Transliteration of Algerian Arabic dialect into Modern Standard Arabic

Imane GUELLIL
Ecole Supérieure d'Informatique ESI Alger
Ecole préparatoire des sciences et techniques Alger
i_guellil@esi.dz
i.guellil@epsta.dz

Faical AZOUAOU
Ecole Supérieure d'Informatique ESI Alger
f_azouaou@esi.dz

Mourad ABBAS
Centre de Recherche Scientifique et Technique pour le Développement de la Langue Arabe (CRSTDLA)
m_abbas04@yahoo.fr

Fatiha SADAT
Université du Québec à Montréal(UQAM)
sadat.fatiha@uqam.ca

Abstract

Machine transliteration is an important research area in the field of machine translation. Neural Machine transliteration (NMTR) is a new approach to machine transliteration that has shown promising results. However research on NMTR of Arabic has just begun to give results while no research has been done on neural transliteration of Arabic dialect written in Latin letters known by "Arabizi". In this paper, we propose a method of applying a neural transliteration based on a character-level for transliterating the Arabizi into Arabic script. Our method is composed of two important steps: 1) An Arabizi corpus construction 2) A character-based neural transliteration of Arabizi to Arabic. Evaluation was performed on internal and external dataset. The best precision obtained is 73.66% on the internal dataset and 45.35% on the external one. We also conduct the same experiments for Statistical Machine Transliteration (SMTR), which has largely been studied in the literature, albeit we found that NMTR outperforms SMTR by 2.18%.

1 Introduction

Arabic language has many variants; the formal language which is called Modern Standard Arabic (MSA) and the Dialectal Arabic (DA) which differs from one Arabic country to another. Arabic dialects are grouped into six categories: Egyptian, Levantine, Gulf, Iraqi, Maghrebi and others (Zaidan and Callison-Burch, 2014). The present study focuses on under-resourced language variants, namely Algerian dialects which belong to the Maghrebi family. Few works have been done on this set of dialects (Cotterell et al., 2014; Meftouh et al., 2015). It is reported in (Darwish and Magdy, 2014) that the language used in social media is highly dialectal. To confirm this and to analyze the Algerian dialect in social media, we automatically extract from an Algerian Facebook pages a set of 18602 commentaries. In this content, we find many messages such as: 1) "عفسة مليحة", which means "a good thing" or 2) "Mandirwalou", which means "I will do nothing". We observe that the dialect is ubiquitous in this page. Moreover, these two commentaries are written in two different Alphabets, the first one is written in Arabic script, whereas the second one is written in Latin one. This latter is known as "Arabizi" (Darwish, 2014). As a consequence, to translate these messages written in Arabizi to a more formal language like: MSA or French, we firstly have to standardize them which means, to transliterate them from Arabizi to Arabic script. Transliteration is the process of transforming text from one script or one alphabet to another (Josan and Lehal, 2010; Kaur and Singh, 2014). Machine transliteration is a very important

research area in the field of machine translation(Bhalla et al., 2013; Kaur and Singh, 2014).

Various works have been carried out on the transliteration. Some of them dealt with Arabic language (Al-Onaizan and Knight, 2002; Habash et al., 2007). However, mainly due to the increase of user generated content on social media, studies have been interested to study Arabizi than Arabic script (Chalabi and Gerges, 2012; Darwish, 2014; Habash et al., 2007). All these works handle transliteration in a statistical way. By contrast, we have not found any work handling transliteration using neural networks. Mainly due to the importance of Arabizi transliteration and the lack of study focusing on neural machine transliteration of Arabizi and based on the advantages of the neural networks, we focus in this paper on neural machine transliteration of Arabizi.

Our approach is composed of two important steps: 1) An Arabizi corpus construction. 2) A character-based neural transliteration of Arabizi to Arabic script. In order to evaluate our approach, we carry out two experiments: 1) the first one using internal data set which represents a part of PADIC's corpus (Meftouh et al., 2015) and 2) the second one using external data set which represents a part of COTTRELL's corpus (Cotterell et al., 2014). To highlight the advantages and disadvantages of our technique, we compare our results to the results obtained in statistical transliteration.

The present paper is organized as follows: In section 2, we review previous work based on the transliteration of Arabizi. In section3, we describe our proposed architecture of the transliteration model. In section 4, we present our experiments and results. After that, we analyze the results and errors in the discussion part in section 5. We finish by a conclusion and perspectives of our work in section 6.

2 Related work

Machine transliteration techniques can be divided into three modeling categories : grapheme-based model, phoneme-based model and hybrid model (Kaur and Singh, 2014).

In the grapheme model, the basic unit of a written language is considered. In the phoneme model, the authors consider the smallest significant unit of a sound and do not consider the orthographic information in the transliteration process. In the work of (Josan and Lehal, 2010), the authors performed character mapping based on phonetics sounds. The hybrid model could be the combination of the grapheme-based model with the phoneme-based model (Oh and Choi, 2005). It also could be the combination of different approaches of grapheme model (Darwish, 2014; Van der Wees et al., 2016). Because we work on the basic unit of Arabizi and that the most of research work focus on grapheme model, we choose to use this model in the rest of the paper.

The works based on grapheme model that have been done can be divided into two main categories: 1) The works on Named Entities Transliteration and 2) The works on all corpus transliteration. Concerning the Named Entity Transliteration, some researches focus on Chinese-English transliteration (Kwong, 2015; Shao and Nivre, 2016) whereas others focus on Arabic (Al-Onaizan and Knight, 2002). The main goal of our work is to proceed to transliteration of entire corpus for translating it in the future work. Then, we particularly focus on the works handling the transliteration of all corpora. These works can be divided into three categories: 1) Rules-based approaches. 2) Statistical Machine Transliteration (SMTR) approaches and 3) Neural Machine Transliteration (NMTR) approaches. For rule-based approach, we can mention the work in (Bhalla et al., 2013) which concentrates on Punjabi, the work reported in (Habash et al., 2007) which focuses on Arabic. Works in (Darwish, 2014; May et al., 2014; Van der Wees et al., 2016) applied rule-based approach only to construct their transliterated corpus. However, the majority of the works used SMTR techniques, for example the work in (Malik et al., 2013) concerning the Urdu Hindi transliteration or the one of (AbdulJaleel and Larkey, 2003) concerning Arabic transliteration. We conclude with the works in (Al-Badrashiny et al., 2014; Darwish, 2014; May et al., 2014; Van der Wees et al., 2016), which focus on Arabizi transliteration. These works split sentences into a set of words and split the words into a set of characters and focus on a character level. We finish by the category of works which represent new tendencies to handle the transliteration problem. The authors in (Almahairi et al., 2016; Jadidinejad, 2016; Rosca and Breuel, 2016; Shao and Nivre, 2016) apply neural networks to consider the transliteration problem. The authors in (Shao and Nivre, 2016) studied the Chinese-English transliteration and

compared between the SMTR and the NMTR. Their results showed that SMTR always outperforms the NMTR. In (Jadidinejad 2016), the authors presented a character based model for NMTR and showed better results compared to the baseline. Note that, the baseline was developed using the MOSES toolkit (Koehn et al., 2007). (Rosca and Breuel, 2016) focused on NMT by considering sequence to sequence. This work considers Arabic and other languages like Chinese. We conclude with the NMTR researches by the work in (Almahairi et al., 2016) that only concentrates on Arabic and presents the first results of NMTR of Arabic. However, we do not find any work on NMTR of Arabizi. We bridge this gap by proposing and implementing an approach that applies neural network technique to transliterate an Arabizi corpus.

Table 1 summarizes all the cited works based on grapheme model (on which we focus on) and our point of view.

Languages/ Research category	Named Entities	All corpus transliteration		
		Rules-based	SMTR	NMTR
All languages	(Kwong 2015; Shao and Nivre 2016)	(Kwong 2015)	(Malik, et al. 2013)	(Jadidinejad 2016; Rosca and Breuel 2016)
Arabic	(Al-Onaizan and Knight, 2002)	(Habash et al., 2007)	(AbdulJaleel and Larkey 2003)	(Almahairi et al., 2016; Rosca and Breuel, 2016)
Arabizi		(Darwish, 2014; May et al., 2014; Van der Wees et al., 2016)	(Al-Badrashiny et al., 2014; Darwish, 2014; May et al., 2014; Van der Wees et al., 2016)	**"Our proposed research is situated in this category"**

Table 1: The classification of research on transliteration based on grapheme model

3 Neural Machine Transliteration of Arabizi

In this section, we explain our methodology for the transliteration of Arabizi messages to Arabic. The methodology we propose (see Figure 1) is divided into two important steps: First, we construct an Arabizi corpus from aPADIC's corpus(Meftouh et al., 2015) in (Section 3.1) and then, we transliterate an Arabizi corpus by applying a character-level neural transliteration in (Section 3.2).

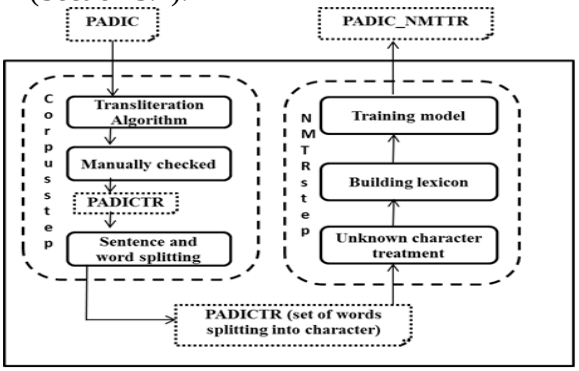

Figure 1: Neural Machine Transliteration of Arabizi steps

Our work is similar in spirit to the work in (Darwish, 2014; May et al., 2014; Van der Wees et al., 2016), but it differs from them in two points: First, we start with an Arabic corpus and construct its Arabizi corpus and not the reverse and then, we apply a NMTR instead of a SMTR like in the mentioned work. We detail below the two steps.

3.1 Arabizi corpus construction

In this first step, we build a parallel transliterated corpus containing 6233 sentences. We transliterate PADIC (Meftouh et al., 2015)(which is written in Arabic letters) to Arabizi letters. We observe that for the sound (ع), the users use the number (3) to represent it and the sound (غ) is replaced by the two letters (gh). First we define an algorithm to automatically transliterate Algerian Dialect written in Arabic letter to Arabizi form. In addition, this algorithm replaces the letter "ا", by "a" (when it appears with other letters, for example, "با" becomes "ba"). It makes the same thing for the letter "و" which is replaced by "o" and the letter "ي" which is replaced by "i" when they appear with other letters (for example "بو" becomes "bo" and "بي" becomes "bi").

This algorithm automates the first part of transliteration. For instance, the sentence: " خدمت في وحد السبيطار قريب من دارنا الحمدو لله راني لاباس و عايش مع بابا", the algorithm gives us: "**5dmt** fi **w7d**elsbitar 9rib **mn**darna**el7mdollh** rani labas w **3aychm3** baba" whereas the correct transliteration is "**5damt** fi **wa7ad**elsbitar 9rib **men**darna**el7amdolilah** rani labas w **3ayachm3a**

baba". So, among the 15 words in this sentence, the algorithm transliterates only 8 words correctly. Having these results, we manually post-edited the transliteration. At this time, we can only review 1300 sentences. So for the rest of the paper, we concentrate on a dataset containing 1300 parallel transliterate sentences that was manually checked.
Based on the work of (Darwish 2014), we divide each sentence to a set of words and each word to a set of characters, so we work on a character level. Neural Machine Transliteration based on a character level

Neural networks are powerful learning models generally divided into three kinds of architecture: Feed format, Recurrent Neural Network (RNN)(Goldberg 2015). In the work in (Goldberg 2015), the author affirms that RNN achieved very good results for language model. Mainly due to this reason, almost all researches based on neural machine transliteration use RNN(Cho et al., 2014b; Finch et al., 2015). The RNN Encoder-decoder proposed by (Cho et al., 2014a) and (Sutskever et al., 2014) is considered as the simplest version of neural machine transliteration. The idea of Encoder-decoder is to read word by word of an input sentence (in our case character by character of an input word), and encode them to a sequence of hidden states. Then the decoder, computes all the possible transliterations based on the context and generates the corresponding transliterations (Cho et al., 2014b; Jadidinejad, 2016; Kikuchi et al., 2016). In this paper, we used RNN Encoder-decoder model. To train this one, we use a development set separated from the prepared 1300 transliteration pairs to measure how well the model is generalizing during the training. Then, we use an external lexicon indicating mapping between characters and their probabilities. To create this lexicon, we use a character Alignment (Neubig, 2016).

4 Experiments and results

In this section, we present the the results of Applying NMTR on Arabizi corpus.
To test our model, based on (Neubig, 2016), we use Lamtram toolkit (Neubig, 2015), which is the combination of the two models (Bahdanau et al., 2014) and (Luong et al., 2015). We perform four types of experiments for training the model: 1) The use of only the training set. 2) The use of the train+dev set. 3) The use of the train+dev+lexicon (external lexicon, with different rates). 4) The use of deeper layer LSTM (with tain+devset+lexicon). We use Adam trainer (Kingma and Ba, 2014). Below, we present the used data and the obtained results.

4.1 Data

We focus on 1300 parallel transliterate sentences. We divide this set as follows: 1000 sentences (6444words) for training, 100 sentences (732 words) for development and 200 sentences (934 words) for test. For training, we perform our experiments on four data sets: 1)100 sentences (1078 words). 2) 250 sentences (2208 words). 3) 500 sentences (3537 words) and 4) 1000sentences (6444 words). We conduct our experiments on two test data sets: 1) internal test data set, a part of PADIC corpus (200 sentences which represents 934 words). 2) External test data set, a part of COTTREL corpus (50 sentences which represents 527 words). The transliteration of external test data set was done by an Algerian researcher team. By internal test data set, we speak about a part of sentences that we got from PADIC, so the same corpus on which we train our approach. So, we devided PADIC into three parts: The first one for training, the second one for the development and the third one for the test. By external test data set, we speak about a set of sentences that we got to test our approach, but these sentences not belong to the same corpus on which we trained our approach. These sentences belong to COTTREL corpus.

4.2 Results

For each experiment, we use epoch=10, 20, 30 and 100. Epoch represents the number of passes over the training data. However, we observe that when epoch=30, the transliteration is better for the two test data set. We present in "Table 2", in addition to the different accuracy results that we obtained from the two test data sets and the different training sets.
The accuracy is calculated by the given formula:

$$Accuracy = \frac{\mathrm{N} * 100}{(\mathrm{Total\_Number\_Of\_words})}$$

These results are for the two levels (character and word level). N stands for the number of words correctly transliterated.

Table 2 shows that the best obtained results on word level is 73.66 for internal test data

set and 45.35 for external test data set. However, we clearly observe that accuracy is higher when the training corpus size is bigger. Then, we could then suppose that this accuracy will be improved when increasing the training data set size.

Training size Sentences/ words	Accuracy level	Internal	External
100/1078	Character	67.24	46.98
	Word	59.63	38.14
250/2208	Character	75.73	49.95
	Word	69.37	38.70
500/3537	Character	78.23	51.48
	Word	70.87	39.27
1000/6444	Character	**80.97**	**55.16**
	Word	**73.66**	**45.35**

Table 2 : Neural Machine transliteration results on character and word level

5 Results comparison and error analysis

In this section, we compare our results to others cited in (section 2). We also made an error analysis with some details on the proposed solutions.

5.1 Results comparison

We have cited above that our work is the same in spirit to the work in (Darwish, 2014; May et al., 2014; Van der Wees et al., 2016). We notice that all these works are statistical based machine transliteration (SMTR). Then, to compare our results with the cited works, we apply SMTR on our data by using MOSES toolkit (Koehn et al., 2007). We use KenLM (Heafield, 2011) as a language model and GIZA++ for alignment (Och and Ney, 2000). For language model, we train the 10-gram model.

The best results obtained on word level are 71.48 for internal test data set and 44.59 for external test data set. When we use the entire size of training corpus (1000 sentences), we clearly observe that the NMTR gives better results than SMTR. To illustrate the different accuracy variation between SMTR and NMTR on internal and external dataset, we present "Figure 2" where we focus on word level to not overload the graphic.

We present on Figure 2 the accuracy of transliteration related to the corpus size. According this figure, we observe that NMTR gives better results than SMTR for the whole training corpora in the case of internal data set. However, for external data set, the NMTR may exceed SMTR only for the bigger training corpus (which contains 1000 sentences).

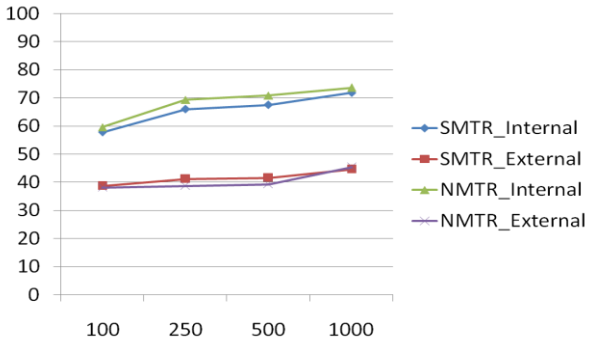

Figure 2: Comparison between NMTR and SMTR on word level

5.2 Error analysis

We distinguish between two kinds of errors; 1) errors appearing on internal test data set and 2) errors appearing on external test data set.

On internal test data set, the most important errors are related to the vowels (As shown in the words 1,3 and 4 in Table 4). The letter "a" could be transliterated as nothing (so no letters to replace it) or as the Arabic letter "ا" (or as "أ" where it is at the beginning of word). We observe the same for the letter "o" or the two letters "ou" which could be transliterated as nothing or as the letter "و".The letters "i" and "e"could be transliterated as "إ" (where they are at the begging), or as the letter "ي",or the two letters "اي". However, this kind of errors appears in NMTR less than in SMTR (As shown in the words N° 2,8 and 10 in the Table 4).We also observe the confusion between the two letters "ط" and "ت" for replacing the letter "t" and between the three letter "ص", "س" and "ز" for replacing the letter "s" (As shown in the words N° 5,6,7 and 8 in Table 4).

However, we observe that the sound "a" at the end which is often transliterated as "ة" (and means that the adjective is feminine) is recognized by NMTR but is not recognized by SMTR (As shown in the word N° 9 in Table 4). We finish by the two words N° 11 and 12 where

the correct words contain "ا" and the transliterated words contain "أ", which both are pronounced "a", so the two transliteration are correct.

To avoid these errors, we propose the following: 1) Increase the training corpus for helping NMTR to learn better models. 2) To consider different correct transliteration in some cases where two or more transliterations are correct (Like for words N°11 and 12).

N°	The arabizi word	Correct translit eration	NMT R	SMTR
1	5ayarha	خيرها	خايرها	خايرها
2	elplacard	البلاكار	الـبلاكر	البلاكرد
3	elsemastr	السوماستر	السماستر	السماستر
4	economi	ايكونومي	اكونومي	يكونومي
5	El5it	الخيط	الخيت	الخيت
6	latart	لاتارت	لاترت	لاطرت
7	swaswa	سواسوا	سوصوا	سواصوا
8	elmsayab	المصايب	المسايب	المصيب
9	malfa	مالفة	مالفة	مالف
10	lelsma	للسما	للسما	لالسما
11	A7fadhhom	احفظهم	أحفظهم	حفظهم
12	amin	آمين	أمين	مين

Table 3: Errors analysis on internal test data set

On external test data set, we first observe the occurrence of the same errors that appears in internal test data set (so related to vowels, "s" and "t" letters). However, we also observe other errors as shown in Table 4.

N°	The arabizi word	Correct transliter ation	NMTR	SMTR
1	naksou	نقصو	نكسو	نكسو
2	fikou	فيقو	فيكو	فيكو
3	hasbatlek	حسبتلك	هسبتلك	هسبتلك
4	mliha	مليحة	مليها	مليها
5	yakhi	ياخي	ياكهي	يكهي
6	khoya	خويا	كهيا	كهويا
7	promble	بروبلام	پرومبل	پرومبـل
8	ghir	خير	غير	غير

Table 4: Errors analysis on external test data set

The first category of errors related to the letter "k" which could transliterated as "ق" or as "ك"(as shown in the words N°1 and 2 in the Table 4). The second category related to the letter "h" which could be transliterated as "ح" or as "ه" (As shown in the words N° 3 and 4 in the Table 4). The third category related to the two letters "kh" which could be transliterated as "كه" or as "خ" (As shown in the words N° 5 and 6 in the Table 4). In all these cases, the system confuses between the right transliteration and the other transliteration which could be right in other case. We finish by the last category (shown in the words N° 7 and 8 in the Table 4), which is not related to the system but to "human transliterator". In this case the NMTR are right, the problem related to the reference.

Following these error analysis, we observe that they are mainly due to our technique to construct the transliterated corpus where we begin with the Arabic side to arrive to Arabizi side. In this case, models cannot analyze all cases.

6 Conclusion and Perspectives

In this paper we propose to transliterate the Arabizi messages into Arabic in the aim of automatic translation. Our proposed methodologyon neural machine transliteration of Arabizi isexplained through the following steps: 1) An Arabizi corpus construction 2) A character-based neural transliteration of Arabizi to Arabic. We carry out two experiments: 1) On internal test data set and 2) On external test data set. To compare our approach to the state of the art, we make the same experiments on SMTR. However, we observe that NMTR gives better results than SMTR, when the training corpus is big (so contains 1000 sentences).

After analyzing the different errors that occur in the transliteration process, we extract three main causes: 1) the system confusion. 2) Our transliterated corpus technique and 3) Transliterators errors. To avoid these errors and improve the results we plan in our future work to:

1) Manually check the transliteration that will be done by "transliterators".

2) Inverse the corpus construction technique, so we have to start with Arabizi corpus and transliterate it into Arabic.

3) To eliminate the system confusion, we plan to adopt a new approach based on syllabification. We keep the character level and add the sound level part. We also plan to propose a hybrid method combining rules based, statistical and neural machine transliteration at the same time.

References

Al-Onaizan, Yaser, and Kevin Knight. 2002. Machine transliteration of names in Arabic text. *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, 2002, pp. 1-13. Association for Computational Linguistics.

Almahairi, A., Cho, K., Habash, N., & Courville, A. 2016. First Result on Arabic Neural Machine Translation. arXiv preprint arXiv:1606.02680.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Bhalla, Deepti, Nisheeth Joshi, and Iti Mathur. 2013. Rule based transliteration scheme for English to Punjabi. arXiv preprint arXiv:1307.4300.

Chalabi, Achraf, and Hany Gerges. 2012. Romanized arabic transliteration.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Cotterell, Ryan, et al. 2014. An algerian arabic-french code-switched corpus. *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools*.

Darwish, Kareem. 2014. Arabizi Detection and Conversion to Arabic. *ANLP*.

Darwish, Kareem, and Walid Magdy. 2014. Arabic information retrieval. Foundations and Trends, *Information Retrieval* 7(4):239-342.

Dyer, Chris, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. *Association for Computational Linguistics.*

Finch, A., Liu, L., Wang, X., & Sumita, E. (2015, July). Neural network transduction models in transliteration generation. In Proceedings of NEWS 2015 The Fifth Named Entities Workshop (p. 61).

Goldberg, Yoav. 2015. A primer on neural network models for natural language processing. arXiv preprint arXiv:1510.00726.

Habash, Nizar, Abdelhadi Soudi, and Timothy Buckwalter. 2007. On Arabic transliteration. In *Arabic computational morphology*. pp. 15-22: Springer.

Heafield, Kenneth. 2011. KenLM: Faster and smaller language model queries. Proceedings of the Sixth Workshop on Statistical Machine Translation, 2011, pp. 187-197. *Association for Computational Linguistics.*

Jadidinejad, Amir H. 2016. Neural Machine Transliteration: Preliminary Results. arXiv preprint arXiv:1609.04253.

Josan, Gurpreet Singh, and Gurpreet Singh Lehal 2010 A Punjabi to Hindi Machine Transliteration System. *Computational Linguistics and Chinese Language Processing* 15(2):77-102.

Joshi, Hardik, Apurva Bhatt, and Honey Patel. 2013. Transliterated Search using Syllabification Approach. *Forum for Information Retrieval Evaluation*, 2013.

Kaur, Kamaljeet, and Parminder Singh. 2014. Review of Machine Transliteration Techniques. International Journal of Computer Applications 107(20).

Kikuchi, Yuta, Neubig, Graham, Sasano, Ryohei, Takamura, Hiroya and Okumura, Manabu. 2016. Controlling output length in neural encoder-decoders. arXiv preprint arXiv:1609.09552.

Kingma, Diederik, and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Koehn, Philipp, et al. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pp. 177-180. Association for Computational Linguistics.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

Malik, M.G. Abbas, Boitet, Christian, Besacier, Laurent and Bhattcharyya, Pushpak. 2013.Urdu Hindi machine transliteration using SMT. *WSSANLP2013*.

May, Jonathan, Yassine Benjira, and Abdessamad Echihabi. 2014. An Arabizi-English social media statistical machine translation system. *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, pp. 329-341.

Meftouh, Karima, Harrat, Salima, Jamoussi, Salma, Abbas, Mourad and Smaili, Kamel. 2015. Machine Translation Experiments on PADIC: A Parallel Arabic DIalect Corpus. *The 29th Pacific Asia Conference on Language, Information and Computation*.

Neubig, Graham. 2015. Lamtram: A toolkit for language and translation modeling using neural networks.

Neubig, Graham. 2016. Lexicons and minimum risk training for neural machine translation: *NAIST-CMU at WAT*2016. arXiv preprint arXiv:1610.06542.

Och, Franz Josef, and Hermann Ney. 2000. Giza++: Training of statistical translation models.

Oh, Jong-Hoon, and Key-Sun Choi. 2005. An ensemble of grapheme and phoneme for machine transliteration. *International Conference on Natural Language Processing*. 2005. pp. 450-461. Springer.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, pp. 3104-3112.

van der Wees, Marlies, Arianna Bisazza, and Christof Monz. 2016. A Simple but Effective Approach to Improve Arabizi-to-English Statistical Machine Translation. *WNUT* 2016.

Zaidan, Omar F, and Chris Callison-Burch. 2014. Arabic dialect identification. Computational Linguistics 40(1):171-202.