Noisy Uyghur Text Normalization

Osman Tursun **Ruket Çakıcı**
Computer Engineering Department
Middle East Technical University
Ankara Turkey
wusiman.tuerxun, ruken@ceng.metu.edu.tr

Abstract

Uyghur is the second largest and most actively used social media language in China. However, a non-negligible part of Uyghur text appearing in social media is unsystematically written with the Latin alphabet, and it continues to increase in size. Uyghur text in this format is incomprehensible and ambiguous even to native Uyghur speakers. In addition, Uyghur texts in this form lack the potential for any kind of advancement for the NLP tasks related to the Uyghur language. Restoring and preventing noisy Uyghur text written with unsystematic Latin alphabets will be essential to the protection of Uyghur language and improving the accuracy of Uyghur NLP tasks. To this purpose, in this work we propose and compare the noisy channel model and the neural encoder-decoder model as normalizing methods.

1 Introduction

Uyghur is an alphabetic language, whose alphabet includes 32 phones. Currently, the Uyghur is written with Perso-Arabic, Latin or Cyrillic-based scripts. The most widely used Uyghur alphabet is the modified Perso-Arabic script. However, in some situations, especially in social media, users adopt Latin letters to overcome certain limitations of the Perso-Arabic script. A major problem is that Latin letters are irregularly used as alternatives to Perso-Arabic script because mapping between Perso-Arabic script and Latin alphabet is not trivial. For example, “X”, “SH” or “Ş” are all used as alternative representations for the Perso-Arabic character ش (phoneme [ʃ]). Table 1, which based on the result of a conducted survey, shows that 15 out of 32 letters have two to four alternatives. To the best of our knowledge, although unsystematic usage of Latin-based alphabets is a well-discussed problem within Uyghur society, it does not appear in the literature. As far as we know it is only described in (Duval and Janbaz, 2006) as “unsystematic transliterations”. In this paper, we refer to this issue as **unsystematic usage of Latin alphabets** (UULA).

UULA problem is similar to text normalization, which has received attention recently (Sproat et al., 2001; Ikeda et al., 2016) because of a large amount of unnormalized text in the social media. In this work, with respect to the smallest text element, we divide the text normalization problem into two sub-categories: *word-based* and *character-based* normalization. The word-based normalization (Sproat et al., 2001; Ikeda et al., 2016) turns non-standard words such as slang, acronyms and phonetic substantiation into standard dictionary words. On the other hand, character-based normalization transform the raw text through substituting the irregularly used characters with proper ones. Character-level normalization includes problems such as diacritic restoration (DR) (Mihalcea and Nastase, 2002), de-ASCIIfication (Arslan, 2015) and so on.

UULA normalization is a character-level normalization, yet it is harder than other character-level normalization problems. It is a many-to-many mapping problem while most of the other types of character-level normalizations are one-to-many. As mentioned above, Table 1 shows 15 of 32 characters have 2 to 4 alternatives. Besides that, UULA texts suffer heavy ambiguity as well. For instance, if the sentence “I gave a Yuan” is written in Uyghur UULA as “Men bir koy berdim”, which may mean “I gave a sheep” or “I gave a Yuan”.

Table 1: Possible Latin alphabet alternatives of Uyghur Perso-Arabic alphabet.

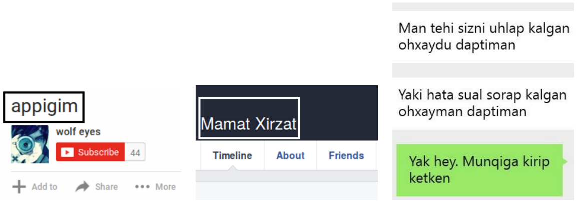
Uyghur	ژ	ئو	ئە	ق	ئى	ئۈ	چ	ئۆ	غ	خ	ش	ھ	ڭ	ئې	ۋ
Phonetics	ʒ	o	ɛ	q	i	e	t͡ʃ	ø	ʁ	χ	ʃ	h	ŋ	e	v
CTA[1]	J	O	E	Q	I	Ü	Ç	Ö	Ğ	X	Ş	H	Ñ	É	V
Alternatives	ZH, J Z, Y	O U	A E	Q K	I E	V, U O, Ü	Ç, Q CH	O, U V, Ö	G, GH H, Ğ	X H	X SH, Ş	H Y	G NG, Ñ	E, I Ë, É	V W

Table 2 shows some other cases of ambiguity. In short, UULA restoration which is addressed in this paper is a non-trivial problem.

Table 2: Examples of confusion cases.

UULA	CTA(Means)
Kan	Qan(Blood) — Kan(Mine)
Soz	Söz(Word) — Soz(Stretch)
Oruk	Oruq(Thin) — Örük(Apricot)
Soyux	Söyuş(Kiss) — Soyuş(Peel off)
Kalgin	Kelgin(come) — Qalghin (stay)

UULA restoration techniques are critical to process non-standard Uyghur text and develop a new type of input method editor (IME) that automatically suggests correctly written words and thus reduce the amount of UULA text. Figure 1 and Table 3 show several real examples of the increasing amount of UULA text on social media and the Internet. In this study we aim to 1) process and standardize the UULA text on the web so that it can be used for other NLP tasks such as information retrieval 2) help to create IMEs equipped with UULA restoration techniques that will prevent the generation of more non-standard text. Furthermore, although UULA restoration is a problem specific to the Uyghur language, the result will be useful for other character-level normalization problems and may be used for languages with similar mapping issues.



appigim

wolf eyes

Mamat Xirzat

Man tehi sizni uhlap kalgan ohxaydu daptiman

Yaki hata sual sorap kalgan ohxayman daptiman

Yak hey. Munqiga kirip ketken

(a) Video title (b) Facebook name (c) Chatting

Figure 1: Examples of UULA cases from social media.

The rest of the paper is organized as follows: we first talk about the background and related work in Section 2 and 3. Then, the methods for UULA restoration are described in Section 4. The experimental setup is given in Section 5 which is followed by results, and discussion. Finally, we talk about the conclusion and future work.

Table 3: Illustrations of examples displayed in Figure1.

Figure	UULA	CTA	Means
1a	Appigim	Apiim	My Baby
1b	Mamat irzat	Memet irzat	Uyghur Person Name
1c	Men tehi sizni uhlap kalgan ohxaydu daptiman.	Men texi sizni uxlap kalgan oxshaydu deptimen.	I thought you were sleeping.
1c	Yaki hata sual sorap kalgan ohxayman daptiman.	Yaki hata soal sorap kalgan oxshaymen deptimen.	Or I asked improper question.
1c	Yak hey. Munqiga kirip ketken.	Yak hey, munchigha kirip ketken.	Nothing happen. I was taking a bath.

2 Background and Survey

2.1 Uyghur Alphabets

Uyghur is the native language of more than 15 million Uyghur people. Currently, the modern Uyghur Perso-Arabic alphabet (UPAA) is the most used and official script of Xinjiang Uyghur Autonomous regions of China. In the last century, due to cultural and political reasons (Duval and Janbaz, 2006), Uyghurs have witnessed several reforms of the Uyghur writing system. Each of them brings certain adverse effects on Uyghur culture and society such as creating generation gaps, increasing illiteracy ratio, loss of materials written in previous scripts and so on. As a result, Uyghur society tends to refuse any new alternative scripts to the currently used UPAA. Furthermore, this social atmosphere causes unsuccessful propagation of an authentic Uyghur Latin alphabet system: Uyghur Latin alphabet (ULA), which is a project by Xin-

[1] CTA: Common Turkic Alphabet, which is composed of 34 Latin letters.

jiang University in July 2001 (Duval and Janbaz, 2006). However, many Uyghur people have not adopted or even learned this system yet.

With the digital information age, Uyghur people, especially the young generation, are starting to use Latin letters to bypass the limitation related to the UPAA in social media and the internet. There are intrinsic and extrinsic limitations of UPAA. The intrinsic limitation is that, in many new computer programs, web pages, applications etc., UPAA suffers many problems such as unqualified display, absence of IME, and so on. On the other hand, the extrinsic limitation comes from users. Many Uyghur people are not familiar with the UPAA keyboard. Additionally, some Uyghur people consider typing with UPAA input method or switching to it from the other input methods like English as inconvenient work.

Although Uyghur people use Latin letters as an alternative to UPAA, many of them have not chosen the authentic ULA as the alternative. Before and after the announcement of ULA, both systematic and unsystematic transliterations with Latin letters were actively used. According to the survey mentioned in (Duval and Janbaz, 2006), up to 18 different systematic Latin Alphabet systems existed in 2000. These are replaced by the ULA since it is announced as the official Latin alternative of UPAA. However, UULA is still very common in spite of anti-UULA propaganda. Possible explanations can be found for this from many aspects: linguistic, social, political, and so on. These discussions are not in the scope of this paper as our goal is restoring and preventing UULA texts with the aid of an automated system.

2.2 Survey

In 2016, we conducted a small e-survey[2] about how Uyghur-speaking people use Latin alphabets when writing in Uyghur. In this survey, we included questions about the participants' favorite alphabet system and Latin-based alternatives to UPAA. Besides that, we asked them to write 10 different words or phrases given in Latin-derived alphabets they personally use (Table 5).

Among 170 attenders, 39.8% mainly used UPAA, 29.7% mainly use ULA, 30.5% use UULA. However, we also discovered that Uyghur people use different scripts in different circumstances. We discover that nearly half of the people use Latin-based characters as alternatives to UPAA frequently. Nevertheless, through asking attendees to type 10 different words or sentences with Latin letters, we concluded the pattern of UULA is the one shown in Table 4. According to the table, if a sentence includes all of these characters, there will be nearly 450,000 different alternative representations of that sentence.

Table 4: Possible alternatives of corrupted characters.

Char.	Alt.	Char.	Alt.	Char.	Alt.
u	u, ö, ü	a	a, e	w	v
v	v, ö, ü	k	k, q	ch	ç
o	o, ö, ü	n	n, ñ	ng	ñ
i,	i, é	j	j, c	sh	ş
h	h, x, ğ	q	q, ç	gh	ğ
y	y, h, j	x	x, ş	zh	j
e	e, é, i	k	k, q	ë	é
g	g, ñ, ğ	z	z, j		

Table 5: Selected survey results.

Samples	Feedback
ژورنال	Jurnal, Jornal, Zhornal, Zhurnal, Zornal, Zurnal, Yornal, Yurnal
ئەقل	akil, eqil, ekil, akel
هوقۇق	huquq, hukuk, hokok, hoqoq, hoquq, hokuk
بلەيزۈك	Bilayzuk, Belayzuk, Bilayzvk, Beleyzuk, Bileyzvk, Bileyzk, Bileyzvk, Bilayzuk, Bileyzuk
چۆچۈرە	Qoqura, Chochure, Ququra, Chchre, Ququre, Qoqure, Qoqore, Chochvre, Chuchure, Qvqvra, Qoqvra, Qoqora, Chchvre, Chochore, Ququra Qoqvre, re, Chchre
مېنڭ	Menig, Mening, Mning, Mning, Mening, Mineg, Mineng, Minig, Mining
دوغاپ	dogap, doghap, dohap, dugap, dughap, dohap, duhap
گېزت قەغزى	gezit qeghizi, gezit kagizi, gizit qeghizi, gizit kagizi, gizit qeghizi, gizit kegizi, gizit kagizi, gezit kegizi, gzit qeghizi, gezit kagizi, gzit qeghizi, gezit qeghizi, gizit qegizi, gizit kagaz
خەيرى - خوش هەسەن	xeyr xosh hesen, hayri hox hasan, xeyir xosh hesen, hair hox hasan, heyir hosh hesen, xeyir xosh hesen, heyir hox hesen
هېلقى فونتان	heliki fontan, heliqi fontan, hiliki fontan, hiliqi fontan, heliqi fontan, yiliki fontan, heliki fontan, hiliki funtan, hiliki fontan, hliqi fontan

3 Related Work

This is the first study on UULA restoration to our knowledge. However, the problem is closely related to text normalization, which is the focus of

[2] available at http://goo.gl/forms/5Pi2vCeUr3

studies given in this section. With an exponential growth of noisy texts, the text normalization study has become a hot topic in NLP. In the literature, text normalization is viewed as being related to either spell-checking (Cook and Stevenson, 2009; Choudhury et al., 2007) or machine translation (Aw et al., 2006; Kobus et al., 2008; Ikeda et al., 2016). However, it is pointed out that traditional spell-checking algorithms are not very effective on some text normalization problems such as normalizing text messages like SMS, tweets, comments, etc (Pennell and Liu, 2010; Clark and Araki, 2011).

According to Kukich's early survey (Kukich, 1992) on automatic word correction, there are several types of spelling correction techniques such as minimum edit distance (Damerau, 1964), similarity key (Odell and Russell, 1918), rule-based methods (Yannakoudakis and Fawthrop, 1983), N-gram-based models (Riseman and Hanson, 1974), probabilistic (Bledsoe and Browning, 1959; Cook and Stevenson, 2009; Choudhury et al., 2007) and neural net techniques (Cherkassky and Vassilas, 1989). Among them, probabilistic models (e.g. noisy channel model) are successfully used for text normalization (Cook and Stevenson, 2009; Choudhury et al., 2007). The noisy channel model method normalizes non-standard words with the channel model and the language model, which are achieved by analyzing and processing a large corpus of noisy and formal texts.

Statistical (Aw et al., 2006), rule-based (Beaufort et al., 2010) and neural network techniques (Ikeda et al., 2016) from machine translation are used for text normalization. Since the neural machine translation (Cho et al., 2014) showed promising results, it has also been adapted to other problems such as text normalization and language correction. Xie et al. (2016) applied character-based sequence modelling with attention mechanism for language correction. The most closely related previous work to our study is Ikeda et al. (2016). They used a neural encoder-decoder model for normalizing noise in Japanese text introduced by the usage of three different writing systems. They also built a synthetic database with predefined rules for data augmentation. They compared their neural network model with rule-based methods, while we compare our neural network model with a probabilistic model.

4 Method

For UULA restoration, the aim is to recover the target sequence Y from the source sequence X. Word-based or character-based models can be used for this. In the character-based model, $X =< l_1^x, l_2^x, \ldots, l_n^x >$, $Y =< l_1^y, l_2^y, \ldots, l_n^y >$ where l_1^x is the first character of X, and n is the length of the word(s) . On the other hand, for the word-based model, $X =< w_1^x, w_2^x, \ldots, w_m^x >$, $Y =< w_1^y, w_2^y, \ldots, w_m^y >$ where m is number of words in X or Y, and w is a word. For word-based restoration, we adopt the noisy channel model. Meanwhile, we use an encoder-decoder based sequence to sequence model for character-based restoration. In fact, both of models can be character or word based. In the encoder-decoder model, to reduce the input dimension, we picked the character-based solution over the word-based. However, we choose the word-based solution for the noisy channel model because of simple implementation and robust filtering with a dictionary.

4.1 Noisy Channel Model (NCM)

Noisy channel model (Church and Gale, 1991; Mays et al., 1991) is a widely applied method for spell checking. It assumes spelling mistakes were introduced while inputs were passing through a noisy communication channel. If P is the probabilistic model of the noisy channel, then the correct word w_i^y, from the dictionary V, corresponding to the word w_i^x can be found by using the following formula:

$$\begin{aligned} w_i^y &= \operatorname*{argmax}_{w \in V} P(w|w_i^x) && (1) \\ &= \operatorname*{argmax}_{w \in V} \frac{P(w_i^x|w)P(w)}{P(w_i^x)} && (2) \\ &= \operatorname*{argmax}_{w \in V} P(w_i^x|w)P(w) && (3) \end{aligned}$$

Equation 3 shows that the target word w_i^y depends on conditional probability $P(w_i^x|w)$ and prior probability $P(w)$. $P(w)$ is calculated with the language model, while $P(w_i^y|w)$ is calculated with the error model. The error model is achieved with static analysis on real error samples. Since our error samples are created synthetically, we build the error model with the same confusion table with which we generated corrupt data. Here, the confusion table is at the character-level but we need a word-level confusion table. In order to overcome this issue, we apply the Bledsoe-Browning technique (Bledsoe and Browning, 1959). It calculates the word-level confusion probability by multiplying the confusion probability of the letters as in Equation 4.

$$P(w_i^y|w_i^x) = \prod_n^i P(l_i^y|l_i^x) \tag{4}$$

4.2 Neural Encoder-Decoder Model (NEDM)

From a different perspective, the text normalization task can be considered as a text regeneration process starting with the information extracted from noisy data. We can view text reconstruction as rewriting new text with same meaning. During generation, the text process model (encoder) extracts abstract information from un-normalized text. The generalization model (decoder) starts to generate the text once it receives information from the text processing model. The generation model is trained by maximizing the probability of the generated text, $P(Y)$. According to the chain rule, it is decomposed into:

$$P(Y) = \prod_{t=1}^{M} p(y_i|y_1, y_2, \ldots, y_{i-1}) \tag{5}$$

where M is the length of the sequence, and y_i is a unit in the sequence. Therefore, we need a model that learns the conditional distributions: $p(y_i|y_1, y_2, \ldots, y_{i-1})$.

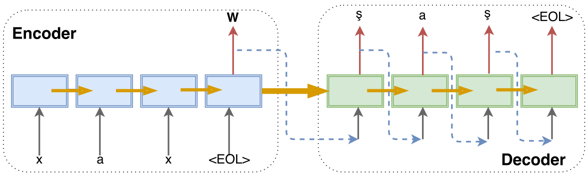


Figure 2: Encoder-decoder model.

The encoder-decoder model in (Cho et al., 2014) works in a similar fashion. It divides many-to-many mappings into many-to-one and one-to-many mappings. The encoder does a many-to-one mapping, while the decoder performs a one-to-many mapping. Both the encoder and the decoder are recurrent neural networks. One of the advantages of this model is that the encoder and the decoder are jointly trained to maximize the conditional probability, $P(Y|X)$.

$$P(Y|X) = \prod_{t=1}^{M} p(l_i^y|l_1^y, l_2^y, \ldots, l_{i-1}^y, X) \tag{6}$$

As the Figure 2 and Equation 6 show, the encoder extracts abstract information W from input X, and then the decoder starts generating target text sequentially with the information that comes from the encoder and the previous time step.

5 Experiment and Results

5.1 Dataset

In the experiments, we use both synthetic and authentic data. We train/build our models with synthetic data because of limited access to the real cases and difficulties of building ground truth. Nevertheless, we conduct tests both on synthetic and real data that we have collected. [3]

5.1.1 Synthetic Data

The synthetic dataset used in our experiments is built by scrawling raw text from news websites such as "tianshannet.com", "okyan.com" and "uycnr.cn". In total, we collected 2GB of data for training and testing, 10 text files of different genres, each of which includes around 586 words. Note that these data are written in UPAA, while we convert them to the CTA format for convenience.

The training of the encoder-decoder model uses pairs of source and target sequences. Target sequences are collected from raw text, while source sequences are created synthetically by randomly replacing letters in the target sequence using the mapping shown in Table 4. Notice that words in synthetic UULA text may include more characters than ground-truth target words. This is caused by replacing some single letters by double letters. For example, ş to sh , ç to ch, and so on. To ensure that corresponding words in source-target pairs have the same length, we pad n "w"s at the end of a target word whose corresponding source word includes n additional letters. The reason for choosing the character "w" is that it is not in CTA. Similarly, we generate the target and source text for testing. However, for more convincing test results on synthetic data, we generated 10 different source texts for each of the target text. Testing results on each of the synthetic files are the mean of 10 cases, while the final accuracy of all synthetic data is the mean of all the results on the synthetic files.

[3] The Noisy Uyghur Text Dataset used in this study is freely accessible from the URL: http://kovan.ceng.metu.edu.tr/~osman.

5.1.2 Real Data

We collect 226 sentences (1372 words in total) from social media platforms such as “Wechat”, “Facebook” and so on. For building the ground truth, we first use our model for restoration. Secondly, we restore texts manually. Finally, we apply a spell-checker for further restoring. While collecting real data and building the corresponding ground, we found that the real data has more noise than the usual UULA. We found in real data that there are various types of spelling errors, misuse of punctuation and repetitions.

5.2 Implementation Details

5.2.1 Neural Encoder-Decoder Model

We built our neural encoder-decoder model with TensorFlow (Abadi et al., 2015). Both encoder and decoder models used three layers stacked LSTMs with 256 hidden units and 256 dimension character embeddings. For training the model, the Adam optimizer with 0.0001 learning rate is applied. We trained the model in only 2 epochs with a 128 batch size. We selected the model with the best validation results on the validation set that is described below. The training process is accomplished on Tesla K40 GPU.

In this model, the length of the target and the source sequences is 30, and, instead of special tokens, blank space is placed at the beginning and the end of a sequence. Note that these sequences are constructed by grouping words in the raw text by keeping sequence length under 30. We build them as follows: First, we tokenize the text with blank space or new line character, then we append a blank space to the beginning of each token. Then, we concatenate them in order by keeping the sequence length at maximum 30. If concatenating the next word makes the current sequence length bigger than 30, then only blank spaces are appended. However, the new sequence will start from the next word. In total, we generate 63,824,760 sequences. We divide them into training, validation and testing sets in this portion: 60%, 20%, 20%.

5.2.2 Noisy Channel Model

The channel probability, in other words, the error model, in the NCM is generated according to the Table 1. For example, the probability of l_1=‘ş’ turning into l_2=‘x’, $p(l_2|l_1)$ is 1/3, since ş has three alternatives. We generate a 3-gram language model by running Kenlm language modeling tool (Heafield, 2011) on our collected text.

The Noisy channel model method normalizes the text word-by-word by selecting the most probable candidate from all possible candidates by ranking their probabilities. These candidates are generated with Table 1. For example, the word ”xax” will have 8 candidates: “xax, şax, xex, şex, xeş, şeş, şaş, xaş”, since both “x” and “a” have two alternatives. According to our experiment, on average, 1074 candidates are proposed for each word. However, we filter these candidates with the use of a dictionary. The dictionary includes all unique words from the raw text. With this dictionary filter, 1074 candidates are filtered to an average of 1.6 candidates. After filtering, a candidate is passed to the noisy channel model to find the candidate with the highest likelihood. If all candidates are filtered, then the original is kept.

5.3 Results and Analysis

The performance of two models is evaluated by conducting two tests: UULA text restoration test and the IME recommendation test. The former tests the accuracy the model on restoring documents with UULA noise. On the other hand, the latter checks a model’s prediction accuracy of the word being typed. In the IME recommendation test, we conjecture that the models have limited access to previous words. Therefore, we test two models by providing a limited number of previous words to them (at most two words in IME testing). In fact, the noisy channel model always has limited access to the previous context, therefore its results are the same for two tests.

Accuracy results of the tests are calculated as in Equation 7.

$$Accuracy = \frac{\text{\# of correct words}}{\text{\# of words}} \tag{7}$$

where “correct words” means correctly recommended or restored words. We did not calculate the precision-recall value, since the recall is always equal to 1, and precision is equal to the accuracy.

From Table 6, we can see both the neural encoder-decoder model and the noisy channel model show high performance on the synthetic dataset. However, the noisy channel model is slightly better than the encoder-decoder model. Table 7 shows that both of the models are suitable for developing IME specialized for UULA restoration. However, the 2-gram noisy channel model returns the best performance. We believe that there are three possible explanations for why the NCM outperforms the NEDM on the synthetic dataset: 1) The dictionary used in NCM is very robust, it filters out almost all of the unqualified candidates. 2) The channel model used in NCM is too ideal because it is exactly calculated not generally approximated. 3) The NEDM model needs more training with synthetic data pairs.

In the real cases as Table 8 shows, the neural encoder-decoder model is slightly better than the noisy channel model. In the real dataset, some words are not included in the dictionary, therefore noisy channel model cannot restore them correctly. Besides, other factors such as spelling errors, misuse of punctuation and redundant repeating bring more challenges to the noisy channel model as compared to the neural encoder-decoder model, since the former works at word-level but the latter at character-level.

Table 6: The results of UULA restoration on synthetic dataset (Before restoration, the accuracy is ***19.40* ± *0.03***).

Model	Accuracy (%)
NEDM	93.09 ± 2.21
NCM 1-gram	94.16 ± 0.08
NCM 2-gram	**94.54 ± 0.11**
NCM 3-gram	94.52 ± 0.11

Table 7: The results of IME recommendation on synthetic dataset.

N-gram	NEDM (%)	NCM (%)
1-gram	91.65	94.16
2-gram	94.38	**94.54**

Table 8: The results of UULA restoration on real noisy data (Before restoration, the accuracy is ***26.14 %***).

Model	Accuracy (%)
NEDM	**65.69**
NCM 2-gram	64.95

In Tables 9, 10 and Figure 3, the qualitative results are given, where both NCM and NEDM fail to restore certain noisy words. The NCM fails in restoring a noisy word when the corresponding

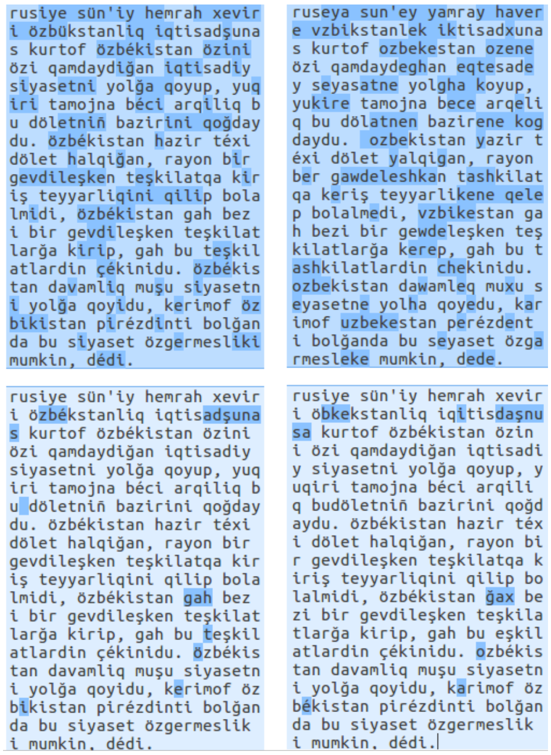
Table 9: Examples of comparison of two models and the baselines on synthetic UULA texts (Underlined means the original noisy text. *Italic* means the text is erroneously restored to non-standard text. **Bold** means the text is wrongly restored to an unwanted (but in dictionary) text).

	Sentences
UULA	pütukqilek tarehiy nayayeti uzun bir kesip. qademda orda-saraylargha, yamulgha mexsus pütvkqeler qoyulğan.
Baseline	pütükçilik tarixiy nahayiti uzun bir kesip. qedimde orda-saraylarğa, yamulğa mexsus pütükçiler qoyulğan.
NCM	pütükçilik tarixiy nahayiti uzun bir kesip. **qedemde** orda-*saraylargha*, *yamulgha* mexsus *pütvkqeler* qoyulğan.
NEDM	pütükçilik tarixiy nahayiti uzun bir kesip. **qedemde** orda-saraylarğa, yamulğa mexsus pütükçiler qoyulğan.
UULA	tulum ilgerki zamanlardeki uyghurlar saparga çeqkan vaketta ozuq-tvlvk we başka lazematlik turmux buyumlerine kaqilaydehan tëriden yasalhan halta yam xundakla kadimki uygurlar eshlitip kalgan muyem qatnax korallerining biri.
Baseline	tulum ilgirki zamanlardiki uyğurlar seperge çiqqan vaqitta ozuq-tülük ve başqa lazimetlik turmuş buyumlirini qaçilaydiğan téridin yasalğan xalta hem şundaqla qedimki uyğurlar işlitip kelgen muhim qatnaş qoralliriniñ biri.
NCM	tulum ilgirki zamanlardiki uyğurlar seperge çiqqan vaqitta ozuq-tülük ve başqa lazimetlik turmuş buyumlirini qaçilaydiğan téridin yasalğan xalta hem şundaqla qedimki uyğurlar işlitip kelgen muhim qatnaş qoralliriniñ biri.
NEDM	tulum ilgirki zamanlardiki uyğurlar seperge çiqqan vaqitta ozuq-tülük ve başqa lazimetlik turmuş buyumlirini qaçilaydiğan téridin yasalğan xalta hem şundaqla qedimki uyğurlar işlitip **qalğan** muhim qatnaş qoralliriniñ biri.

Table 10: Examples of comparison of two models and the baselines on real UULA texts (The text formatting has the same meaning as in Table 9).

	Sentences
UULA	nur xirkitinig adrisini bildihanlar bamu?
Baseline	nur şirkitiniñ adrésini bilidiğanlar barmu?
NCM	nur şirkitiniñ *adrisini bildihanlar bamu*?
NEDM	nur şirkitiniñ *adrisini* bildiğanlar *bamu*?
UULA	muxu hakta taklip pikir berilsa?
Baseline	muşu heqte teklip pikir bérilse?
NCM	muşu *hakta* teklip pikir bérilse?
NEDM	muşu heqte teklip pikir *birilse*?
UULA	chishliri chushup ketkuche eytiptu bichare ashiq boway
Baseline	çişliri çüşüp ketküçe éytiptu biçare aşiq bovay
NCM	çişliri çüşüp ketküçe éytiptu biçare aşiq bovay
NEDM	çişliri çüşüp *ketküçi* éytiptu biçare aşiq bovay

original word does not appear in the dictionary or has an ignorable N-gram score. Meanwhile, the NEDM model tends to map characters to popular patterns. Therefore, in a few cases, it restores noisy words to unexpected ones.



```
rusiye sün'iy hemrah xevir
i özbükstanliq iqtisadşuna
s kurtof özbékistan özini
özi qamdaydiğan iqtisadiy
siyasetni yolğa qoyup, yuq
iri tamojna béci arqiliq b
u döletniñ bazirini qoğday
du. özbékistan hazir téxi
dölet halqiğan, rayon bir
gevdileşken teşkilatqa kir
iş teyyarliqini qilip bola
lmidi, özbékistan gah bez
i bir gevdileşken teşkilat
larğa kirip, gah bu teşkil
atlardin çékinidu. özbékis
tan davamliq muşu siyasetn
i yolğa qoyidu, kerimof öz
bikistan pirézdinti bolğan
da bu siyaset özgermesliki
mumkin, dédi.
```

```
ruseya sun'ey yamray haver
e vzbikstanlek iktisadxuna
s kurtof ozbekestan ozene
özi qamdaydeghan eqtesade
y seyasatne yolgha koyup,
yukire tamojna bece arqeli
q bu dölatnen bazirene kog
daydu.  ozbekistan yazir t
éxi dölet yalqigan, rayon
ber gawdeleshkan tashkilat
qa keriş teyyarlikene qele
p bolalmedi, vzbikestan ga
h bezi bir gewdeleşken teş
kilatlarğa kerep, gah bu t
ashkilatlardin chekinidu.
ozbekistan dawamleq muxu s
eyasetne yolha qoyedu, kar
imof uzbekestan perézdent
i bolğanda bu seyaset özga
rmesleke mumkin, dede.
```

```
rusiye sün'iy hemrah xevir
i özbékstanliq iqtisadşuna
s kurtof özbékistan özini
özi qamdaydiğan iqtisadiy
siyasetni yolğa qoyup, yuq
iri tamojna béci arqiliq b
u  döletniñ bazirini qoğday
du. özbékistan hazir téxi
dölet halqiğan, rayon bir
gevdileşken teşkilatqa kir
iş teyyarliqini qilip bola
lmidi, özbékistan gah bez
i bir gevdileşken teşkilat
larğa kirip, gah bu teşkil
atlardin çékinidu. özbékis
tan davamliq muşu siyasetn
i yolğa qoyidu, kerimof öz
bikistan pirézdinti bolğan
da bu siyaset özgermeslik
i mumkin, dédi.
```

```
rusiye sün'iy hemrah xevir
i öbkekstanliq iqitisdaşnu
sa kurtof özbékistan özin
i özi qamdaydiğan iqtisadi
y siyasetni yolğa qoyup, y
uqiri tamojna béci arqili
q budöletniñ bazirini qoğd
aydu. özbékistan hazir téx
i dölet halqiğan, rayon bi
r gevdileşken teşkilatqa k
iriş teyyarliqini qilip bo
lalmidi, özbékistan ğax be
zi bir gevdileşken teşkila
tlarğa kirip, gah bu eşkil
atlardin çékinidu. ozbékis
tan davamliq muşu siyasetn
i yolğa qoyidu, karimof öz
békistan pirézdinti bolğan
da bu siyaset özgermeslik
i mumkin, dédi.
```

Figure 3: An example output of the neural encoder-decoder model on a subset of the synthetic UULA text (Top and bottom left part are the same ground truth, top right is the UULA, bottom right is the restoration. Blue highlights are differences.).

6 Conclusion and Future Work

In this work, we propose two models for normalizing Uyghur UULA texts. The noisy channel model views the problem as a spell-checking problem, while the neural encoder-decoder model views it as a machine translation problem. Both of them return highly accurate results on restoration and recommendation tasks on the synthetic dataset. However, their accuracy on real datat would benefit from further improvement. To improve their performance on the real dataset, one possible strategy is to consider other noisy factors appearing in the real dataset. In future work, we will update our models to handle other noisy elements such as spelling errors and the misuse of punctuation on the real dataset. However, we believe that it is easier to adapt the neural encoder-decoder model to the new challenges than the noisy channel model. This is because it only requires fine-tuning on extra data for different kinds of noise, while the noisy channel model requires redesigning of the model structure.

7 Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

Ahmet Arslan. 2015. Deasciification approach to handle diacritics in turkish information retrieval. *Information Processing & Management* .

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, pages 33–40.

Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 770–779.

Woodrow Wilson Bledsoe and Iben Browning. 1959. Pattern recognition and reading by machine. In *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*. ACM, pages 225–232.

Vladimir Cherkassky and Nikolaos Vassilas. 1989. Performance of back propagation networks for associative database retrieval. *Int. J. Comput. Neural Net* .

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International journal on document analysis and recognition* 10(3):157–174.

Kenneth W Church and William A Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing* 1(2):93–103.

Eleanor Clark and Kenji Araki. 2011. Text normalization in social media: progress, problems and applications for a pre-processing system of casual english. *Procedia-Social and Behavioral Sciences* 27:2–11.

Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text message normalization. In *Proceedings of the workshop on computational approaches to linguistic creativity*. Association for Computational Linguistics, pages 71–78.

Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM* 7(3):171–176.

Jean Rahman Duval and Walis Abdukerim Janbaz. 2006. An introduction to latin-script uyghur. In *Middle East & Central Asia Politics, Economics, and Society Conference*. pages 7–9.

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

Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Japanese text normalization with encoder-decoder model. *WNUT 2016* page 129.

Catherine Kobus, François Yvon, and Géraldine Damnati. 2008. Normalizing sms: are two metaphors better than one? In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 441–448.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)* 24(4):377–439.

Eric Mays, Fred J Damerau, and Robert L Mercer. 1991. Context based spelling correction. *Information Processing & Management* 27(5):517–522.

Rada Mihalcea and Vivi Nastase. 2002. Letter level learning for language independent diacritics restoration. In *proceedings of the 6th conference on Natural language learning-Volume 20*. Association for Computational Linguistics, pages 1–7.

M Odell and R Russell. 1918. The soundex coding system. *US Patents* 1261167.

Deana L Pennell and Yang Liu. 2010. Normalization of text messages for text-to-speech. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, pages 4842–4845.

Edward M Riseman and Allen R Hanson. 1974. A contextual postprocessing system for error correction using binary n-grams. *IEEE Transactions on Computers* 100(5):480–493.

Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer speech & language* 15(3):287–333.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727* .

Emmanuel J Yannakoudakis and David Fawthrop. 1983. The rules of spelling errors. *Information Processing & Management* 19(2):87–99.