NLP-based Typo Correction Model for Croatian Language

Maja Mitreska[1], Kostadin Mishev[1,2] and Monika Simjanoska[1,2]
[1]iReason, LLC, Skopje, N. Macedonia
[2]Ss. Cyril and Methodius University,
Faculty of Computer Science and Engineering,
Skopje, N. Macedonia
maja.mitreska@ireason.mk, {kostadin.mishev, monika.simjanoska}@ireason.mk,
{kostadin.mishev, monika.simjanoska}@finki.ukim.mk

***Abstract*—Spelling correction plays an important role when applied in complex NLP-based applications and pipelines. Many of the existing models and techniques are developed to support the English language as it is the richest language in terms of resources available for training such models. The good occasion is that few of the methodologies provide the opportunity to adapt to other, low-resource languages. In this paper, we explore the power of the Neuspell Toolkit for training an original spelling correction model for the Croatian language. The toolkit itself comprises ten different models, but for the purposes of our work, we use the leverage of pre-trained transformer networks due to their experimentally proven spelling correction efficiency in the English language. The comparison is performed over different pre-trained Subword BERT architectures, including BERT Multilingual, DistilBERT, and XLM-RoBERTa, due to their subword representation support for the Croatian language. Furthermore, the training is done as a sequence labeling task on a newly created parallel Croatian dataset where the noisy examples are synthetically generated, and the misspelled words are labeled with their correct version. Finally, the model is tested in-vivo as part of our originally developed speech-to-text model for the Croatian language.**

***Index Terms*—Natural language processing, Typo correction, Croatian language.**

I. Introduction

Spelling correction consists of two tasks: detecting and correcting spelling mistakes. Today's data are mostly user-generated and with that comes the various possibilities of building systems with incorrect or noisy data. Thus, automatic spelling correction is very important in many complex NLP systems such as machine translation, speech recognition, text summarization, and search engines. We need these spelling correction systems, so we can build reliable and robust models whose performance will not be badly affected when the data used for building those models contain noise in a form of typos [1].

There is a lot of research that provides different spell correction architectures, from more traditional approaches such as using dictionary lookup methods and n-grams systems [2, 3] and using distance metrics that measure the similarity between two strings [4, 5, 6, 7]. These traditional techniques require a long computational time, so more researchers are navigating to the different Artificial Neural Networks.

More recent papers and researchers are using different types of Recurrent Neural Networks (RNN) individually, or in combination with other RNNs [8, 9, 10] constructing sequence-to-sequence [11] models for spelling detection and correction. With the development of these seq-2-seq models, the usage of pretrained Transformer models [12] can be seen. Some researchers are even experimenting with combinations of RNNs with Convolutional Neural Networks (CNN) [13, 14, 15]. Many of the existing models [8, 16] are built and developed to support the English language as the richest language in terms of available resources. Lately, some of those architectures provide opportunities to be re-purposed for other languages. The above-mentioned papers that explained character-based methods can easily be used with rich morphological languages whose structure heavily depends on the positions of their characters. Moreover, these models can be leveraged in end-to-end speech recognition systems. Often there are not enough audio-text pairs that are available for training the models and the performance of the language model component of such systems may perform inadequately. Thereby many new architectures strive to enhance the overall performance of the systems by including language components that are trained on text-only data or include spell checkers to improve and refine the text output [17, 18, 19].

In this paper, we propose the usage of Neuspell Toolkit [20] to train models for accurate correction of spelling mistakes in the Croatian language by using the Subword BERT architecture. This architecture uses a pretrained transformer network, BERT [21], but can easily implement similar transformer architectures such as DistilBERT [22] and XLM-RoBERTa [23]. As mentioned in the original Neuspell paper, the subword model uses averaged sub-word representations to obtain the word representations which are then fed to a classifier to predict the corrections. We are using these pretrained transformers because of their proven classification accuracy and efficiency, as we are treating this problem as a sequence labeling task. Because of the data scarcity and low resources available for the Croatian language, we are using the Croatian Language

Dataset (CLD) [1][24] whose goal is to serve as a reusable standardized Croatian dataset. The dataset is constructed with sentences from the Croatian Wikipedia dump and the Open Subtitles project. For the purposes of our research, we are synthetically noising the dataset, creating a Croatian parallel dataset for spell correction.

The rest of the paper is organized as follows. In Section II we present a brief review of the recent achievements closely related to pretraining models for typo correction in low-resource languages. In the next Section III, we introduce the architectures used for pretraining and a more detailed explanation of the aforementioned dataset. The results are presented and discussed in Section IV, and we conclude our work in the final Section V.

II. Related Work

When working with resource-scarce languages the most challenging problem is finding or creating a suitable dataset for a specific task. Due to the unavailability of the required data, the best way to create parallel data for spelling correction is to intentionally noise the original examples either using an error generator or incorporating highly probable spelling errors and using real misspoken or mistyped words [25]. This paper proposes a character-based seq-2-seq model using the LSTM architecture for spell correction in Indic languages. The described model uses a separate one-layer LSTM encoder as a corrector and two layers LSTM decoder with attention as a language model. The training dataset consists of the most frequent words in Hindu and Telugu, and movie names from both languages. The parallel dataset is created by introducing errors from a list of highly committed spelling errors from both languages. As presented in the paper, this proposed model achieves 85.4% in the Hindu language and 89.3% in the Telugu language and it gives much better performance than other baseline models for the same languages.

In [26] an additional advanced procedure is applied when a character is substituted with another one, i.e, a character is substituted with one of its most frequent misspelled versions. Moreover, because of the morphological structure of the Azerbaijani language, the tokenization is not done on a sentence level but on a character level. As for the other languages, a character-based seq-2-seq model is the most suitable architecture also for the Azerbaijani language. The architecture consists of an encoder and a decoder of LSTM layers and an attention mechanism. The evaluation of this model gives 75% accuracy when distance 0 is taken into consideration which means that the word is predicted correctly. Distance 1 to 3 means that the predicted words are some edit distance from the correct one and for those distances the accuracy increases up to 98% for the longest distance.

A slightly different architecture for low-resource languages is proposed in [27] where a knowledge-based model and a prediction model are embedded in the spell correction system. The idea is that the prediction model predicts each received word from the user and if that prediction is accepted by the user and it is corrected, the knowledge base is updated. This model is enabled not only for English but also for Spanish, Turkish, and Finnish. The knowledge model constructs a map of the misspelled words and their corrections, and the prediction model labels unseen words. For the prediction model, two methods are experimented with, an LSTM model and a simple character trigram language model (CharTriLM) where the likelihood is calculated in terms of log probabilities, and the likelihood of the word being correct is the sum of log probabilities from all its trigrams as stated in the paper. The datasets that are used are the TOEFL11 dataset for the English language, another real-world Russian dataset, and separate Wikipedia datasets for Finnish, Italian, Spanish, Turkish, and Russian. As for the results, the CharTriLM prediction model achieves better results in low-resource languages when it is evaluated both on natural data and on synthetically generated data because of the size and the low availability of resources for those languages. If more data become available it is recommended to use a combination of both prediction models.

Novel research for spell correction systems for the Vietnamese languages proposes using the Transformer architecture for training such models. The VSEC model proposed in [28] proposes the usage of Deep Learning model for the Vietnamese language instead of using the state-of-the-art statistical model which relies on the N-gram language model. Here, the spelling correction problem is considered as a machine translation task where the incorrect sentence is translated to its correct version. They train a Transformer based model on a synthetically generated dataset using a Byte-Pair Encoding (BPE) tokenizer in the pre-processing phase so the vector embedding is kept in a reasonable dimension. The dataset is extracted from a Vietnamese news corpus and additionally, a realistic dataset with true misspelled mistakes is added for the testing phase. For the task of evaluation, six metrics are used, precision, recall and F1-score for both detection and correction. In comparison with the current Vietnamese state-of-the-art models, their method gives an F1-score of 86.8% for the detection and 81.5% for the correction task which is an improvement of 1.9% and 2.2% respectively.

In terms of existing spell checkers for the Croatian language, there is a Croatian Academic Spelling Checker called Hascheck also known as Ispravi.me [29]. This system is an expert system that learns and upgrades itself every time new unseen words are received. Moreover, supervised learning is applied to preserve the purity of the vocabulary and human input is needed for the maintenance and improvement of the service. The system uses a corpus of 100 billion word occurrences and a dictionary of 2 million words-variants all confirmed in Croatian texts. Hascheck system is used for creating Croatian n-gram systems which can further be used as a database for building Croatian language technologies.

[1]Dataset available at: https://github.com/matkosoric/Croatian-Language-Dataset

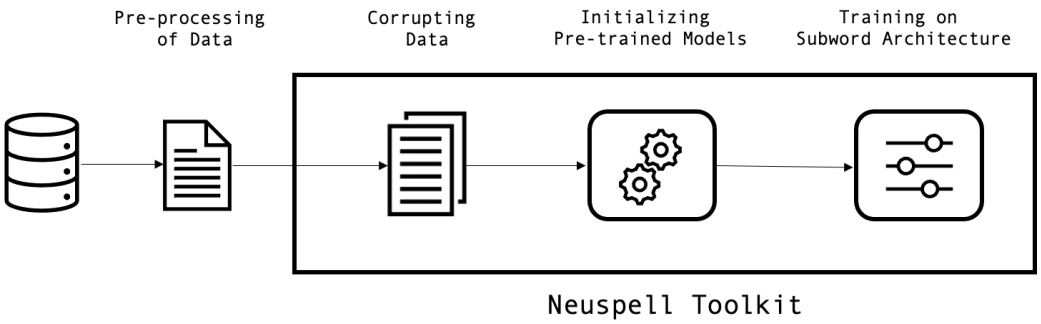

Fig. 1. The methodology pipeline.

III. Methodology

In this section, we explain our proposed methodology in detail. The first part is related to the dataset and its creation, and the second part is more concentrated on the architectures we used to build our models. The complete methodology is illustrated on Figure 1. Each part is comprehensively described in the following subsections.

A. Dataset

For the purposes of our research, i.e., creating spell correction models for the Croatian language, we used the Croatian Language Dataset (CLD), which is a public dataset in a form of Spark's dataframe. The dataset consists of about 14.7 million entries gathered from Croatian Wikipedia and the OpenSubtitles project.

Because of the size of the dataset, the prepossessing was inevitable. First, we deleted all the duplicates in the dataset since many of the entries were repeated several times, then we removed all entries that contained extra characters that are not suitable and necessary for our problem. Next, we proceeded with processing the entries that contained characters only from the Croatian alphabet and eliminated the examples that contained only numbers. Furthermore, because the size of the data decreased by only 2%, we proceeded with taking the first 10 million sentences from the dataset and filtered out the sentences that had less than five words. That way we obtained a dataset with a size of 6 million entries. For the training and testing phase, we split the dataset 80:10:10 for training, validation, and testing, correspondingly. Since this dataset only contains the original correct sentences, we needed to generate the corresponding incorrect noisy sentences to create a synthetic parallel dataset for training neural models for spell correction. As an addition to the Neuspell toolkit, there are three strategies provided for noising correct sentences. These strategies include random manipulation made with the internal characters of a word in the form of permutation, deletion, insertion, and replacement. The replacement method incorporates two approaches: word-based and character-based. The word-based approach implements a simple lookup-based replacement where a word is replaced with its most common misspelled version. On the other side, the character-based approach uses a character-level confusion matrix built of pairs of characters and a list of their most potential character replacements. All of these strategies are created and intended to be used on English corpora, and only the first-mentioned random strategy when modified can be used in generating Croatian misspellings. Thus, we modified the existing noising function to generate a synthetic parallel dataset for the Croatian language, i.e., taking into consideration the Croatian alphabet, we noised our dataset using random manipulation of the characters that comprise a word's inner structure.

B. Architecture

The Neuspell Toolkit comprises ten spell correction models. Two of them are off-the-shelf non-neural models, three are already existing LSTM models and one is a pretrained transformer network. The last four are Neuspell's authors' original models.

They represent a combination of one of the LSTM models with two different deep contextual representations of pretrained ELMO [30] and BERT. Because the current implementation of the toolkit does not provide training models on a custom dataset for their original combination of models, we decided to try the Subword BERT implementation of the BERT architecture. Due to the proven efficiency of the BERT model, we tried the Multilingual BERT model as we are working with a non-English dataset. Furthermore, to compare the results, we trained a DistilBERT model on one side as representative of distilled or smaller models and the XLM-RoBERTa model on the other side due to its proven significance in multilingual and non-English NLP problems.

We experiment with different batch sizes and vocabularies to observe the performances of the models. The size of the set of unique words which appear in the dataset, known as the vocabulary, depends on various other hyperparameters.

1) BERT: BERT (Bidirectional Encoder Representations from Transformers) is a language model that is very important in various areas in the field of NLP. The key features that make BERT so empirically powerful in obtaining new state-of-the-art results on multiple natural language processing tasks are its

TABLE I
RESULTS OF THE MODELS DURING TRAINING AND TESTING PHASE

		Accuracy		Word Rate Correction (Recall)	Precision	F1
Models	**Type of test data**	**Training**	**Testing**	**Testing**	**Testing**	**Testing**
DistilBERT	less corrupted	0.902861	0.875846	0.724485	0.929724	0.814373
DistilBERT	more corrupted	0.902861	0.856941	0.722325	0.961643	0.824978
BERT Multilingual	less corrupted	0.911237	0.909300	0.806377	0.944196	0.869862
BERT Multilingual	more corrupted	0.911237	0.897053	0.804493	0.969810	0.879450
XLM-RoBERTa	less corrupted	0.988281	0.943634	0.862252	0.988580	0.921105
XLM-RoBERTa	more corrupted	0.988281	0.931534	0.860914	0.993769	0.922583

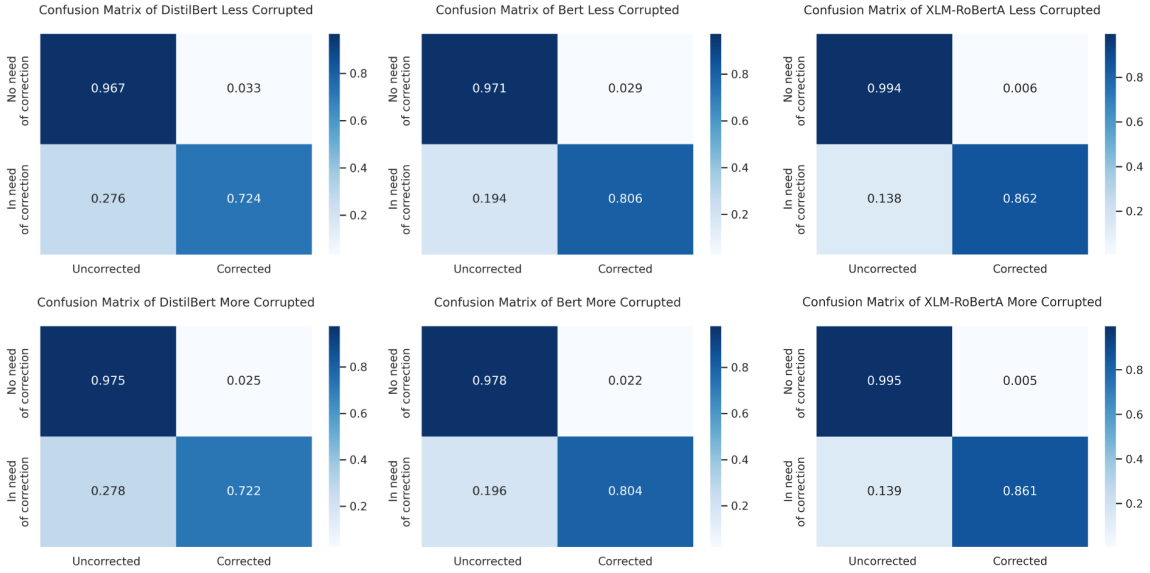

Fig. 2. Confusion matrices representing the efficiency of the models.

method for producing bidirectional language representations and the way of its pretraining [21]. BERT is pretrained using two unsupervised tasks, more precisely Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

At the same time, another important feature of this model is that its architecture remains unified no matter the downstream task that it is fine-tuned. So, the problem of spell correction is considered as a sequence labeling downstream task. The only small difference in the Neuspell implementation in comparison with the original BERT implementation is that the sub-word representations are averaged to obtain the word representations. Thus obtained representations are fed to a classifier that predicts the correct labels, i.e. correct word. Since we are working with the Croatian language, the multilingual version of BERT is used. The fine-tuning of the model over our previously described dataset was made within two epochs with a batch size of 128 and a vocabulary of 200000 words. Everything else remained as in the original paper.

2) DistilBERT: The DistilBERT model is a compressed or distilled version of the original BERT. Using transfer learning, this smaller model is trained on the same corpus as BERT and uses the same architecture. The only difference is that DistilBERT applies modern linear algebra frameworks to achieve high optimization. With that, the final result is a model that is 60% faster and 40% smaller and lighter and reaches similar performances on many downstream tasks.

To fine-tune this model on our dataset as a spell correction system we just initialized it through Neuspell's Subword BERT architecture with a batch size of 256 and a vocabulary of 100000 words.

3) XLM-RoBERTa: XLM [31] is a Transformer-based cross-lingual language model (XLM) that uses two methods to learn the representations of the words, one unsupervised that uses monolingual data, and one supervised method which leverages parallel data. The model is trained with the masked and casual language modeling (CLM and MLM) objective and a new cross-lingual objective called translation language modeling (TLM) objective which is an extension of MLM to pairs of parallel sentences. This way the model is forced to learn similar representations for different languages. The RoBERTa model [32] is just a BERT model that is robustly optimized which means that all the hyper-parameters and their effects, better design choices, and training strategies are carefully evaluated to improve its performance on downstream tasks. The XLM-RoBERTa model is a multilingual model that is pre-trained in 100 different languages. This model follows the XLM approach and the only changes that are introduced exist to improve performance at scale. The RoBERTa appendix

comes from its training routine which is the same as the original RoBERTa model only considering the MLM objective. The model is fine-tuned using the Neuspell implementation using a batch size of 64 and a set of unique words that appear in the training dataset - a vocabulary of 100000 words during 2 epochs.

C. Evaluation Metrics

The evaluation of the models and their comparison is made using accuracy and word rate correction. With accuracy, we measure the percentage of correct words of all the words in a sentence and with word rate correction we measure the percentage of corrected words over the words that are indeed in need of correction. In other words, word rate correction represents how good the model is at correcting only misspelled words. The word rate correction metric is just renamed metric equivalent to the recall metric. In these kinds of problems, accuracy is not always the best metric to measure the performances of the models since the goal is to build models that aim to correct only misspelled words, and with accuracy, all the corrections are taken into consideration even if it means correcting words into their correct version. Therefore, we incorporate the precision and F1-score to gain a clearer understanding of the outcomes of each of the distinct models.

IV. Results

In this section, we are comparing the multiple different architectures that were described above. As explained, we used three different models that we fine-tuned on the Croatian parallel dataset to detect and correct misspelled words. Each of the models is trained on a more corrupted dataset, meaning a dataset that consists of parallel examples where the syntactically generated noisy sentence contains more words that are intentionally made incorrect, i.e., a higher percentage of the words are misspelled. The models are tested on two different datasets, where one is less corrupted meaning a lower percent of the words in the sentences contain misspelled words and noise, and the other is more corrupted, created in the same way as the training set described above.

In the following example, we present how sentences are noised with the less corrupted and more corrupted strategy. The words marked with red are the words that are intentionally syntactically misspelled.

Example: Ivan uči matematiku dok rješava zadatke iz radne bilježnice.
Less corrupted sentence: Ivan uči matmatiku dok rešava zadatke iz radne bilježnice.
More corrupted sentence: Ivan uci matemtiku dok resjava zdaatke iz rade bliježnice.

Table I represents the obtained results during training and testing the models. As seen from Table I, the best scores are obtained with the XLM-RoBERTa model with an accuracy of 94.36% and 93.15% for the less corrupted and the more corrupted dataset respectively. It can be noticed that the accuracy for the less corrupted dataset is slightly higher than the accuracy for the more corrupted dataset for all three models. That is the case since the accuracy takes into consideration all of the matches, including the correct words that remain unchanged, and in the less corrupted dataset, the percentage of correct words is higher than the percentage of the incorrect words. Therefore, we additionally used the word rate correction metric i.e recall in the testing phase as a true measure of the performances of the models. Once more, XLM-RoBERTa gives the best results when evaluated with the second metric, i.e., it gives around 86% word rate correction in comparison with the 80% and 72% for the other models, BERT and DistilBERT. Moreover, it can be noticed that word rate correction is slightly lower for the more corrupted dataset. The models give higher scores when evaluated on the less corrupted dataset because when the models are evaluated on a more corrupted dataset they try to correct as many words as possible, and sometimes that means changing incorrect words into different incorrect words. So in this case, all models have a higher percentage of falsely converting incorrect words and consequently have a lower word rate correction on the more corrupted dataset. Additionally, the models are evaluated with precision and F1-score to obtain better insight into their performances. Once again, XLM-RoBERTa surpasses the other models. The results are depicted in Figure 2 where we present confusion matrices of the models when evaluated on both datasets. The rows present the actual classes of the words (whether they need to be corrected or not), and the columns present the models‘ predictions if they performed correction on the specific word or not. The matrices prove the efficiency of the typo correction models for detecting incorrectly written words and translating them into their valid form as their percentage is higher than 96% for not changing the correct words and more than 72% accuracy for correcting the words with typos. The best-performing model is XLM-RoBERTa since it shows better word rate correction and better performance overall evaluation metrics.

V. Conclusion

In this paper, we proposed using already existing and pre-trained architectures for English spelling correction to build a spell correction system in the Croatian language. Our method shows that different Deep Learning implementations with the right modifications have the potential to be leveraged for languages other than English. With this work, we prove that spell correction systems for different languages do not have to rely on current traditional and statistical methods, and the human contribution and input can be reduced when a Deep Learning method is applied.

The scores on two differently corrupted datasets reveal that the XLM-RoBERTa model when trained on subword architecture gives the highest results. In future work, we plan to include real frequent Croatian misspelled words to improve the accuracy of the recent model, so it can be used as part of a complex speech-to-text system for the Croatian language.

References

[1] Yonatan Belinkov and Yonatan Bisk. "Synthetic and natural noise both break neural machine translation". In: *arXiv preprint arXiv:1711.02173* (2017).

[2] Daniel Hládek, Ján Staš, and Matúš Pleva. "Survey of automatic spelling correction". In: *Electronics* 9.10 (2020), p. 1670.

[3] Karen Kukich. "Techniques for Automatically Correcting Words in Text". In: 24.4 (1992). ISSN: 0360-0300. DOI: 10.1145/146370.146380. URL: https://doi.org/10.1145/146370.146380.

[4] Fred J Damerau. "A technique for computer detection and correction of spelling errors". In: *Communications of the ACM* 7.3 (1964), pp. 171–176.

[5] Aouragh Si Lhoussain, GUEDDAH Hicham, and YOUSFI Abdellah. "Adaptating the levenshtein distance to contextual spelling correction". In: *International Journal of Computer Science and Applications* 12.1 (2015), pp. 127–133.

[6] Julian R. Ullmann. "A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words". In: *The Computer Journal* 20.2 (1977), pp. 141–147.

[7] Farag Ahmed, Ernesto William De Luca, and Andreas Nürnberger. "Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness". In: *Polibits* 40 (2009), pp. 39–48.

[8] Keisuke Sakaguchi et al. "Robsut wrod reocginiton via semi-character recurrent neural network". In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[9] Hao Li et al. "Spelling error correction using a nested RNN model and pseudo training data". In: *arXiv preprint arXiv:1811.00238* (2018).

[10] Gheith A Abandah, Ashraf Suyyagh, and Mohammed Z Khedher. "Correcting Arabic Soft Spelling Mistakes using BiLSTM-based Machine Learning". In: *arXiv preprint arXiv:2108.01141* (2021).

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215.

[12] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762.

[13] Yoon Kim et al. "Character-aware neural language models". In: *Thirtieth AAAI conference on artificial intelligence*. 2016.

[14] Shaona Ghosh and Per Ola Kristensson. *Neural Networks for Text Correction and Completion in Keyboard Decoding*. 2017. arXiv: 1709.06429.

[15] Shamil Chollampatt and Hwee Tou Ng. "A multilayer convolutional encoder-decoder neural network for grammatical error correction". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[16] Jung-Hun Lee, Minho Kim, and Hyuk-Chul Kwon. "Deep Learning-Based Context-Sensitive Spelling Typing Error Correction". In: *IEEE Access* 8 (2020), pp. 152565–152578. DOI: 10.1109/ACCESS.2020.3014779.

[17] H. M Mahmudul Hasan et al. "A Spell-checker Integrated Machine Learning Based Solution for Speech to Text Conversion". In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2020, pp. 1124–1130. DOI: 10.1109/ICSSIT48917.2020.9214205.

[18] Jinxi Guo, Tara N. Sainath, and Ron J. Weiss. *A spelling correction model for end-to-end speech recognition*. 2019. arXiv: 1902.07178 [eess.AS].

[19] Xiaoqiang Wang et al. *A Light-weight contextual spelling correction model for customizing transducer-based speech recognition systems*. 2021. arXiv: 2108.07493 [cs.CL].

[20] Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. *NeuSpell: A Neural Spelling Correction Toolkit*. 2020. arXiv: 2010.11085.

[21] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805.

[22] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108.

[23] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: 1911.02116.

[24] Matko Soric. *Croatian Language Dataset*. 2019. URL: https://github.com/matkosoric/Croatian-Language-Dataset.

[25] Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. "Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning". In: 2018, pp. 146–152. DOI: 10.18653/v1/P18-3021. URL: https://aclanthology.org/P18-3021.

[26] Ahmad Ahmadzade and Saber Malekzadeh. *Spell Correction for Azerbaijani Language using Deep Neural Networks*. 2021. arXiv: 2102.03218.

[27] Yiyuan Li, Antonios Anastasopoulos, and Alan W Black. *Comparison of Interactive Knowledge Base Spelling Correction Models for Low-Resource Languages*. 2020. arXiv: 2010.10472.

[28] Dinh-Truong Do et al. *VSEC: Transformer-based Model for Vietnamese Spelling Correction*. 2021. arXiv: 2111.00640.

[29] Šandor Dembitz, Gordan Gledec, and Ivan Srdić. *Hrvatski akademski spelling checker*. 2022. URL: https://ispravi.me/.

[30] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: 1802.05365.

[31] Guillaume Lample and Alexis Conneau. *Cross-lingual Language Model Pretraining*. 2019. arXiv: 1901.07291.

[32] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692.