêřdirênewe.

Fig. A.1. Transliteration of an example text, in the first row, from the AbO to output text in the second row in the LbO. The manually transliterated text is shown in the last row. The errors are shown in bold. Both texts are in the Sorani Kurdish language.

Sa ew kesane ke eşkence kirawin, nek tenya be hoy kurdî nûsînewe tawanbar bûwin, xwêndinewe, biław kirdinewe û ragirtinî çapemenîy kurdî le nêw małekanda û tenanet "çak zanînî kurdî"ş sûçêkî gewreyan bûwe.
سا ئەو کەسانە کە ئەشکەنجە کراون، نەک تەنیا بە هۆی کوردی نووسینەوە تاوانبار بووون، خوێندنەوە، بڵاو کردنەوە و راگرتنی چاپەمەنیی کوردی لە نێو ماڵەکاندا و تەنانەت "چاک زانینی کوردی"ش سووچێکی گەورەیان بووە.
سا ئەو کەسانە کە ئەشکەنجە کراون، نەک تەنیا بە هۆی کوردی نووسینەوە تاوانبار بووون، خوێندنەوە، بڵاو کردنەوە و راگرتنی چاپەمەنیی کوردی لە نێو ماڵەکاندا و تەنانەت "چاک زانینی کوردی"ش سووچێکی گەورەیان بووە.

Fig. A.2. Transliteration of an example text, in the first row, from the LbO to the output text in the second row in the AbO. The manually transliterated text is in the third row. No errors found.

REFERENCES

[1] Stefan Sperl and Philip G. Kreyenbroek. 2005. *The Kurdish Question: A Historical Review*. Routledge. 17–34.

[2] Hossein Hassani and Dzejla Medjedovic. 2016. Automatic Kurdish dialects identification. *Computer Science & Information Technology* 6.2 (2016), 61–78.

[3] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication* 56 (2014), 85–100. DOI: https://doi.org/10.1016/j.specom.2013.07.008

[4] Kurdish Academy of Language. [n.d.]. Kurdish Unified Alphabet. Retrieved May 24, 2018 from http://www.kurdishacademy.org/?q=node/2.

[5] Amir Hassanpour. 1992. *Nationalism and Language in Kurdistan*, 1918-1985. Edwin Mellen Pr.

[6] Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Comput. Linguist.* 24, 4 (Dec. 1998), 599–612. http://dl.acm.org/citation.cfm?id=972764.972767

[7] Kyumars Sheykh Esmaili. 2012. Challenges in Kurdish text processing. *arXiv preprint arXiv:1212.0074* (2012).

[8] Kyumars Sheykh Esmaili, Shahin Salavati, and Anwitaman Datta. 2014. Towards Kurdish information retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)* 13, 2 (2014), 7.

[9] Purya Aliabadi, Sina Ahmadi, Shahin Salavati, and Kyumars Sheykh Esmaili. 2014. Towards building kurdnet, the Kurdish wordnet. In *Proceedings of the 7th Global WordNet Conference (GWC'14)*. 1–6.

[10] Hossein Hassani. 2017. Kurdish interdialect machine translation. *VarDial 2017* (2017), 63.

[11] Yaser Al-Onaizan and Kevin Knight. 2002. Machine transliteration of names in Arabic text. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*. Association for Computational Linguistics, 1–13.

[12] Mehdi M. Kashani, Fred Popowich, and Anoop Sarkar. 2007. Automatic transliteration of proper nouns from Arabic to English. In *Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages*. 275–282.

[13] Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in Arabic text. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*. Association for Computational Linguistics, 34–41.

[14] Vladimir Pervouchine, Haizhou Li, and Bo Lin. 2009. Transliteration alignment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*. Association for Computational Linguistics, 136–144.

[15] Hassan Sajjad, Helmut Schmid, Alexander Fraser, and Hinrich Schütze. 2017. Statistical models for unsupervised, semi-supervised, and supervised transliteration mining. *Computational Linguistics* 43, 2 (2017), 349–375.

[16] Hassan Sajjad, Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. Comparing two techniques for learning transliteration models using a parallel corpus. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. 129–137.

[17] Sara Noeman and Amgad Madkour. 2010. Language independent transliteration mining system using finite state automata framework. In *Proceedings of the 2010 Named Entities Workshop*. Association for Computational Linguistics, 57–61.

[18] Kurdish Academy of Language. [n.d.]. Orthography, standardization and unification. Retrieved May 24, 2018 from http://www.kurdishacademy.org/?q=node/499.

[19] Kurdish Academy of Language. [n.d.]. Kurdish Orthography, a historical view. Retrieved May 24, 2018 from http://www.kurdishacademy.org/?q=node/116.

[20] Taufiq Wahby. 1929. *Desturî zimanî kurdî (Grammar of Kurdish Language)*. Al-Haditha Publishers, Baghdad.

[21] Abdurrahman Sharafkandi. 1991. *Henbane Borîne (Kurdish-Kurdish-Persian Dictionary)*. Soroush Pub.

[22] W. M. Thackston. 2006. Sorani Kurdish: A reference grammar with selected readings. *Harvard University. Department of Near Eastern Languages & Civilizations* (2006).

[23] W. M. Thackston. 2006. *Kurmanji Kurdish: A Reference Grammar with Selected Readings. Harvard University. Department of Near Eastern Languages & Civilizations*.

[24] Roger Lescot Emir Djeladet Bedir Khan. 1970. *Grammaire kurde: dialecte kurmandji*. Librairie d'Amérique et d'Orient.

[25] Joyce Blau. 2000. *Manuel de kurde: Sorani*. L'Harmattan.

[26] Ernest N. McCarus. 1958. A Kurdish grammar: Descriptive analysis of the Kurdish of Sulaimaniya, Iraq. *American Council of Learned Societies Program in Oriental Languages* Publications Series B-Aids, 10 (1958).

[27] Joyce Blau and Veysi Barak. 1999. *Manuel de kurde: kurmanji*. Editions L'Harmattan.

Received August 2017; revised June 2018; accepted September 2018