Proceedings of the 11th International Conference on Applied Informatics
Eger, Hungary, January 29–31, 2020, published at http://ceur-ws.org

Automatic Diacritic Restoration With Transformer Model Based Neural Machine Translation for East-Central European Languages

László János Laki[ab], Zijian Győző Yang[abc]

[a]MTA-PPKE Hungarian Language Technology Research Group, Hungary

[b]Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, Hungary
{laki.laszlo,yang.zijian.gyozo}@itk.ppke.hu

[c]Eszterházy Károly University, Faculty of Informatics, Hungary
yang.zijian.gyozo@uni-eszterhazy.hu

Abstract

In the last few years the size of texts written on mobile devices suddenly increased. People often type messages without diacritic marks, therefore more and more corpora are generated online which contain texts without accents. This fact causes difficulties in natural language processing (NLP) tasks. An accent restore application could be able to clean and prepare the corpus for training data of higher level NLP tools.

In our study, we created a diacritic restore method based on the state-of-the-art neural machine translation (NMT) techniques (transformer model and sentence piece tokenization [SPM]). Our system was tested on 14 languages from the East-Central European region. Most of our systems performance are above 98%. We made deeper analysis with the Hungarian system, where we could reach 99,8% relative accuracy, which is the state-of-the-art solution for Hungarian accent restoration techniques.

Furthermore, we created some multilingual models as well, where the restoration engine is able to handles all of the languages. This system has comparable performance to the monolingual ones, despite the fact that it

has much less resource requirements. Finally, a online demo was created to present our application.

Keywords: diacritic restoration, neural machine translation, NMT, transformer model, sentence piece

1. Introduction

Nowadays huge amount of written text is available on the Internet. Computer linguists have great opportunity to collect and use these data in their studies in many areas, such as machine translation, text extraction, or sentiment analysis etc. However, the highest possible quality of these texts is essential for solving above tasks efficiently.

Writing without diacritic symbols became mass phenomenon in the case of the comment sections of the social media. This type of texts are really important data, for machine learning algorithms. The commonly used natural language processing models do not work well in the case of data without accents or incorrect spelling. With an accent restoration program, we are able to restore misspelled words, which causes the quality improvement of the text processing algorithms.

In recent years, the results of neural network-based methods have outperformed the previous best systems. This is also evident in the field of language technology, so our aim was to investigate the problem of accent and diacritical character restoration with the current state-of-the-art NMT-based system.

2. Related works

During recent few years several attempts have been made to restore the accents. Mihalcea and Nastase [9] made experiments with language independent machine learning methods. One of them is when the position and the environment of the accented letter is taken into consideration. The accuracy of this approach is 95%. In their another method the distribution of different accented words was estimated from the corpus, and 98% accuracy was reached. However, the disadvantage of the system is that it is not able to handle those unknown words which are not represented in the corpus.

Charlifter was also looking for a language-independent solution [14], in which lexicon-based statistical methods were used to restore the accents. It monitors the immediate environment and applies a character-based statistical model to handle unknown words. The accuracy of the system was only 93%. Language-dependent methods were investigated for Spanish and French by Yarowsky [17] as well as for French by Zweigenbaum and Grabar [18].

For Hungarian, Németh et al. [12] presented a text-to-speech application in which they handle the words without accents. Morphological and syntactical analyzers were used to solve the problem. They results achieved 95% accuracy. Ács and Halmai [19] were created an n-gram based statistical system, which doesn't use

any kind of language dependent dictionaries. Their reported 98.36% of accuracy for Hungarian texts. Náplava et al. [11] published a bi-RNN based system, and they measured the performance for multiple languages. They reached 99.29% accuracy for Hungarian diacritic restoration.

There are some solutions in which machine translation techniques are used to solve the task of accent restoration. Novák and Siklósi [13] restored accents with statistical machine translation (SMT) methods. Experiments were executed with and without a morphological analyzer in their SMT systems. They best result - 99.06% accuracy - was achieved with morphological analyzer. In his BSc. thesis Nagy [10] used an RNN-based neural machine translation system to solve the problem. Its best result achieves 99.5% accuracy. In his work, he performed BPE (Byte pair encoding) tokenization [15] using a separate vocabulary for training sets on the source and target language sides. This study is considered to be the most similar to our work.

In our research, we use the current "state-of-the-art" transformer model instead of the RNN model, furthermore Sentence Piece (SPM) tokenization with a common dictionary instead of BPE. With this technique we are able to increase the accuracy even more.

3. Restoration of diacritic words

The essence of a corpus-based machine translation system is that it performs a transformation between source and target language sentences, with the help of parallel corpora. It is an obvious choice to use machine translation techniques to restore diacritic words, since the sentences with and without accents are almost grammatically same, they have monotone word order and similar vocabulary.

Training of the neural network requires a huge amount of training data, which is very easy to produce for the present task. The training sets were created by removing the diacritic symbols from a monolingual text.

3.1. Neural machine translation system

Statistical machine translation systems had reached their limits by the first half of the 2010s. Development of their base methods and the frameworks stopped despite the lots of invested works made by researchers. The breakthrough step was brought by [1] system, which was an attention model supported encoder-decoder architecture based NMT. The essence of the model is the separation of the translation process into two parts. First part is encoding, where an RNN-based seq2seq model is created. This model - similarly to the word embedding model - creates an n-dimensional vector from the models of the source side. This vector corresponds to the red/dark node in the middle of the Figure 1. The second phase is decoding, where the system generates the target language sentence from the previously created sentence vector using an RNN layer.

	# of segm	# of words	% of diac chars	% of diac words	diacritic symbols
bs	10,193,537	65,902,719	2.55%	13.90%	Š, Ž, Õ, Ä, Ö, Ü, š, ž, õ, ä, ö, ü
cs	28,670,112	190,069,578	9.01%	42.51%	Á, Č, Ď, É, Ě, Í, Ň, Ó, Ř, Š, Ť, Ú, Ů, Ý, Ž, á, č, ď, é, ě, í, ň, ó, ř, š, ť, ú, ů, ý, ž
et	8,644,741	53,161,686	2.75%	15.12%	Š, Ž, Õ, Ä, Ö, Ü, š, ž, õ, ä, ö, ü
hr	24,404,990	160,678,700	2.78%	15.36%	Č, Ć, Dž, Đ, Š, Ž, č, ć, dž, đ, š, ž
hu	28,680,000	177,008,412	6.59%	35.75%	Á, É, Í, Ó, Ö, Ő, Ü, Ű, á, é, í, ó, ö, ő, ü, ű
lt	1,056,994	5,378,525	5.55%	32.73%	Ą, Č, Ę, Ė, Į, Š, Ų, Ū, Ž, ą, č, ę, ė, į, š, ų, ū, ž
lv	500,000	2,838,673	7.50%	40.21%	Ā, Č, E, Ģ, I, Ķ, Ļ, Ņ, Š, Ū, Ž, ā, č, ē, ģ, ī, ķ, ļ, ņ, š, ū, ž
pl	28,146,303	178,743,981	6.01%	33.86%	Ą, Ć, Ę, Ł, Ń, Ó, Ś, Ź, Ż, ą, ć, ę, ł, ń, ó, ś, ź, ż
ro	34,111,333	256,679,157	4.84%	24.21%	Ă, Â, Î, Ș, Ț, ă, â, î, ș, ț
sk	6,352,357	40,924,451	7.40%	37.45%	Á, Ä, Č, Ď, Dž, É, Í, Ĺ, Ľ, Ň, Ó, Ô, Ŕ, Š, Ť, Ú, Ý, Ž, á, ä, č, ď, dž, é, í, ĺ, ľ, ň, ó, ô, ŕ, š, ť, ú, ý, ž
sl	13,454,728	83,828,565	2.43%	13.51%	Č, Š, Ž, č, š, ž
sq	1,452,124	9,896,094	6.97%	35.33%	Ç, Ë, ç, ë
sr	27,805,861	186,375,745	1.75%	9.72%	Č, Ć, Đ, Š, Ž, č, ć, đ, š, ž
me	50,000	312,806	3.11%	17.41%	Č, Ć, Đ, Š, Ś, Ž, Ź, č, ć, đ, š, ś, ž, ź
multi	12,550,000	80,315,253	5.07%	26.29%	

Table 1: Statistics of trainig corpora

From this time the NMT systems took over the domination from SMT. In 2017 a multi-attention NMT system - referred as transformer-based architecture - was published and made accessible by Google LLc. [16]. The attention model is a hidden layer between source and target words, and its role is to support the decoder during the generation of the target words. The essence of transformer method is that more attention layers are placed into the NMT architecture instead of one,

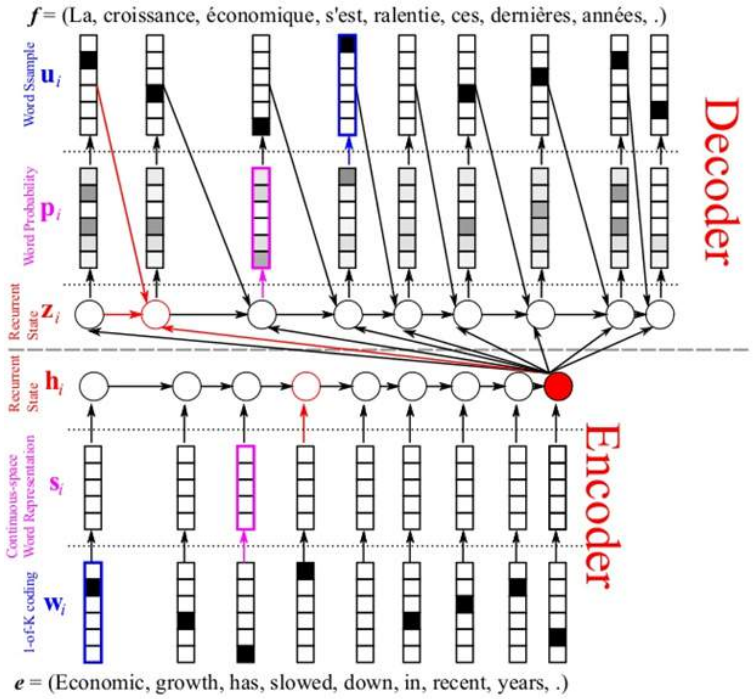

Figure 1: Schematic diagram of the encoder-decoder architecture[1]

consequently the quality of the translation of ambiguous words greatly improved.

In our study we used the framework Marian NMT [4], which is an open-source package written in C++. It has been chosen, because it is a well-documented, memory and resource optimized implementation[2], moreover easy to use it. For these reasons Marian NMT is the most commonly used framework by academic users as well as commercial developers [2].

3.2. Sentence Piece Tokenizer

Since the NMT systems are working on GPUs, one of their limitations is the size of the GPU memory. This factor defines the size of the dictionary that can be created by NMT. In a word-based system, usually the system is limited to 100K individual words, thus remaining words are handled as unknown ones.

The problem was solved by reducing the smallest translation unit from word to subword (word fragments) [15]. BPE (Byte Pair Encoding) is a data compression procedure in which the most common byte pairs are replaced by a byte that is not included in the data itself. The procedure first creates a character-based dictionary

[1] `https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-2/`
[2] https://marian-nmt.github.io/

from the corpus, where all words are represented as character sets. Secondly, the frequent character sets as stand-alone tokens based on their frequency. This process not only compresses the data, but also solves handling of unknown words, since such composition can be also created from subwords, which were originally not included in the corpus.

This method has been further developed by Kudo and Richardson [7]. They created an unattended text tokenizer and detokenizer, called Sentence Piece primarily for neural network-based machine learning tasks. BPE metric is implemented in it, which is weighted by an unigram language model [6]. By using this system the language-specific preprocessing steps – such as tokenization or lowercase – are not needed. The essence of this method is the limitation of the different "words" and elimination of the unknown words in the training set. In this way the number of the parameters in neural networks can be significantly reduced.

(1) Plain text: Petőfi Sándor egy nagyszerű költő.
SPM text: Pető fi □ S ándor □egy □nagyszerű □költő .

The example (1) shows the output of the SPM model. The words of the plain text are broken into frequently occurring character sequences. It is interesting to note that the spaces in the original sentences are also attached to the words and treated as a separate characters (□).

3.3. Demo interface

We have created a demo interface[3] (see Figure 2) to demonstrate different models. Using a drop-down menu an actual model can be selected, and there is an input field where the words can be typed. This demo examines the text typed before spaces, and if that is found to be incorrect, a suggestion is made to correct it dynamically.

4. Experiments

The theoretical base of our work were the newly available transformer and SPM technologies. Our goal was to improve the quality of diacritic restoration with these new methods.

4.1. Corpora

In our work the online available parallel corpus – called Open Subtitles[4] – was used. This corpus contains texts written in 62 languages, consists of film subtitles, but includes mainly shorter, informal sentences. We tested our system on 14 East-Center European languages, where Latin alphabet is used. The whole list of the

[3] http://nlpg.itk.ppke.hu/projects/accent
[4] http://opus.nlpl.eu/OpenSubtitles-v2018.php

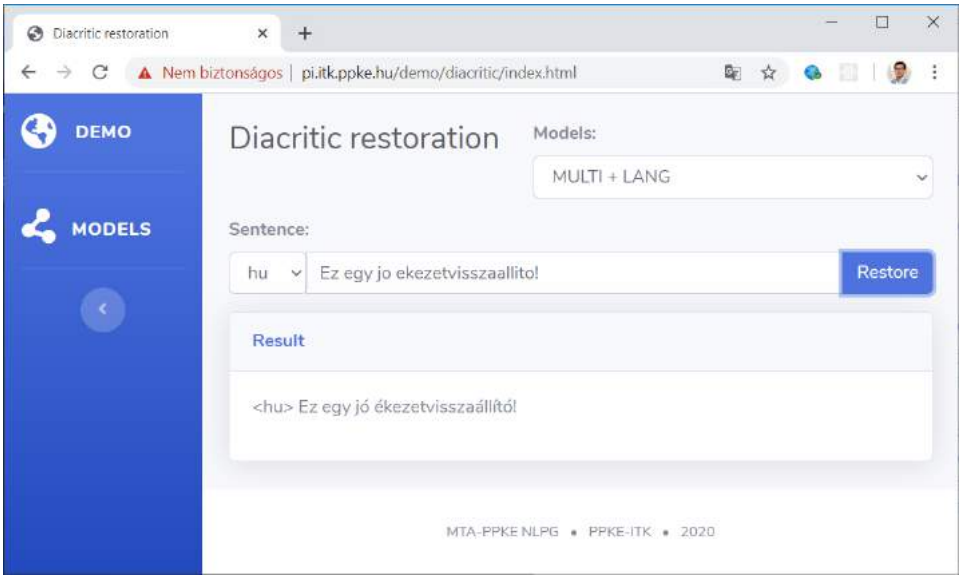

Figure 2: Demo website

selected languages, their sentence and word statistics, and diacritic symbols are shown in Table 1. We can see, that some languages are really less resourced, while most of them has more than 20 million segments. It is quite interesting to examine the ratio of the diacritic symbols in words and character sets.

4.2. NMT system

Marian NMT was applied to train and decode the NMT, in which the setup values were based on its default parameter settings. Against the related publications same SPM vocabulary were used, which force the system to tokenize source and target words in the same way. This lead the character coverage on the test corpora 100%.

The followig parameters were used to train our NMT-TM model:

- Transformer model: Size of vocabulary: 16000; Number of the encoder-decoder layers: 6; transformer-dropout: 0.1;
- learning rate: 0.0003; lr-warmup: 16000; lr-decay-inv-sqrt: 16000;
- hyperparameters: 0.9 0.98 1e-09; beam-size: 6; normalize: 0.6
- label-smoothing: 0.1; exponential-smoothing

4.3. Trained models

First of all, we created transformer NMT based diacritic restoration system for all the 14 languages separately. To create the training data all diacritic symbols were

removed from the words as a first step. Hereinafter this architecture will be referred as *NMT-TM*. To train the system we have separated 5000 segment for validation set and 3000 segment for tests from all of the training corpora. With the validation set the neural network parameters were optimised during the training. The results calculated from the separated data will be referred as *monolingual models*.

Secondly, two multilingual models were trained. We randomly selected 1-1 million segments from all language resources and mix them to one training corpora. This model called *multi1M*. The advantage of this model is the limited training time and it requires much less hardware resources, but on the other hand it has much less training data for a specific language. Furthermore the similar languages could reduce their restoration quality, but in the case of the low resourced languages this technique could help. To increase the quality of the *multi1M* system we trained a similar system, but we inserted the language codes as the first word token into the beginning of all segments, as a result the quality drop could be eliminated. This system is called *multi1M+lang*.

Thirdly, we compared our method with the previous state-of-the-art machine translation based solutions for Hungarian language case [8]. For quality comparison were trained, such as SMT without the morphological analyzer described by Novák [13] (SMT) and RNN-based neural machine translation system (NMT-RNN) [10].

To train the SMT system framework Moses [5] was used, where the language model was created with KenLM [3]. During the training default settings of the system were used, and also the word-binding and rearranging steps were leaved. These steps are not needed in the case of accent recovery as the word count and word order is the same on both - source and target - side. The text preprocessing phase consists of tokenization and "truecase" step. Trucaseing is the process, which decides if the initial word of the sentence is basically used with lowercase or uppercase. Accordingly, during the postprocessing detruecase and detokenization steps are performed. In the case of *NMT-RNN* MarianNMT was used with s2s model type trained on BPE tokenized data.

5. Results and Evaluation

Precision, recall and absolute accuracy of the word-based results were measured in our research. Since the originally correct words may change during machine translation, it is necessary to check the accuracy of the translation for all words (ALL). Furthermore these metrics were calculated only for the translation of the words which could contain diacritic characters.

In Table 1 the quality of the restoration system is shown for all monolingual models, in which 14 different translation systems were trained on monolingual data. We can see that all of our systems reached higher accuracy than 94% and half of them are above 99%. Unfortunately, we were not able to check the reference data of all system manually, since we do not have any language specialist for most of the languages. Against the huge amount of the Romanian data it reached the lowest

	monolingual models		multi1M model		multi1M + lang	
	F1	acc	F1	acc	F1	acc
et	99.29%	**99.79%**	98.68%	99.62%	98.61%	99.66%
sl	99.15%	**99.78%**	97.91%	99.48%	98.69%	99.71%
hu	99.31%	**99.61%**	97.61%	98.75%	97.71%	98.95%
lv	99.20%	**99.50%**	94.46%	96.80%	94.83%	97.55%
sr	97.07%	99.31%	95.86%	99.03%	96.62%	**99.31%**
cs	98.62%	**99.18%**	96.32%	97.84%	97.36%	98.69%
pl	98.54%	**99.15%**	97.32%	98.50%	97.60%	98.84%
hr	96.84%	99.08%	96.14%	98.89%	96.34%	**99.08%**
sk	97.89%	**98.70%**	95.94%	97.61%	97.01%	98.47%
me	94.88%	98.59%	99.20%	99.76%	99.34%	**99.84%**
lt	95.57%	97.70%	95.10%	97.48%	95.08%	**97.91%**
bs	91.17%	97.51%	90.47%	97.34%	90.75%	**97.76%**
sq	93.32%	95.33%	94.16%	95.86%	94.17%	**96.42%**
ro	89.76%	94.97%	88.20%	94.27%	88.44%	**95.03%**
multi1M			95.82%	98.16%		
multi1Mlang					95.93%	98.47%

Table 2: Results of our NMT-TM models. These numbers were calculated on all words.

quality. During the human evaluation of the training data we found, that more than 20% of the data were lack of diacritic symbols, which could be the explanation of this result.

In the other two columns the qualities of the multilingual systems are listed. For comparability of the systems the shown results were calculated on the same test set as the monolingual ones. None of the test sentences were part of the training or the validation data. We can see that the language code insertion into the beginning of the segments increases the quality of the *multi1M* system significantly, and all of them are comparable with the monolingual ones. It is really interesting to see, that *multi1M+lang* system outperforms the quality of the less effective monolingual ones, which means that it could use some information from other languages. It is really important to note, that *multi1M* system is trained on corpora, which includes only 1M-1M segments from all languages. In the last two rows the *multi1M* systems were evaluated on multilingual test sets, which were separated from their own training data.

Finally we compared our system with the previous state-of-the-art machine translation based solutions tested on Hungarian language. During our first measurement we have found that there are several misspelled words in our reference sets, so we corrected them manually in the case of the Hungarian one. The results with this modified test set are shown in Table 3. From this table we can see, that our system significantly outperforms the previous MT based systems from every point of view. We found that against *SMT* and *NMT-RNN* systems *NMT-TM* did

	All words			Diacritic words		
	prec	rec	acc	prec	rec	acc
SMT	97,96%	96,97%	98,49%	98,13%	97,04%	98,49%
NMT-RNN	97,04%	97,54%	98,58%	97,16%	97,60%	98,56%
NMT-TM	**99,38%**	**99,28%**	**99,63%**	**99,42%**	**99,33%**	**99,62%**

Table 3: Results of Hungarian models

Error type	**Ratio (piece)**	**Examples** (reference (ref) - result (res))
Correct output	**55.07% (38)**	
Equivalent form	7.26%	hova - hová (where), tied - tiéd (yours)
Replaceable output	92.74%	ref: Érdekelné ez a dolog? res: Érdekelne ez a dolog? ref: Különben nem hoznák haza. res: Különben nem hoznak haza.
Real errors	**44.93% (31)**	
Proper noun	45.16%	Liúról - Liuról, Ramával - Rámával
Wrongly replaced	54.84%	még - meg, melyen - mélyen, teli - téli

Table 4: Error analysis of the result for Hungarian language

not replace the correct words (without accent) to incorrectly spelled ones.

During the deeper analysis we found that from the 18,438 tokens (8,957 type) only 69 words differ from the reference. When the results were manually evaluated, we found that more than half of the errors (38 pieces) had equivalent meaning or correct replaceable form (e.g.: hova-hová (where); tied-tiéd (yours) etc.). The rest 31 words were incorrectly restored indeed; 14 words were foreign proper names and 17 had ambiguous meaning with and without accent. Most of these cases would need further context for disambiguation. (e.g. 2)

(2) REF: Különben nem hoznák haza. (Otherwise, they may not bring her home.)
RES: Különben nem hoznak haza. (Otherwise, they will not bring me home.)

The other advantage of the multilingual models is the significant learning time reduction. On average, one monolingual model training time was 36 hours, which for 14 languages took about 504 hours. Compare with The multilingual model training time was 33 hours.

6. Conclusion

In this research a diacritic restoration system was created based on the state-of-the-art neural machine translation techniques. First of all, this system was trained on 14 different East-Central languages. In most cases our system performs accuracy over 99%. Secondly, two multilingual models were created, which reduce the hardware requirements and training time of the system, besides gain comparable performance. Finally, our method was compared with the existing state-of-the-art Hungarian accent restoration systems. Our system reaches 99.83% relative accuracy, which significantly outperforms them. We created a demo site, where the system can be tried. In the future, we would like to extend our multilingual models with the remaining Latin based languages, such as German, French, etc.

Acknowledgements. This research was implemented with support provided by the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008).

References

[1] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015), Y. Bengio and Y. LeCun, Eds.

[2] Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., MĂźller, M., Pal, S., Post, M., and Zampieri, M. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)* (Florence, Italy, August 2019), Association for Computational Linguistics, pp. 1–61.

[3] Heafield, K. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation* (Edinburgh, Scotland, United Kingdom, July 2011), pp. 187–197.

[4] Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 116–121.

[5] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (Stroudsburg, PA, USA, 2007), ACL '07, Association for Computational Linguistics, pp. 177–180.

[6] Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 66–75.

[7] Kudo, T., and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Brussels, Belgium, Nov. 2018), Association for Computational Linguistics, pp. 66–71.

[8] Laki, L. J., and Yang, Z. G. Automatikus ékezetvisszaállítás transzformer modellen alapuló neurális gépi fordítással. *XVI. Magyar Számítógépes Nyelvészeti Konferencia* (2020), 181–190.

[9] Mihalcea, R., and Nastase, V. Letter level learning for language independent diacritics restoration. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20* (Stroudsburg, PA, USA, 2002), COLING-02, Association for Computational Linguistics, pp. 1–7.

[10] Nagy, P. Magyar nyelvű zajos szövegek automatikus normalizálása. Master's thesis, Pázmány Péter Katolikus Egyetem, 2018.

[11] Náplava, J., Straka, M., Straňák, P., and Hajič, J. Diacritics restoration using neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan, May 2018), European Language Resources Association (ELRA).

[12] Németh, G., Zainkó, C., Fekete, L., Olaszy, G., Endrédi, G., Olaszi, P., Kiss, G., and Kis, P. The design, implementation, and operation of a hungarian e-mail reader. *International Journal of Speech Technology 3*, 3 (Dec 2000), 217–236.

[13] Novák, A., and Siklósi, B. Automatic diacritics restoration for Hungarian. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), Association for Computational Linguistics, pp. 2286–2291.

[14] Scannell, K. P. Statistical unicodification of african languages. *Language Resources and Evaluation 45*, 3 (Jun 2011), 375.

[15] Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *CoRR abs/1508.07909* (2015).

[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[17] Yarowsky, D. *A Comparison of Corpus-Based Techniques for Restoring Accents in Spanish and French Text.* Springer Netherlands, Dordrecht, 1999, pp. 99–120.

[18] Zweigenbaum, P., and Grabar, N. Accenting unknown words in a specialized language. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain* (Phildadelphia, Pennsylvania, USA, July 2002), Association for Computational Linguistics, pp. 21–28.

[19] Ács, J., and Halmi, J. Hunaccent: Small footprint diacritic restoration for social media. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (Portoroz, Slovenia, 2016), pp. 3526–3529.