Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 142 (2018) 238–245

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Arabic Computational Linguistics (ACLing 2018),
November 17-19 2018, Dubai, United Arab Emirates

A Sequence-to-Sequence based Approach For the double Transliteration of Tunisian Dialect

Jihene Younes[a], Emna Souissi[b], Hadhemi Achour[a], Ahmed Ferchichi[a]1

[a]*Université de Tunis, ISGT, LR99ES04 BESTMOD, 2000, Le Bardo, Tunisia*
[b]*Université de Tunis, ENSIT, 1008, Montfleury, Tunisia.*

Abstract

Transliteration consists of automatically transforming a grapheme's transcription from one writing system to another, while preserving its pronunciation. It is usually used in the context of machine translation and cross language information retrieval, mainly to deal with the issue of named entities and technical terms. In the case of some Arabic dialects, which are used on the social web in both Latin and Arabic scripts and which are still low-resource languages, transliteration is of great benefit for the automatic generation of various linguistic resources (parallel corpora and lexica), useful for their automatic processing. In this work, we focus on the Tunisian dialect transliteration. We propose a deep learning based Sequence-to-Sequence approach to perform a word-level transliteration of the user generated Tunisian dialect on the social web, in both Latin to Arabic and Arabic to Latin senses.

Keywords: Tunsian dialect; transliteration; Latin transcription; Arabic transcription; Sequence-to-Sequence; deep learning, natural language processing.

* Corresponding author. *E-mail address:* jihene.younes@gmail.com

10.1016/j.procs.2018.10.481

1. Introduction

Transliteration is the operation of substituting a grapheme of a writing system with a corresponding grapheme of another system, while preserving its pronunciation. The need of the transliteration task has mainly emerged to facilitate machine translation. Being a crucial task for many multilingual applications, and a technology in great demand in recent years, machine translation has faced several issues when dealing with textual entities that require conserving the original pronunciation [1]. These issues include technical terms and personal names and places. Indeed, the coexistence of multiple variants in the spelling of people's names and places raises real problems for information retrieval, document processing and data interoperability [1]. The transliteration task has become consequently, of great benefit to solve such problems, as it preserves the phonetic value of the textual content.

Transliteration is not only helpful in machine translation, but it also allows searching and indexing content, and helps finding information written in a different alphabet from that of the query [2].

In recent years, interest in this task has grown significantly, due to the multilingual nature of the web and the increasing need for transliteration in information retrieval. This is especially true for the dialectal varieties of the Arabic language, as their use on the web has increased in both the Latin and the Arabic alphabet, especially in social media.

Despite their growing use in the web, Arabic dialects are still poorly endowed languages in terms of resources allowing their automatic processing. The task of transliteration can therefore help overcoming this problem in creating parallel corpora and dictionaries allowing the study and processing of these dialects.

In this paper, we focus on the Tunisian dialect (TD) in particular, since it is widely produced on the web in both Latin and Arabic scripts. Its transliteration can prove to be an extremely useful operation, especially for the generation of various TD language resources (corpora and dictionaries) and can be particularly relevant for allowing the search for information in both scripts.

It also should be noted that, when performed from Latin to Arabic, TD transliteration can help reuse and adapt NLP tools, originally developed for the Modern Standard Arabic (MSA), for the TD automatic processing. It can, in addition, ease writing dialectal content when users are unfamiliar with Arabic keyboards or when such keyboards are not available. The Latin script can also be particularly difficult to read for some users who are not accustomed to this kind of writing, since it is characterized by the use of digits, acronyms, abbreviations, and essentially multilingualism. From Arabic to Latin, TD transliteration is mainly essential for creating TD parallel corpora and dictionaries, to better deal with the translation task and to generate language resources that help identifying dialectal content in code-switched textual data. Therefore, we propose in this work, to adopt a deep learning approach based on Sequence-to-Sequence (Seq2Seq) models to perform the transliteration of the TD in both Arabic-to-Latin and Latin-to-Arabic senses.

The remainder of this paper will be organized as follows: after a brief review of the work on transliteration in Section 2, we present in Section 3 the Seq2Seq methods. Section 4 will be devoted to the presentation of our transliteration approach for the Tunisian dialect in its Arabic and Latin forms. We expose and discuss the results of our experiments in Section 5.

2. Related work

Several works have been carried out on the transliteration task, resorting to different kinds of methods and techniques. Rule-based approaches have been adopted in many works especially those concerning the Arabic script, as it can be written using the Arabic or the Latin alphabet. The Latin transcription of Arabic took various names such as Romanized Arabic, Arabizi, Franco Arabic, Arabish, etc. [3]. Rule-based methods were used by Buckwalter [4] who developed an Arabic to Latin transliteration system, when online communication was restricted to ASCII only environments [3]. The Buckwalter system was upgraded by Habash et al. [5] who aimed to facilitate the pronunciation of the transliterated words by adding non- ASCII characters. Souissi and Debili [6] focused on the transliteration of Arabic proper names to the Latin transcription and vice versa. Their work was based on a computer-aided definition of a set of contextual rules. To convert Romanized Persian to the Arabic writing, Maleki and Ahrenberg [7] performed a syllabication on input strings and implemented context rules as well. Regarding other languages, Sen and Garg [8] focused on the Bengali to English transliteration following a set of rules. Wan and Verspoor [9] generated Chinese characters corresponding to person names and places written in English.

Several works have been carried out on the transliteration task using language modeling techniques. This was the case of Chalabi and Gerges [3] who built a Romanized Arabic transliteration engine that creates all possible Arabic transliterations by a generator model composed of mapping rules, and their probabilities. Darwish [10] tackled the Arabizi identification task and converted the identified content (dialect or MSA) to the Arabic transcription using language modeling. Marlies et al. [11] performed character mappings of Arabizi to Arabic sequences, to generate transliteration candidates for a given Arabizi word using an Arabic character language model. Other works focused on the Bengali-English named entities transliteration [12] and the bilingual transliteration of Japanese-English technical terms in the scientific domain.

To deal with the Arabic writing system, some researchers resorted to the probabilistic finite state automata, such as Al-Onaizan and Knight [13] who focused on the transliteration of Arabic names into English, and Saadane and Semmar [1] who developed an automatic transliteration system for the Arabic names into Latin characters as well.

A conventional orthography for dialectal Arabic, named CODA proposed by Habash et al. [14] was adopted for the transliteration problem by Bies et al. [15] to build a parallel corpus for the Arabizi-Arabic writing, by Al-Badrashiny et al. [16] to develop a system that generates the CODA potential transliterations of a given Arabizi word, and Eskander et al. [17] to transliterate Arabizi written on social media into Arabic. Zribi's team [18] developed a CODA for the TD, which was used by Masmoudi et al. [19] to convert Tunisian Arabizi into Arabic.

Machine learning has become the main resort to deal with NLP tasks in the past few years. Several researchers used machine learning models for the transliteration problem, some of them focused mainly on the English language such as Rathod et al. [20] who proposed named entity transliteration for Hindi to English and from Marathi to English, Kang and Choi [21] developed a bi-directional methodology for English-Korean transliteration in both senses, Ganesh et al. [22] focused on the transliteration of Hindi-English, Reddy and Waxmonsky [23] performed a substring-based transliteration of English-Indic languages. Nabende [24] proposed a system for English-Russian based on a pair HMM training, Wang and Tsai [25] tackled the transliteration of English-Korean named entities, and Dhore et al. [26] worked on the transliteration of Hindi-English named entities.

Regarding works including the Arabic script, El-Kahki et al. [27, 28] focused on the transliteration task using a generative graph reinforcement model including the English-Arabic languages. Guellil et al. [29] developed a character-level transliteration system for the Algerian Arabizi into MSA using a recurrent neural network. Sajjad et al. [30] focused on the transliteration of languages including English-Arabic using a generative model with supervised, semi-supervised and unsupervised mining. Deselaers et al. [31] used deep belief network to deal with the transliteration of Arabic named entities. Ammar et al. [32] resorted to CRF to transliterate Arabic named entities into English. Ameur et al. [33] focused on the Arabic to English named entity transliteration as well, using sequence-to-sequence models. The same model was used by Rosca and Breuel [34] who tackled the English to IPA, English to Japanese and Arabic to English named entities transliteration

As regards the Tunisian dialect, only one work was performed, to the best of our knowledge, using a machine learning model, namely HMM [35] to transliterate Tunisian Arabizi into Arabic. The problem was viewed as a sequence labeling task, consisting in assigning to each Latin character its corresponding transliteration in Arabic.

Based on the reviewed works, we can clearly notice that the transliteration task regarding the Tunisian dialect was only tackled in two works, namely in [19] using an orthographic convention, and in [35] using a generative machine learning model (HMM), viewed as a sequence labeling task. In this work, we adopt a new approach based on a sequence-to-sequence deep learning model to perform the TD transliteration from Arabic to Latin and vice versa.

3. Sequence-to-Sequence methods

Seq2Seq models were originally created for machine translation [36]. They have recently witnessed a great success, thanks to their ability to use deep neural networks [37] and have become the norm in most commercial translation systems, like Google Translate.

The two main constituents of the Seq2Seq models are the encoder and the decoder, representing two distinct recurrent neural networks [38]. The encoder browses an input string, one element at a time and transforms the sequence into an output for the decoder. The decoder then decompresses the encoder's output creating the model's output string.

Indeed, the encoder's task is to read the source sequence first to build its dimensional representation as a vector. This representation is a sequence of numbers representing the meaning of the source sequence, which is subsequently processed by the decoder to emit an output sequence. Fig. 1 illustrates the principle of the Seq2Seq approach for the translation task.

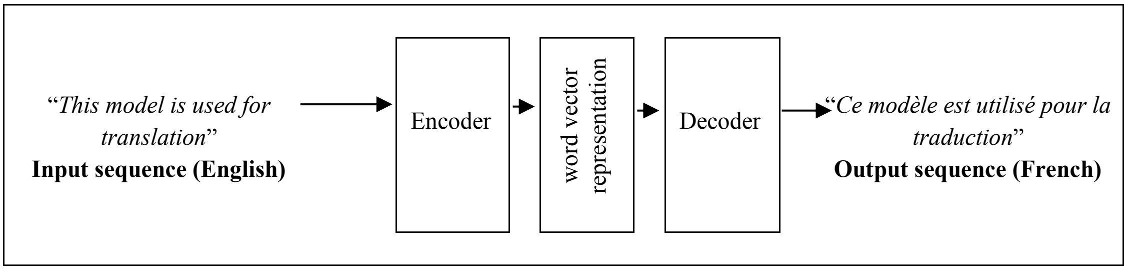
"This model is used for translation"
Input sequence (English)
Encoder
word vector representation
Decoder
"Ce modèle est utilisé pour la traduction"
Output sequence (French)

Fig. 1. Seq2Seq approach for the translation task

In this work, we propose to adapt Seq2Seq approaches to the transliteration task as well. The idea is to change the input and the output to deal with TD words in both writing systems (Latin and Arabic) in both directions. We present our Seq2Seq transliteration approach in the next Section

4. Proposed approach for Tunisian dialect transliteration

To perform the word-level TD user-generated transliteration task, we introduce a Latin TD word as an input sequence and an Arabic TD word as an output sequence for the transliteration, and vice versa for the back transliteration. We should mention that we perform the transliteration task, in this work, without considering the context of the TD word. We propose to apply the Seq2Seq approach on a sequence of characters. When we consider the pair (X,Y) as a input and output words for the transliteration task, $\{x_1..x_n\}$ as the sequence of characters in the input word and $\{y_1..y_m\}$ as the sequence of characters in the output word, we compute [39]:

$$Encoder\ (x_1, x_2, \dots, x_n) = z_1, z_2, \dots, z_n \qquad (1)$$

$\{z_1 \dots z_n\}$ are the dimensional representation of the input sequence as fixed size vectors. The conditional probability of the sequence P(Y |X) can be computed as follows:

$$P(X|Y) = \prod_{i=1}^{m} P(y_i | y_0, y_1, \dots, y_{i-1} \, ; \, z_1, z_2, \dots, z_n) \quad (2)$$

y_0 represents the word's beginning. A hidden state y_i is generated for the next character to be predicted, which goes subsequently through the softmax layer to produce a probability distribution over candidate output characters [39]. An example of the proposed TD transliteration approach is shown in Fig. 2, for the TD word "barcha – برشة (*meaning: a lot*). An example of the back transliteration is shown in Fig. 3 for the word "باهي - behi" *(meaning: good)*.

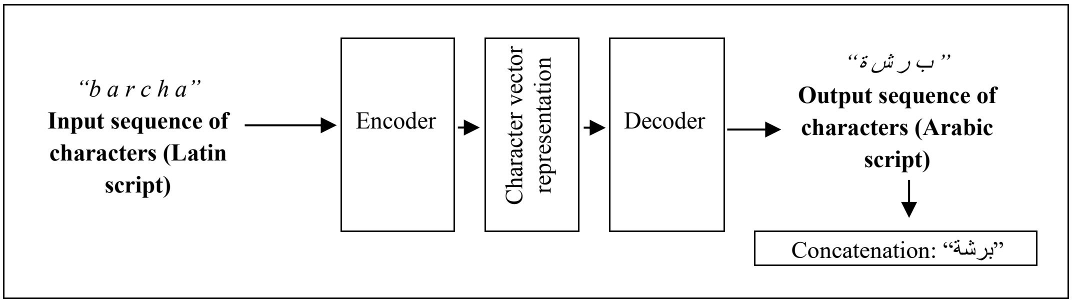
"b a r c h a"
Input sequence of characters (Latin script)
Encoder
Character vector representation
Decoder
"ب ر ش ة"
Output sequence of characters (Arabic script)
Concatenation: "برشة"

Fig. 2. Example of a Latin→Arabic TD word transliteration

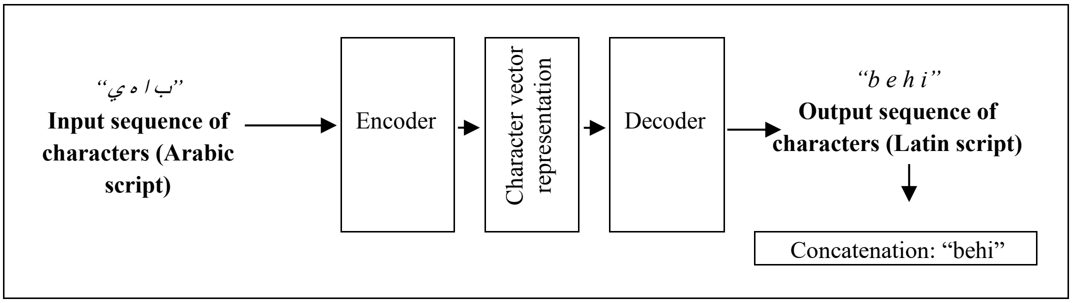
"ب ا ه ي"
Input sequence of characters (Arabic script)
Encoder
Character vector representation
Decoder
"b e h i"
Output sequence of characters (Latin script)
Concatenation: "behi"

Fig. 3. Example of an Arabic→Latin TD word transliteration

5. Experiments and results

To perform the TD transliteration task, we resorted to the corpus of [40], consisting of 6,079 messages written in the Latin script, and generated by Tunisian users in social networks. A step of cleaning and filtering was then performed to delete non-dialectal words and generate a list of Latin TD words. An annotation effort was then invested to assign the corresponding Arabic transliterations to the list of TD words. We opted for the manual annotation since no language resources were constructed for the TD transliteration task and to make sure that the used data sets are as clean as possible. Therefore, we obtained a list of 45,629 TD parallel words which we divided as shown in Table 1.

Table 1. Used data sets

Data set	#Word pairs	#Latin characters	#Arabic characters
Training / validation	36,504 (80%)	181,587	141,649
Test	9,125 (20%)	43,721	34,132

The transliteration was performed using the Neural Seq2Seq Machine Translation system[2]. We experimented using TensorFlow's embeddings for the character sequence vector representations. We first used a Vanilla Seq2Seq, then we adopted Beam Search [41] to test the model's performance. While vanilla Seq2Seq decoding generates a 1-best result [41], Beam search allows the exploration of all possible transliterations by preserving the n top candidates (n=10 in our experiments).

The results given automatically are shown in Table 2.

Table 2. Word level transliteration results before manual verification

Errors per word	Vanilla Seq2Seq (% of words)		Seq2Seq-Beam search (% of words)	
	Latin→Arabic	Arabic→Latin	Latin→Arabic	Arabic→Latin
0	92.57	70.5	92.87	71.18
1	5.76	27.35	5.55	26.54
2	0.72	1.35	0.85	1.51
> 3	0.94	0.79	0.73	0.77
Average #errors per word	0.11	0.3	0.1	0.3

We noticed that several correct TD words, used in real written conversations on Tunisian social media were automatically considered as incorrect. Since we performed an out of context transliteration, we resorted to a manual verification of the incorrect words. We have gone through all the erroneous transliterations and checked if they are

[2] An open source implementation of Neural Machine Translation (seq2seq): https://www.tensorflow.org/tutorials/seq2seq

correct out of context. For example, for the Arabic transliteration of the word “barcha” (*meaning: a lot*), we accept both “برشة” and “برشا”, as the two words are used in the written TD, depending on the user’s preference. The final evaluation results are given in Table 3.

Table 3. Word-level transliteration results after manual verification

Errors per word	Vanilla Seq2Seq (% of words)		Seq2Seq-Beam search (% of words)	
	Latin→Arabic	Arabic→Latin	Latin→Arabic	Arabic→Latin
0	94.98	96.94	95.59	98.25
1	3.65	1.85	3.22	0.38
2	0.55	0.58	0.6	0.73
> 3	0.82	0.62	0.58	0.64
Average #errors per word	0.08	0.05	0.07	0.04

Based on the results shown in Table 3, we notice that the Seq2Seq models performs well when dealing with the TD transliteration task. Indeed, we reached correct transliteration rates ranging between 94.98% to 98.25%. We notice that the Seq2Seq models with Beam search gave better results than vanilla Seq2Seq in both TD transliteration senses. Regarding the Latin to Arabic direction with Beam search, we reached 95.59% of correctly transliterated words, with an average number of errors of 0.07. For the Arabic to Latin direction, we reached a rate of 98.25% of correctly transliterated words, with an average number of errors of 0.04. Introducing the Beam search to the Seq2Seq models allowed us to explore the search space of all possible transliterations by conserving a set of top candidates in the transliteration process, which explains the result improvement.

On the other hand, we notice that the Arabic→Latin transliteration direction gave better results than the Latin→Arabic direction. The difference between the two writing systems lies primarily in the use of vowels. In the Latin script, practitioners of the TD use the letters ‘a’, ‘e’, ‘i’, ‘o’, ‘u’, and ‘y’ to designate the Arabic short vowels ‘ـَ’, ‘ـُ’, ‘ـِ’or long vowels ‘ا’, ‘و’, ‘ي’. In the Arabic script, users tend to completely omit the use of vowels.

For example, when we want to transliterate the word “chwaya – شوية” (*meaning: a little*) from Latin to Arabic, the potential transliterations can be either “شوية” with the letter ‘ة’ at the end, or “شويا” with letter ‘ا’ at the end. From Arabic to Latin, the word “شوية” can have several potential transliterations like “chwaya”, “chouaya”, chweya”, “chwiya”, “chouiya”, etc. These transliterations are all correct and used in the Tunisians’ everyday written communications in social media. This explains the good transliteration rates given in the Arabic to Latin direction.

Cases of incorrect transliterations are mainly related to the Seq2Seq principle in one hand, and to the nature of the TD language on the other hand. In fact, as we mentioned in Section 3, the Seq2Seq model's encoder reads the TD word's sequence of characters, then represents its meaning in a vector. The decoder subsequently processes this vector to emit the corresponding transliteration. The characters constituting the TD word are not, thus, processed separately, which sometimes generates a high number of errors for a single word (exceeding 13 errors in some cases). These errors include the substitution of characters or the emission of an output word having a different meaning from that of the input.

In addition, several errors are related to some characters with low frequency in the corpus such as ‘ء’, ‘ؤ’, ‘ئ’, ‘x’, ‘v’, or ‘p’ which are rarely used in the dialectal communication.

6. Conclusion

We presented in this paper a deep learning based approach for the Tunisian dialect transliteration from Latin to Arabic and from Arabic to Latin. We adopted a Sequence-to-Sequence method, originally dedicated to the translation task and used sequences of characters as inputs and outputs. We experimented using Vanilla Seq2Seq models and by introducing Beam search. We achieved an encouraging rate in both transliteration directions with Seq2Seq using Beam search and reached about 96% of correctly transliterated TD words in the Latin→Arabic direction, and 98% in the opposite direction. The transliteration task will help us construct and enrich parallel corpora and dictionaries for the TD to allow further studies and processing of the language.

We aim in our future work to experiment using other attention mechanisms. We also plan to explore the efficiency of Transformer models in the transliteration task, which would likely give better results than traditional neural Seq2Seq models.

The major limitation of our work is related to the transliteration level, as we did not take into account the context of the input word within the message in which it is contained, we only considered the context of the characters within the word to be transliterated. Therefore, we aim in future studies to explore further alternatives for in-context transliteration.

References

[1] Saadane, Houda, and Nasredine Semmar. (2012) "Utilisation de la translittération arabe pour l'amélioration de l'alignement de mots à partir de corpus parallèles français-arabe. " in *Proceedings of the Joint Conference JEP-TALN-RECITAL*, Grenoble, France, 127-140.

[2] Raj, Anand Arokia, and Harikrishna Maganti (2009) "Transliteration based Search Engine for Multilingual Information Access." in *Proceedings of CLIAWS3, Third International Cross Lingual Information Access Workshop*, Boulder, Colorado, 12–20.

[3] Chalabi, Achraf, and Hany Gergers. (2012) "Romanized Arabic transliteration." in *Proceedings of the 2nd Workshop on Advances in Text Input Methods (WTIM 2)*, Mumbai, India, 89-96.

[4] Buckwalter, Tim (2004) "Issues in Arabic orthography and morphology analysis", in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Geneva, Switzerland, 31-34.

[5] Habash, Nizar, Abdelhadi Soudi, and Tim Buckwalter. (2007) "On Arabic transliteration". in Soudi, Abdelhadi; van den Bosch, Antal; Neumann, Günter (eds.) *Arabic Computational Morphology: Knowledge-based and Empirical Methods*.

[6] Souissi, Emna and Fethi Debili. (2001) "Transliteration of Arab proper names". in *Proceedings of the 9th International Conference on Human-Computer Interaction (HCI)*, New Orleans, USA.

[7] Maleki, Jalal, and Lars Ahrenberg. (2008) "Converting Romanized Persian to the Arabic Writing Systems." in *Proceedings of the 6th International Language Resources and Evaluation (LREC)*, Marrakech, Morocco, 2904-2908.

[8] Sen, Debashis, and Kamal D. Garg. (2015) "A review on Bengali to English machine transliteration system." *International Journal of Software and Web Sciences (IJSWS)*, 60-64.

[9] Wan, Stephen, and Cornelia M. Verspoor. (1998) "Automatic English-Chinese name transliteration for development of multilingual resources." in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Quebec, Canada, 1352-1356.

[10] Darwish, Kareem (2014) "Arabizi detection and conversion to Arabic", in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, Doha, Qatar, 217-224.

[11] Marlies, Vander W., Arianna Bisazza, and Christof Monz. (2016) "A Simple but Effective Approach to Improve Arabizi-to-English Statistical Machine Translation." in *Proceedings of the 2nd Workshop on Noisy User-generated Text*, Osaka, Japan, 43-50.

[12] Ekbal, Asif, and Sivaji Bandyopadhyay. (2007) "Transliteration of named entity: Bengali and English as case study." in *Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference*, Florida, USA, 223-229.

[13] Al-Onaizan, Yaser, and Kevin Knight. (2002) "Machine transliteration of names in Arabic text." in *Proceedings of the ACL-02 workshop on Computational approaches to Semitic languages*, Philadelphia, Pennsylvania, 1-13.

[14] Habash, Nizar, Mona Diab, and Owen Rambow. (2012) "Conventional orthography for dialectal Arabic." in *Proceeding of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

[15] Bies, Ann, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Sreassel, Nizar Habash, Ramy Eskander, and Owen Rambow. (2014) "Transliteration of Arabizi into Arabic orthography: Developing a parallel annotated Arabizi-Arabic script SMS/chat corpus." in *Proceedings of Neural Information Processing Systems (NIPS)*, Doha, Qatar, 93-103.

[16] Al-Badrashiny, Mohamed, Ramy Eskander, Nizar Habash, and Owen Rambow. (2014) "Automatic transliteration of Romanized dialectal Arabic." in *Proceedings of the 18th Conference on Computational Natural Language Learning*, Baltimore, USA, 30-38.

[17] Eskander, Ramy, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. (2014) "Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script." in *Proceedings of the 1st Workshop on Computational Approaches to Code Switching*, Doha, Qatar, 1-12.

[18] Zribi, Ines, Rahma Boujelbane, Abir Masmoudi, Mariem Ellouze, Lamia Belguith and Nizar Habash. (2014) "A Conventional Orthography for Tunisian Arabic." in *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland, 2355–2361.

[19] Masmoudi, Abir, Nizar Habash, Mariem E. Khemakhem, Yannick. Esteve, and Lamia. H. Belguith. (2015) "Arabic transliteration of Romanized Tunisian dialect text: A preliminary investigation." in *Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics*, Cairo, Egypt, 608-619.

[20] Rathod, Pravin H., Manikrao L. Dhore, and R M. Dhore. (2013) "Hindi and Marathi to English machine transliteration using SVM." *International Journal on Natural Language Computing (IJNLC)* **2** (**4**): 55-71.

[21] Kang, Byung-Ju, and Key-Sun Choi. (2000) "Automatic transliteration and back-transliteration by decision tree learning." in *Proceedings of the 2nd international conference on Language Resources and Evaluation*, Athens, Greece, 227-233.

[22] Ganesh, Surya, Sree Harsha, Prasad Pingali, and Vasudeva Varma. (2008) "Statistical transliteration for cross language information retrieval using HMM alignment and CRF." in *Proceedings of the 2nd Workshop on Cross Lingual Information Access*, Hyderabad, India, 42-47.

[23] Reddy, Sravana, and Sonjia Waxmonsky. (2009) "Substring-based Transliteration with Conditional Random Fields." in *Proceedings of the 2009 Named Entities Workshop*, Suntec, Singapore, 92-95.

[24] Nabende, Peter (2009) "Transliteration system using pair hmm with weighted FSTs", in *Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP*, Suntec, Singapore, 100-103.

[25] Wang, Yu-Chun, and Richard T. Tsai. (2011) "English-Korean Named Entity Transliteration Using Statistical Substring-based and Rule-based Approaches." in *Proceedings of the 2011 Named Entities Workshop*, Chiang Mai, Thailand, 32-35.

[26] Dhore, Manikrao L., Shantanu K. Dixit, and Tushar D. Sonwalkar. (2012) "Hindi to English Machine Transliteration of Named Entities using Conditional Random Fields." *International Journal of Computer Applications* **48** (**23**): 31-37.

[27] El-Kahki, Ali, Kareem Darwish, Ahmed S. Aldein, Mohamed A. El-Wahab, Ahmed Hefny, and Waleed Ammar. (2011) "Improved transliteration mining using graph reinforcement." in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, United Kingdom, 1384-1393.

[28] El-Kahki, Ali, Kareem Darwish, Ahmed S. Aldein, and Mohamed A. El-Wahab. (2012) "Transliteration Mining Using Large Training and Test Sets." in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, 243-252.

[29] Guellil, Imane, Faiçal Azouaou, Mourad Abbas, and Fatiha Sadat. (2017) "Arabizi transliteration of Algerian Arabic dialect into Modern Standard Arabic." in *Proceedings of the 1st workshop on Social Media and User Generated Content Machine Translation*, Prague, Czech Republic.

[30] Sajjad, Hassan, Helmut Schmid, Alexander Fraser, and Hinrich Schütze. (2017) "Statistical Models for Unsupervised, Semi-Supervised and Supervised Transliteration Mining." *Computational Linguistics*, **43** (**2**): 349-375.

[31] Deselaers, Thomas, Sasa Hasan, Oliver Bender, and Hermann Ney. (2009) "A Deep Learning Approach to Machine Transliteration." in *Proceedings of the 4th Workshop on Statistical Machine Translation*, Athens, Greece, 233–241.

[32] Ameur, Mohamed Seghir Hadj, Farid Meziane, and Ahmed Guessouma. (2017) "Arabic Machine Transliteration using an Attention-based Encoder-decoder Model." *Procedia Computer Science*, **117**, 287–297.

[33] Rosca, Mihaela, and Thomas Breuel. (2016) "Sequence-to-sequence neural network models for transliteration." in *CoRR arXiv, abs/1610.09565*.

[34] Ammar, Waleed, Chris Dyer, and Noah A. Smith. (2012) "Transliteration by Sequence Labeling with Lattice Encodings and Reranking." in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Republic of Korea, 8-14.

[35] Younes, Jihene, Emna Souissi, and Hadhemi Achour. (2016) "A Hidden Markov Model for the automatic transliteration of Romanized Tunisian dialect." in *Proceedings of the 2nd International Conference on Arabic Computational Linguistics, ACLing 2016*, Konya, Turkey.

[36] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. (2014) "Sequence to Sequence Learning with Neural Networks." in *CoRR abs/1409.3215, 2014, http://arxiv.org/abs/1409.3215*.

[37] Luong, Minh-Thang, Eugene Brevdo, and Rui Zhao. (2017) "Neural Machine Translation (seq2seq) Tutorial." *https://github.com/tensorflow/nmt*.

[38] Prickett, Brandon (2017) "Vanilla Sequence-to-Sequence Neural Nets cannot Model Reduplication", *OWP Linguistics*.

[39] Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. (2016) "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation" *Corr, arXiv:1609.08144v2.*

[40] Younes, Jihene, Hadhemi Achour, and Emna Souissi. (2015) "Constructing linguistic resources for the Tunisian dialect using textual user-generated contents on the social web." in Daniel F., Diaz O. (eds) *Current Trends in Web Engineering: 15th International Conference, ICWE 2015 Workshops, (NLPIT)*, Rotterdam, Netherlands, 3-14.

[41] Neubig, Graham (2017) "Neural Machine Translation and Sequence-to-sequence Models: A Tutorial." *CoRR abs/1703.01619*.