Diacritic Restoration Using Recurrent Neural Network

Aysenur Uzun
Istanbul Technical University
Computer Engineering
Email: gencayse@itu.edu.tr

***Abstract*—Comparing to other languages, diacritic restoration is a critical task for Turkish NLP task. Turkish language has another form for some critical characters {i,o,u,s,g,c,I,O,U,S,G,C} as {ı,ö,ü,ş,ğ,ç,İ,Ö,Ü,Ş,Ğ,Ç}. Choosing proper form of these critical characters in a word is called as deasciifing or diacritic restoration. In this paper this problem is approached as sequence translation by using recurrent neural networks.**

I. Introduction

In today's world size of text that is written in natural language on social media increases and becomes very crucial source for NLP tasks. People on social media tend to shorten text or use non-Turkish keyboards. Due to these reasons text in social media is not well written and it has to be normalized to use in NLP tasks. Diacritic restoration is one of the main steps of text normalization. In Turkish alphabet there are extra 7 non-ascii characters {ı,ö,ü,ş,ğ,ç,İ,Ö,Ü,Ş,Ğ,Ç} that are another form of {i,o,u,s,g,c,I,O,U,S,G,C} characters. Deciding which form of these critical character is applicable for a word is called as "Diacritic Restoration" or "Deasciifing". In this paper, deascifiing problem is considered as a language translation. Source language input is non-normalized and asciified sentence and target language output is normalized and deasciified sentence. Recurrent Neural Networks are very popular and give promising result in sequence to sequence learning in [1][2][3]. Therefore in this project recurrent neural network architecture is chosen and approached deasciifing task as sequence to sequence translation.

II. Diacritic Restoration

In Turkish language word disambiguation is very hard task when compared to other languages. Beacuse of the Turkish is a agglutinative language, there are several surface form for a word. The words which contain critical letters cause disambiguation problem because each form of critical letter in word can be applicable. In Figure 1 which is taken from [4], a deasciified word "aci" is not valid for Turkish language and it has to be deasciified. There are four possible word which can be normalized from "aci" and they are valid for Turkish language. Choosing proper word for given context is diacritic resolution.

Another example from [4] is,

Rüyamda evde *olduğunu* gördüm. (1)
Rüyamda evde *öldüğünü* gördüm. (2)

The word "olduğunu" is a valid token for Turkish language.

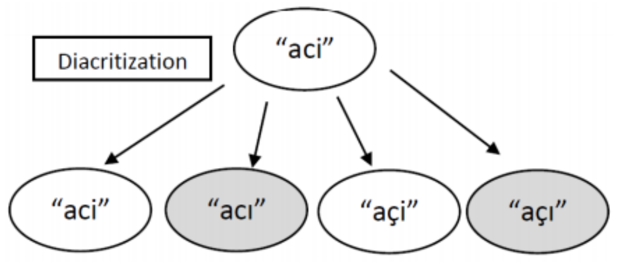

Fig. 1. Deascifiied form of word "aci"

TABLE I
Diacratization: Instance Representation for the word "olduguNu"

Curr. Letter	Neig. Word(+1)	Curr. Word	Neigh. Ch(-2)	Neigh. Ch(-1)	Neigh. Ch(+1)	Neigh. Ch(+2)	Class Label
O	GOrdUm	OldUGUnU	-	-	l	d	ö
U	GOrdUm	OldUGUnU	l	d	G	U	ü
G	GOrdUm	OldUGUnU	d	U	U	n	g
U	GOrdUm	OldUGUnU	U	G	n	U	ü
U	GOrdUm	OldUGUnU	U	n	-	-	ü

In this word there are critical letters which "o" can be "ö" and "u" can be "ü". The word "öldüğünü" is also a valid token for Turkish language and meaning of sentence is completely different. In work [4] Adali and Eryigit used CRF [5] to resolve deasciifiyn problem. CRFs produce high result for sequence labeling. In proposed model for each critical character, CRF model produces a label that is ascii or non-ascii form of critical letter. Their work simply based on character labeling. Authors extract features for each critical letter manually. These manually hand-crafted features consist current critical character, its neighbor characters and word, and label. Critical character and its features represent each instance in input file. Their input file for proposed CRF model is showed in Figure I. The test file is prepared in a same way but label part is leaved as empty. CRF predicts the proper label for test data instances. Authors stated that CRF based solution for diacritic restoration results 97,06% accuracy score.

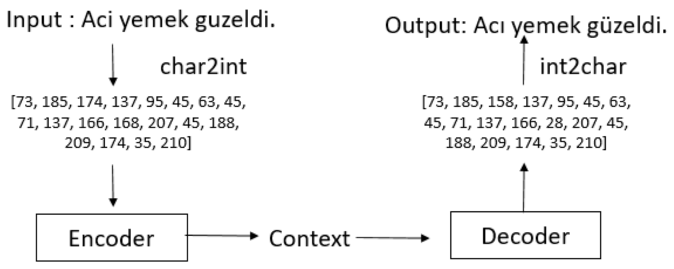

Fig. 2. Recurrent Neural Network Diacritic Restoration

III. Recurrent Neural Network for Diacritic Restoration

In [4], authors extracted features for each instance manually and they evaluate their proposed model on different setups for features. This requires quite a lot effort and knowledge about language processing. As more powerful computers are available, neural networks can be easily trained on larger dataset without put a valuable effort on feature extraction. In NLP tasks very important developments are achieved with neural network based architectures in recent years. To solve diacritic resolution problem recurrent neural network (RNN) architecture is chosen for this project. Diacritic resolution is considered as sequence to sequence translation problem. The asciified sentence which does not contain non-ascii characters is fed to RNN and model produce target language translated sentence which means deasciified sentence. Model description in given in Figure 2. Input layer takes sentence and indexes each character. Indexed characters are embedded to create character embedding vector. RNN runs on given embedding vector and produces probability of each given character.

IV. Experimental Setup

Model is designed as an non-normalized input sentence given to input layer and output layer produced to normalized sentence. In Turkish language there is a corpus which is created from Turkish Newspaper articles. Each sentence in corpus is converted to asciified form by replacing ı-¿i, ö-¿o, ş-¿s, ç-¿c, ğ-¿g, ü-¿u, Ş-¿S, Ö-¿O, Ç-¿C, İ-¿I, Ü-¿U, Ç-¿C, Ğ-¿G. Therefore, asciified input file is prepared and original corpus is used as output file. In IV, size of data is shown. Project is developed in Python environment. To create RNN architecture DyNet[1] Python module is used. Model parameters details is shown in IV Model is trained on stated corpus with given model parameters. Then model is tested with test input file. Accuracy is calculated as each correctly predicted word is count as true prediction and each falsely predicted word is count as false prediction. If a word contains 3 critical character and 2 of them predicted falsely, whole word is count as false prediction. Test result is shown in IV

[1]http://dynet.readthedocs.io/en/latest/index.html

TABLE II
Size of data when training and testing model

File	Sentence Count
Train	630000
Test	161271
Total	791271

TABLE III
Model parameters

Parameter Name	Value
# RNN Layer	2
Embedding Size	12
State Size	24

V. Conclusion

Previous works depends on models which uses hand-crafted features. Neural networks give promising results on NLP tasks and this success makes researchers to reconstruct previous works by using neural networks. Normalized data is very important for NLP tasks to achieve good result. In normalization vowelization and diacritic resolution are performed. To re-implement diacritic resolution with neural networks without hand-crafted features this project idea is proposed. Compared to CNN or MLP, RNN gives better result on sequence to sequence translation. Therefore in this project RNN model is used. Diacritic resolution or deasciifing problem is approached as sequence to sequence translation problem and for a given input sentence (non-normalized, asciified sentence), model translate input sentence to normalized and deasciified sentence. Because lack of more powerful computers, GPUs model is only trained with 3 epoch size on 630K sentences and model achieves 86% accuracy score. In further works, an enhanced RNN model with higher embedding layer and state layer can be trained with more epoch size on larger corpus. This project gives hope about with developments, Turkish diacritic resolution problem can be result with higher accuracy without human effort.

References

[1] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. ”Sequence to sequence learning with neural networks.” Advances in neural information processing systems. 2014.
[2] Graves, Alex. ”Generating sequences with recurrent neural networks.” arXiv preprint arXiv:1308.0850 (2013).
[3] Chung, Junyoung, et al. ”Empirical evaluation of gated recurrent neural networks on sequence modeling.” arXiv preprint arXiv:1412.3555 (2014).
[4] Adali, Kubra, and Gülsen Eryiğit. ”Vowel and diacritic restoration for social media texts.” Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM). 2014.

TABLE IV
Diacritic Resolution Result Comparison

System	Accuracy
# Yuret (2006)	95.93%
Zemberek(2007)	87.71%
Adali and Eryigit(2015)	97.06%
RNN Diacritic Resolution	86%

[5] John D. Lafferty, Andrew McCallum, and FernandoC. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML, pages 282–289.

[6] Yuret, Deniz, and Michael De La Maza. ”The greedy prepend algorithm for decision list induction.” International Symposium on Computer and Information Sciences. Springer, Berlin, Heidelberg, 2006.

[7] Akın, Ahmet Afsin, and Mehmet Dündar Akın. ”Zemberek, an open source nlp framework for turkic languages.” Structure 10 (2007): 1-5.