




A novel hybrid algorithm for morphological analysis: artificial Neural-Net-XMOR

Ayla KAYABAŞ^{1,*}, Ahmet E. TOPCU², Özkan KILIÇ³

¹Department of Computer Engineering, Faculty of Engineering, Kırşehir Ahi Evran University, Kırşehir, Turkey

²College of Engineering and Technology, American University of the Middle East, Kuwait

³Cisco, San Jose, California, USA

Received: 26.08.2021

Accepted/Published Online: 27.05.2022

Final Version: 22.07.2022

Abstract: In this study, we present a novel algorithm that combines a rule-based approach and an artificial neural network-based approach in morphological analysis. The usage of hybrid models including both techniques is evaluated for performance improvements. The proposed hybrid algorithm is based on the idea of the dynamic generation of an artificial neural network according to two-level phonological rules. In this study, the combination of linguistic parsing, a neural network-based error correction model, and statistical filtering is utilized to increase the coverage of pure morphological analysis. We experimented hybrid algorithm applying rule-based and long short-term memory-based (LSTM-based) techniques, and the results show that we improved the morphological analysis performance for optical character recognizer (OCR) and social media data. Thus, for the new hybrid algorithm with LSTM, the accuracy reached 99.91% for the OCR dataset and 99.82% for social media data.

Key words: Algorithm, artificial neural network, rule-based system, morphology, hybrid model

1. Introduction

In natural language processing (NLP), morphological analysis of text documents is essential especially for agglutinative languages such as Finnish or Turkish. Because morphology is intertwined with other aspects of languages such as syntax and semantics, many natural language processing and understanding systems heavily depend on morphological analysis. Morphological analysis entails breaking down a word into the smallest possible pieces. For highly inflected languages such as Turkic languages, Hungarian, Finnish, or Korean, morphological analysis is of critical importance because their morphologies are highly intertwined with the grammar. Morphological analysis finds suffixes and their morphological tags, which include, tense, case, plural, and person [1]. Morphological analysis can be achieved in a supervised rule-based manner using finite-state-based morphological analyzers, semi- or unsupervised neural network-based approach using long short-term memory (LSTM). For the languages with rich morphology, the error rate is more complicated than the languages with poor morphology such as English. Shen et al. [2] proposed an LSTM-based approach for supervised morphological disambiguation models.

Computational morphological analyzers can be grouped into four main types: (1) rule-based, (2) statistical, (3) neural network-based, and (4) hybrid methods. The rule-based methods consist of grammatical rules of the respective languages and are built by experts such as linguists, while neural network-based methods

*Correspondence: ayla.kayabas@ahievran.edu.tr

use a huge amount of text sources, or corpora, to build such systems. In contrast, statistical methods utilize the statistical information of words or characters in a document or corpus. Statistical approaches have been applied in diverse NLP tasks very successfully, including morphological analysis, machine translation, and text classification [3]. With a hybrid model, different approaches are combined in a pipeline to increase the system's overall performance [4]. The combination of the rule-based morphological analyzers TRMOR and TRmorph [5, 6] with a neural network-based LSTM model accomplishes a more precise comparison of rule-based and neural network-based systems. Firstly, the LSTM model corrects the word errors in corpora obtained from OCR documents. The success of the model is later proved with social media data [7, 8]. We investigate the results through a comparison of the morphological analyzers and LSTM results, and we run the LSTM model iteratively to improve the performance.

The model basically has modules of rule-based morphological analyzers and an artificial neural network-based correction mechanism that unified the rule-based system and ANN-based system. Only the morphological analyzer was used to analyze corpus data in order to detect erroneous words. However, it was necessary to improve the model in order for it to have a higher detection ratio for errors. Thus, a new model was built by adapting an ANN-based system into the model that ran the ANN-based and rule-based modules separately and combined the results to improve the results. These components of the hybrid model will be explained further, and a brief description of the ANN-based and rule-based mechanisms will be provided and the hybrid architecture will be presented. This study is based on combining two of these approaches, namely:

- Rule-based approach: TRMOR and TRmorph
- Artificial neural network-based approach: LSTM

The system components comprise two pipelined executions: a rule-based and a neural network-based approach. The data is fed simultaneously into both systems. For the rule-based system, the data has to be prepared, whereas for the ANN-based system, preprocessing is not required. The model proceeded in the following steps: Firstly, a data preprocessing step is needed for the morphological analyzers, which comprise tokenization, lowercasing, and deleting the punctuation. After this, the word list is fed into the analyzers to generate the analysis. For the ANN-based model, the data is fed without any preprocessing. The output of the LSTM model constituted the corrected words, which may increase the performance value of the total corrected words. In order to evaluate the LSTM model, the training process is repeated using 10-fold cross-validation because the LSTM model chooses random initial weights in every running to determine the minimum, average, and maximum performance values of the model. In the next step, the common unanalyzed words from these analyzers and the corrected words are compared. In order to verify them, we check the correctness of the unanalyzed words. The implementation of the hybrid model has a threshold value, which makes it dynamic because the initial values of the variables depend on how the success point value was defined. In summary, our research accomplishes the following:

- Developments of the artificial Neural-Net-XMOR algorithm improved the morphological analysis performance.
- Improvements obtained for TRMOR and TRmorph using the neural network-based language modeling approach LSTM are observed.
- Performances of TRMOR and another state-of-the-art analyzer, TRmorph, are investigated.

The most obvious disadvantage of the rule-based approach is that it requires skilled experts: it takes a linguist or a knowledgeable engineer to manually encode each rule in NLP. Rules must be crafted manually and enhanced continually. Moreover, the complexity of the system can reach a point where some of the rules can begin to contradict each other. Overall, a rule-based system can successfully capture a particular language phenomenon, wherein it can perform linguistic relationship decoding between words and sentence interpretation. Therefore, it can easily conduct sentence-level tasks, including extraction and parsing. For this reason, rule-based approaches are generally better for query analysis.

The obvious advantage of machine learning is that the model can learn from data without manually defined rules and grammar coding. On the other hand, the rule-based approaches require expert skills. In general, ANN-based approaches can significantly simplify the development of several NLP tasks when good training datasets are available. However, it is often not so easy in practice. The advantage of the implemented approaches is that they are able to be applied to documents of other scripts easily and accurately.

ANN-based systems, especially LSTM models show better results when used for language-independent OCR [3]. The LSTM model utilizes optical character recognition (OCR), which provides better performance than rule-based approaches for error correction without using a language model. Several studies have been conducted on the correction of errors for OCR and normalization of the social media documents. Some of the applied methods use dictionary-based approaches while some use state-of-the-art neural network-based LSTM approaches. We will explain these components of our hybrid model further. We next provide a brief description of ANN-based and rule-based mechanisms and present a hybrid architecture.

The remainder of this paper is structured as follows: In Section 2, a brief review is presented of the different approaches and comparisons. Section 3 is related to the rule-based XMOR components, namely TRMOR and TRmorph. The next part of the hybrid model is the LSTM model as ANN-based components have been given in Section 4. Section 5 defines our artificial neural-Net-XMOR model and describes its architecture and algorithm. The experiments and the results are reported and discussed in Section 6. Finally, Section 7 concludes the paper.

2. Related work

Hybrid approaches to NLP were primarily used from the 1980s to 2010s. Various neural network models have been suggested among these, including a hybrid approach with a recurrent neural network (RNN) and a convolutional neural network (CNN) [9]. There are hybrid approaches implemented both on the same levels and on different levels that have been used to accomplish some broad tasks in NLP. These approaches are applied, for example, at neural network-based levels or rule-based levels.

The hybrid methods were adopted in many areas of NLP to improve performance and coverage of tasks, for part-of-speech (POS) tagging [10], and for hybrid speech recognition, with good results from the model [11] developed using neural network-based bidirectional LSTM (BiLSTM) and a statistical-based hidden Markov model (HMM). The well-studied currently published work by Özateş et al. [12] proposes a hybrid approach to dependency parsing for Turkish applying hand-crafted rules, which includes the morphological information combined with a deep learning-based dependency parsing method to improve the performance. Another hybrid approach was found to outperform machine learning-based and rule-based approaches for named entity recognition [4]. This combined a rule-based approach and a machine learning-based approach to recognize named entities in every direction, namely “Person,” “Location,” and “Organization.” Hybrid grammatical error correction [13] and a hybrid approach were applied for morphological disambiguation for Turkish by Kutlu and Cicekli [14], who also achieved good performance. That hybrid approach developed by Kutlu and Cicekli

for morphological parsing appears to be the first one for Turkish. Their hybrid morphological system applies a combination of statistical-based and rule-based approaches for disambiguation in morphological analysis. They used preannotated hand-tagged data with 10-fold cross-validation to evaluate their system. Yucebas and Tintin developed a new tool for Turkish language named GovdeTurk [15] improves the accuracy for stemming and labeling. Luong and Manning [16], meanwhile, developed a hybrid neural machine translation system to successfully complete English to Czech translation tasks. Oflazer's error-tolerant recognition mechanism corrects erroneous input word forms and analyzes them morphologically for Turkish as well as for many European languages [17].

The current work in [18] also used LSTM to predict the correct form or near representatives of erroneous words. Their model was based on word embeddings using a dictionary method by creating the dictionary on their own. The proposed method either corrected the incorrect words or made a replacement with similar words. The authors achieved an accuracy rate of 85.80% with the LSTM model using the dictionary method. In another study [19], the authors developed a model, combining a rule-based finite-state mechanism with a neural network-based mechanism, namely, BiLSTM, to disambiguate the morphological analysis. In [20], the authors proposed a method for correcting misspelled words using two-layer LSTM on Twitter data. Their proposed method yielded 5% absolute improvement over the state-of-the-art Turkish spelling correction systems in a test set that contained human-made misspellings from Twitter messages. To solve OCR-caused errors, in any texts, historical or nonhistorical, researchers have been training language-specific character recognition models. Drobac and Linden [21] developed a hybrid model of CNN-LSTM. The mixed model performed similarly to the Swedish monolingual model. Using the hybrid model, they achieved better results on Finnish data.

The correction accuracy improvement of the hybrid model herein, using rule-based and ANN-based stands, achieved maximum performance at 99.91%, whereas the study of Aydogan and Karci [18], which used a hybrid of LSTM and dictionary method, achieved an accuracy rate of 85.80% for the LSTM model using the dictionary method. Another study, by Buyuk and Arslan [20], developed a proposed method that yielded 5% improvement for Turkish social media messages.

3. TRMOR and TRmorph as rule-based XMOR components

Corpora are prepared for the morphological analyzers, namely with tokenization, lowercasing, and cleaning of the punctuation. Later, data is fed into the morphological analyzers. The TRMOR and TRmorph analyzers are explained in the following subsection.

Phonology and morphology are the main forms of finite-state mechanisms [22]. In this research, we use two different morphological analyzers for Turkish, namely TRMOR and the well-known TRmorph. TRMOR and TRmorph are rule-based morphological analyzers, similarly to other various morphological analyzers that use Stuttgart Finite-State Transducer (SFST)¹. SFST is a toolbox for the implementation of morphological analyzers and other tools that are based on finite-state transducers. The main forms of finite-state mechanisms are phonology and morphology. The SFST tool is noncommercial and can be freely used under the GNU Public License. The implementation of SFST tools consists of a lexicon wherein stem morphemes are given together with their part of speech and inflection classes, a set of regular expressions that specify the morphotactics, and a set of phonological rules that will transform the morphemes' canonical representations to surface forms. Due to the fact that agglutinative languages are extremely productive, it is infeasible to list a set of possible word

¹<http://www.cis.uni-muenchen.de/~schmid/tools/SFST/>

forms, as could be done for English; instead, analysis and testing are required [23]. We first compared analyzers using OCR data with 1031 words.

4. LSTM model as neural network-based component of the model

In order to train the model, two types of data are required: training and test data. The training data are clean data and were increased by five times for training the model. For the two different types of data, the model was executed using k-fold cross-validation. One of the important characteristics of the LSTM-based line recognizers is their ability to achieve excellent recognition accuracy without relying on sophisticated or uniquely specialized features or any postprocessing, such as language modeling, dictionary corrections, or other adaptations. As a result, the entire process is both easy and applicable to many different alphabets and languages [27].

Language models have critical importance in the pre- and postprocessing of OCR. LSTM models are applied for a wide range of tasks, from the recognition of text in images to model generation for language. The most popular of the LSTM models are the BiLSTM models, established on the premise that output at a certain moment may be dependent not only on the prior elements from the sequence but on the succeeding sequence, as well. BiLSTM learns error modeling from erroneous words that it encounters during training.

5. Artificial Neural-Net-XMOR algorithm

The model has modules of TRMOR, TRmorph, and LSTM, which unifies a rule-based system (the morphological analyzers) and neural network-based system. In our previous study [5], we utilized only TRMOR to analyze data to detect erroneous words. However, we need to improve our model to obtain a higher detection ratio, and so we have built a new model by adapting LSTM into it, which will now run LSTM and TRMOR separately and compare the results.

5.1. The architecture of the model

Hybrid systems mostly combine several different systems in a pipelined manner. Here, we have designed a new hybrid model that utilizes a rule-based approach together with a neural network-based approach in such a pipelined process. Both rule-based and neural network-based approaches have their own unique advantages and disadvantages [25]. For example, rule-based paradigms rely on laborious effort to create rules for the system. In addition, the training of the model plays an important role in artificial neural network (ANN)-based paradigms and it is very tedious work. Thus, in this article, we present a new hybrid architecture that is significantly more successful compared to either machine-learning or rule-based systems alone.

The system components comprise two pipelined executions: a rule-based and a neural network-based approach. The execution proceeds via three main phases: (1) comparison of the paradigms; (2) filtering, or the selection and extraction of analyzed and unanalyzed words; and (3) the hybrid phase, or correction of unanalyzed erroneous words. Figure 1 illustrates the architecture for hybrid error correction. In Figure 1, data is simultaneously fed into the morphological analyzers and the LSTM model.

1a. Rule-based component The morphological analyzer applies a rule-based approach that parses words into their smaller units, namely morphemes. The input for this module is text data (i.e. a corpus). We have utilized two different corpora here to quantify the robustness of the overall system. One corpus is obtained from OCR text data and the other from social media data. The output of the morphological analyzer is a list of parsed and unparsed words. These words are filtered from the same lists. The words that were successfully analyzed (A) were defined in the lexicon and processed no further. On the other hand, the unanalyzed words

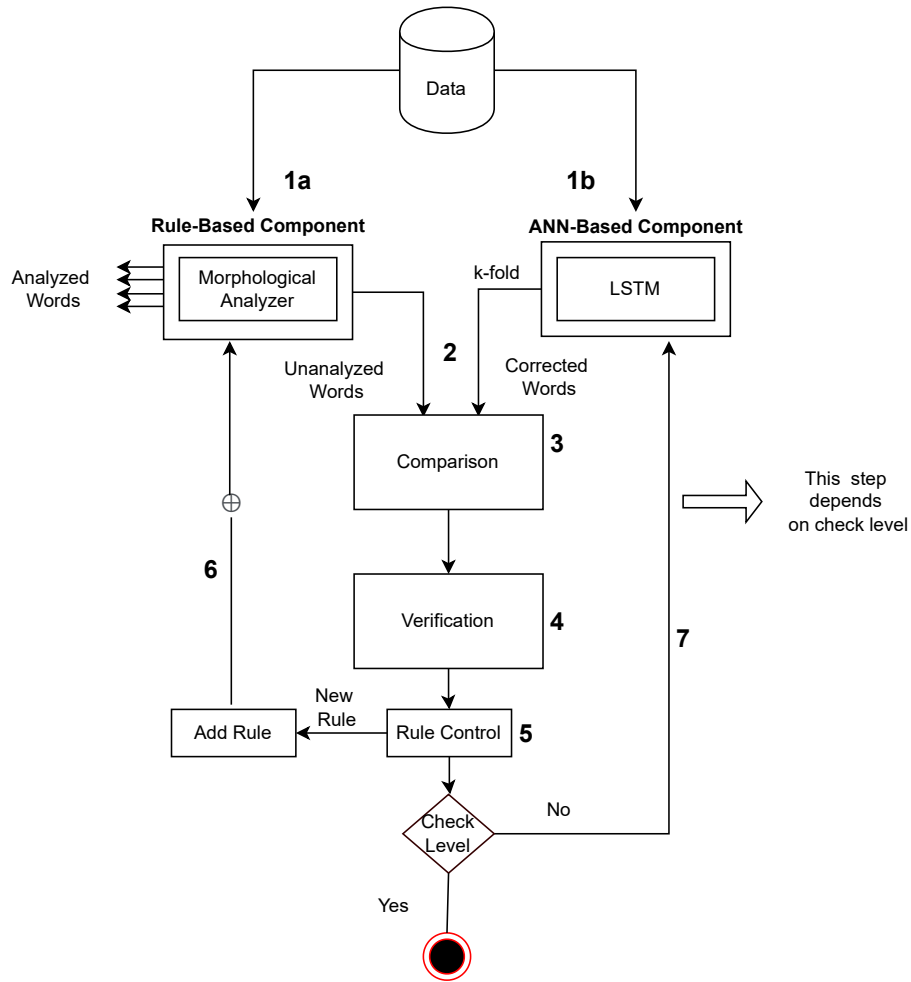


Figure 1. Illustration of overall hybrid model.

(Y) were marked for further analysis in the next stages. Before the morphological process, we tokenize the corpus, whereby words from a sentence are separated and constructed from line to line as a word list. In the next step, capital letters are changed into lowercase letters, and then all punctuation is deleted from the text. After these preprocessing steps, the word list is fed into the analyzers to generate analysis. This process occurs only one time for the module. Figure 2 illustrates the extraction of unanalyzed words from the morphological analyzers.

The main difference between TRMOR and TRmorph is the evaluation methodology. The evaluation of TRMOR was executed on gold-standard words [5]. Both mechanisms of SFST were used for the same data to gain more analyzed words. If no analysis is obtained for one word in one of the analyzers, it can be analyzed in the other one, since both systems use the same tool (SFST). TRmorph covers almost all word forms (including numbers and conjunctions), but there is no evaluation of the gold standard for TRmorph. TRMOR captures fewer word forms than TRmorph. The following are examples for the analyzers. As shown below, TRMOR provided one analysis, whereas TRmorph provided three analyses. TRMOR and TRmorph produces for the wordform *gelmelisiniz* ("you should come") following analysis.

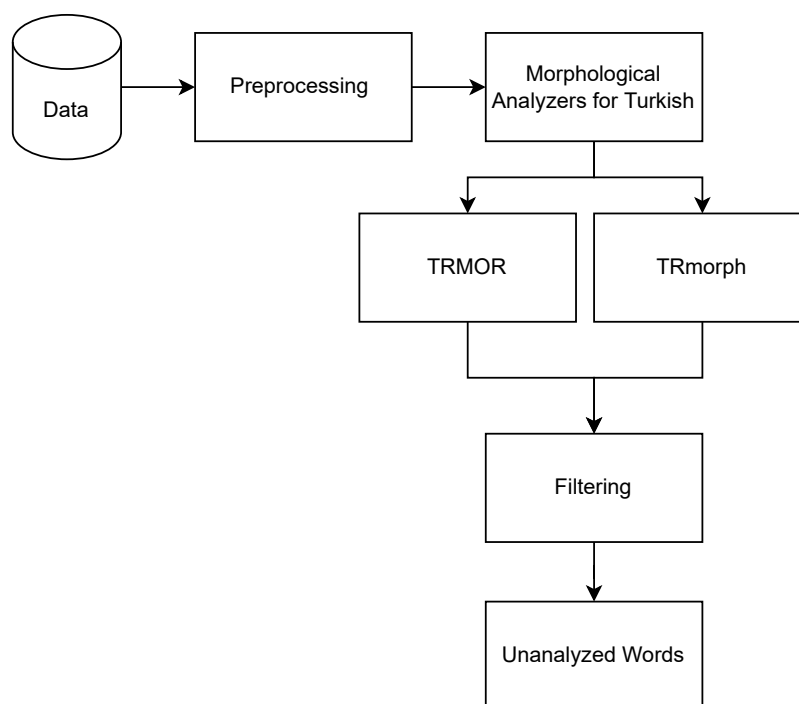


Figure 2. Extraction of nonhyphenated words after comparison of analyzers.

TRMOR analysis result:

- gel⟨V⟩⟨obligative⟩⟨2⟩⟨pl⟩

TRmorph analysis results:

- gel⟨v⟩⟨t_obl⟩⟨2p⟩
- gel⟨v⟩⟨D_mA⟩⟨n⟩⟨D_II⟩⟨n⟩⟨2p⟩
- gel ⟨v⟩⟨D_mA⟩⟨n⟩⟨D_II⟩⟨adj⟩⟨Djn_0⟩⟨n⟩⟨2p⟩

In Table 1, we demonstrated the analyzed/true and unanalyzed/false analysis of the system. There are two reasons that the morphological analyzers do return no results. Firstly, either the word is correct or noisy. It is possible that the word is correct but no result because it does not exist in the lexicon or secondly, the word is the noise and it must be corrected before fed into the systems.

For example, the word *sanatkâr* ('artist'), although the word is true it means there is no noise of the word, but the morphological analyzer does not yield any return because the word does not exist in the systems' lexicon.

1b. Neural network-based component The LSTM extends from the RNN, which is a type of ANN. The input data of this module are the same as those for the rule-based component, namely OCR data or social media data. The datasets have two parts that are used for testing and training the LSTM model. The quantity of training data is five times greater than that of the test data because the model learns from training data that are free of error. LSTM does not require preprocessing steps, contrary to morphological analyzers because

Table 1. Analyzed and unanalyzed strings of TRMOR.

Input string	Unanalyzed	Analyzed	Possible analysis
sanatkâr	yes	no	–
okulların	yes	no	–
ılımlı	yes	no	–
aşırılık	no	yes	aşır⟨V⟩⟨Y⟩⟨N⟩⟨SUFF⟩⟨D_IYk⟩⟨Nom⟩aşır⟨N⟩⟨D_IYk⟩⟨Nom⟩
hikmet	no	yes	hikmet⟨N⟩⟨Nom⟩hikmet⟨N⟩⟨sg⟩⟨PRO⟩⟨3⟩⟨sg⟩

such machine learning systems do these on their own, which is another advantage of the LSTM. We do not perform any preprocessing such as tokenization, lowercasing, or deleting of punctuation such as that done with the morphological analyzers.

The output of the LSTM model constitutes the corrected words, which may increase the performance value of the total corrected words. In the present work, we implemented character-based BiLSTM [25]. The performance calculation of the LSTM model is given in Section 6.1.2.

We use 10-fold cross-validation because whenever we run the LSTM model that chooses the random initial weights, it generates different corrected words depending on the performance value as shown in the third step of the algorithm by defining the k-fold value. In order to evaluate the LSTM model, we repeat the training process to determine the minimum, average, and maximum performance values of the model. The 10-fold LSTM model results from output text are saved for every fold running operation.

2. & 3. Comparison When we operated this module for the first time, the morphological analyzers explained in Module 1 and the LSTM modules were executed to compare the output of the results. In this step, the analyzers (TRMOR and TRmorph) generate two types of output: analyzed and unanalyzed words, according to their internal morphology structures. According to the algorithm, the number of analyzed words is represented as Y. If the total number of words is X, the unanalyzed word count is X - Y. To improve the performance value, we initially ran the LSTM model with 10-fold cross-validation. The common unanalyzed words from these analyzers and the corrected words were compared according to the lexicon of morphological systems or the training data of the LSTM model (Figure 2). Comparison results are presented here as corrected.

4. Verification The verification process is accomplished in a supervised manner; the unanalyzed words are checked for correctness. This step is repeated 10-fold for every analyzed word (Y).

5. and 6. Rule control and add rule In this section, we implement three primary operations:

(i) Rule control and add rule We add the new corrected word(s) to the successfully analyzed set. However, a word may be corrected without the analyzer's accomplishing any analysis because of the lack of a corresponding rule. For the corrected word(s), the rule control module controls the rule-set to make sure there is no redundancy rule. If there are new corrected words (L), then these are added to the total analyzed words (R). The steps for the LSTM and other modules are repeated as shown in steps 3, 4, 5, 7, and 8 until the success value is equal to the threshold value of the overall systems for the uncorrected words from the LSTM model and analyzers.

To correct the erroneous word, it is fed into the LSTM model. If the string is corrected using the LSTM model, analyzers return an output that indicates the word exists in the lexicon and the ruleset is already defined

for this word. Otherwise, analyzers do not return any output. There are two reasons for this outcome: a) The word does not exist in the lexicon, in which case the word is added into the lexicon or b) There is no such rule defined for a word which does not morphologically satisfy the requirement. In this case, we define a new rule relative to the suffix of the word in the ruleset. The new rule is defined as the following steps:

a) Lexicon: word stem and its related inflection class.

b) Inflection class: all possible inflectional surface variants for the word based are listed on the inflection class if verb, noun, or named entity, etc. The system first concatenates stem and suffix morphemes in all possible correct sequences via morphotactic rules.

c) Phonology: mapping the resulting string to the correct surface form via morphophonological rules.

(ii) Increasing the total number of analyzed words (R) The analyzed total value is increased based on newly analyzed words.

(iii) Check level Initially, k-fold iteration is completed to calculate the total analyzed words to reach a successful ending point. The results are combined using union operation in Operation A 3(b) in Section 5.1.1 as shown in the algorithm. Later, we operate the LSTM module. To eliminate the words that exist as an output of the corrected words from the LSTM module, they are compared with the analyzed words from the analyzers.

After the process of rule control, if there is no rule for some corrected words, rules will be added to the analyzer.

7. Check level The corrected words (Z) are combined to determine the total number of analyzed words. This operation is repeated until the defined success point value (DSPV) is achieved. In later operations, in order to reach the corrected success point value, the LSTM module is executed repeatedly. In this part, we also check the total number of iterations reached at the defined maximum iteration number, or the number of maximum operations (NUM_MO), in order to prevent infinite numbers of operations. This limit variable is also defined at the beginning of the algorithm.

The overall system is dynamic because the initial values of variables depend on how we define the success point value.

5.1.1. Algorithm

Initial Variables: The dynamic variable should be defined for calculations.

Definitions:

- RULE_BASED_MODULE: Rule-based morphological analyzer (XMOR)
- ANN_BASED_MODULE: Artificial neural network-based module
- X : Total words
- Y: Analyzed words from RULE_BASED_XMOR
- DSPV : Defined Success Point Value // success value defined by the use
- NUM_MO : Number of Maximum Operations // the overall system should not run infinitely. It defines the maximum operations.

- **CURRENT_ANN_BASED_MODULE_ITERATION_NUMBER**: Current Iteration Number =0; //initial value is zero.

Operation A: Run ANN_BASED_MODULE(X) //find the corrected word.

1. **Run** **RULE_BASED_XMOR(X)** // to find Y.
2. **Find** $Y = \text{Analyzed_RULE_BASED_XMOR}$ from step 1.
3. **Calculate** **ANN_BASED_MODULE** using k-fold value
 - (a) **Define** k, k-fold value, the minimum number of iterations ($k \geq 10$)
 - (b) **Repeat** Operation A k times to **compare** analyzed words and **find** new corrected words (z_i)
 $i=1,2,\dots, k$
 $Z = \{z_1 \cup z_2 \cup z_3 \cup \dots \cup z_k\}$
 $T = \text{Distance}(z_i, z_j)$ for $j = i+1,\dots, k$ // Levenshtein Distance
Until $i > k$
Calculate $\text{Analyzed_Total_Words}(R) = (Y \cup T)$
if ($\text{Number_of_R}(M) \geq \text{DSPV}$) **then**
 Terminate program
 - (c) **CURRENT_ANN_BASED_MODULE_ITERATION_NUMBER** = k;
4. **Call** [Operation A] to get L value // L (new corrected words) is the return value of Operation A
 - **Calculate** $R = R \cup L$
CURRENT_ANN_BASED_MODULE_ITERATION_NUMBER += 1;

if ($\text{Number_of_R}(M) \geq \text{DSPV}$) **then**

Terminate program

else

if (**CURRENT_ANN_BASED_MODULE_ITERATION_NUMBER** < **NUM_MO**) **then**

 Go again to step 4

else

Terminate program

6. Experiments

We tested our novel algorithm for Turkish via TRMOR and TRmorph and compared these to obtain the input data for the neural network-based approach, namely the LSTM model. The experiments were divided into three groups for the hybrid system. The reference dataset was fed into the rule-based component to ensure that the output would represent annotated datasets.

The performance of the rule-based approaches was calculated for TRMOR and TRmorph separately, while 10-fold cross-validation was chosen for the neural network-based approach to measure the approximate average model performance.

6.1. Datasets

Textual resources such as books, newspapers, and many other printed texts can be valuable in NLP studies. The data that are collected from OCR documents and social media extraction are generally examined for two main aims: the detection of erroneous words and the correction of words in various predefined forms. The corpus used in this study relies on OCR data, which plays an important role in language models because large amounts of data are the basis for every work, whether rule-based, statistical, neural network-based, or hybrid. We constructed our own corpus and training data for the model. The dataset is taken from OCR data. Nonword errors are produced through OCR. They are a type of error that does not occur in the dictionary because it is meaningless. However, real words are words that occur in the dictionary. Table 2 shows OCR data samples from corpus. We used social media data [7, 8] in our experiment for rule-based and neural-based approaches. Furthermore, LSTM models are trained from small corpora to large corpora to observe the model performances.

Table 2. OCR data samples.

Real word	Non-word	English
anlayabilecekleri	anlaya- bñecekleri	those who can understand
“algımn ötesinde”dir	“algýnýn ötesinde’üür	beyond perception
edindiğimizdir	edindi ðimizdýr	is what we got

The document files were prepared after conducting the OCR of the text, it was saved in plain text format and since there are some special characters in the Turkish language, the encoding of the collected data was changed based on the Turkish encoding. These were ISO 8859-3, ISO 8859-9, OEM 857, and Windows-1254. Since the model was used on both UNIX and Windows systems, UTF-8 encoding was used for UNIX and ISO 8859-9 encoding was used for Windows.

In order to train the ANN-based model, we divided the datasets into two types: training and test data. The training data consists of clean data. The test data is gained after the text scanned. The table above includes samples of test data.

To obtain a model using LSTM, a small corpus is used to perform corrections at the character level. However, better results can be obtained with more training data because the model will learn relations between characters and accordingly be able to form words [26]. To train character-level LSTM, large amounts of data are used to learn how to generate text character by character [27]. However, it is also argued that higher volumes of training data do not necessarily lead to significant improvement [3], like in our case.

6.2. Results

For the evaluation of the model, we used k-fold cross-validation, with results given in tables below. The experimental results obtained here reveal that adapting the hybrid approach generates the best performance. The performance measure of the OCR data and social media data of the model is illustrated in more detail below. Other reclamation statistics are shown below in the tables. We conclude that if we use both LSTM and rule-based systems together, as the results show below for 10-fold experiments, we will have better performance results. Table 3 illustrates how many times the model responded to the errors in the test data in every test phase on average. The overall analysis was based on OCR errors for 10-fold testing of the model.

The 10-fold test method was used to obtain the average model performances and 1142 words from

Table 3. Test result of OCR data using ANN-based module in the algorithm.

	Corrected words	Noncorrected words
Average (%)	62.62	67.30

social media data were used for testing and training. Among 1031 OCR words, there are 158 words in TRMOR and 30 words in TRmorph that generated no results. Table 4 shows the distribution of data using TRMOR and TRmorph analyzing tools. The table indicates that no results percentages of OCR data using both TRMOR and TRmorph is higher than the social media data.

Table 4. Distribution of no result on different data types (unanalyzed words).

Analyzer tools	OCR data		Social media data	
	# of words	No result (%)	# of words	No result (%)
TRMOR	1031	15.32	1142	28.63
TRmorph	1031	2.91	1142	6.57

In Table 5, we show the distribution of analyzed words from both analyzers (TRMOR and TRmorph) for the OCR and social media data. They have different error types. As the table below shows, OCR data appears to have higher accuracy than social media data because it is printed on paper. Later, the words without generated results are fed into the LSTM model to see how the model is handled.

Table 5. The accuracy of rule-based models.

	TRMOR	TRmorph	Both(common words)
Social media (%)	71.37	93.43	93.78
OCR (%)	84.67	97.09	98.06

Table 6 illustrates the performance analysis of hybrid model (TRMOR, TRmorph, and LSTM) for social media and the OCR data. It indicates that the current hybrid method combines the rule-based morphological analyzers and the long short-term memory-based techniques to improve the morphological analysis performance for OCR data to 99.03% and social media data to 97.37% on average, which constitutes the-state-of-the-art morphological analysis for Turkish to the best of authors' knowledge. Since the average and maximum performance were focused on, minimum values were not indicated. Moreover, the maximum performance stands at 99.91% for OCR data and 99.82% for social media data. We observed that the performance increased by 0.97% for OCR data and by 3.59% for social media data when we use an average of k-fold corrected values. When the hybrid algorithm was utilized with the LSTM, the accuracy ranged from 93.78% (Table 5) to 99.82% for the same dataset. For the OCR data with the same method (hybrid and LSTM), the correction accuracy was improved, ranging from 98.06% to 99.91%. It was observed that the performance increased by 1.85% for the OCR data and by 6.04% for the social media data when a maximum of k-fold corrected values was used. This finding suggests that LSTM has a better correction rate when applied to social media data compared to OCR data.

Table 6. Results of hybrid models.

	Average	Maximum
Social media (%)	97.37	99.82
OCR (%)	99.03	99.91

7. Conclusion

A novel algorithm, the artificial Neural-Net-XMOR, was developed, which uses the rule-based approach, morphological analyzers, and an ANN-based model. In this algorithm, a threshold value was defined that controls the number of times the algorithm may run the k-fold iterations and the number of maximum operations to limit the ANN iterations. These were compared and verified in a supervised manner. The results were combined using union operations described in the artificial Neural-Net-XMOR algorithm. The hybrid model implemented herein was compared with a model combining the neural network-based, a language-independent character-based encoder-decoder architecture for correcting noisy data, and rule-based approaches. The combined model exhibited higher performance than the single-rule-based XMOR mechanism. The ANN system, which was trained on noisy data, was able to correct many of the errors. The results showed that, for the OCR data, the normalization approach more effectively corrected the errors as a result of the OCR data with more dependence on contextualized text than the social media data when solving real-word errors. Performance increments were achieved while using this hybrid model.

In summary, the current research accomplished the following: A hybrid algorithm was experimented with by applying rule-based and ANN-based techniques and the results showed that the morphological analysis performance for the OCR and social media data were improved. When this hybrid algorithm was used with the ANN, the accuracy achieved was 99.82% for the social media dataset and 99.91% for the OCR data. It was observed that the performance increased by 1.85% for the OCR data and by 6.04% for the social media data when a maximum of k-fold corrected values was used. Rule-based natural language generation systems have been quite successful, but they suffer from some limitations. They require extensive human effort. Thus, by using the hybrid model proposed herein, which is one of the ANN models for Turkish, new rules were added into the rule-based systems and the morphological analyzers were improved.

In future work, ANN-based language modeling will be used for disambiguation resolution, which is a big and challenging task for NLP, because these systems are able to preserve contextual neighbors over long distances while running forward and backward. Since the model learns to correct the errors in the ground truth by incorporating context-aware processing, it will be possible to correct potential real-word and nonword errors by applying the model to large datasets from OCR text. Finally, it is aimed to apply this new model to other languages, such as German, English, or Latin, which also use the SFST system for morphological analysis.

References

- [1] Can B. Unsupervised learning of allomorphs in Turkish. *Turkish Journal of Electrical Engineering & Computer Sciences* 2017; 25 (4): 3253-3260. doi:10.3906/elk-1605-216.
- [2] Shen Q, Clothiaux D, Tagtow E, Littell P, Dyer C. The role of context in neural morphological disambiguation. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* 2016; 181-191.

- [3] Ul-Hasan A. Generic text recognition using long short-term memory networks PhD, University of Kaiserslautern, Kaiserslautern, Germany, 2016.
- [4] Oudah M, Shaalan K. A pipeline Arabic named entity recognition using a hybrid approach. *Proceedings of COLING 2012*; 2159–2176.
- [5] Kayabaş A, Schmid H, Topcu AE, Kılıç Ö. TRMOR: a finite-state-based morphological analyzer for Turkish. *Turkish Journal of Electrical Engineering & Computer Sciences* 2019; 27 (5): 3837–3851.
- [6] Çöltekin Ç. A freely available morphological analyzer for Turkish. In: *7th International Conference on Language Resources and Evaluation*; Valetta, Malta 2010. pp. 820-827.
- [7] Eryiğit G, Totunoğlu-Selamet D. Social media text normalization for Turkish. *Natural Language Engineering*, Cambridge University Press 2017; 23 (6): 835.
- [8] Pamay T, Sulubacak U, Torunoğlu-Selamet D, Eryiğit G. The annotation process of the ITU Web Treebank. *Proceedings of the 9th Linguistic Annotation Workshop* 2015; 95–101.
- [9] Kalchbrenner N, Blunsom P. Recurrent continuous translation models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* 2013; 1700–1709.
- [10] Schmid, Helmut. Part-of-speech tagging with neural networks. arXiv preprint [cmp-lg/9410018](https://arxiv.org/abs/cmp-lg/9410018) 1994.
- [11] Graves A, Jaitly N, Mohamed A. Hybrid speech recognition with deep bidirectional LSTM. *2013 IEEE workshop on automatic speech recognition and understanding* 2013; 273–278.
- [12] Özateş Ş, Özgür A, Güngör T, Öztürk B. A Hybrid Approach to Dependency Parsing: Combining Rules and Morphology with Deep Learning 2020. arXiv preprint [arXiv:2002.10116](https://arxiv.org/abs/2002.10116).
- [13] Ji J, Wang Q, Toutanova K, Gong Y, Truong S et al. A nested attention neural hybrid model for grammatical error correction. arXiv preprint [arXiv:1707.02026](https://arxiv.org/abs/1707.02026) 2017.
- [14] Kutlu M, Cicekli I. A hybrid morphological disambiguation system for turkish. *Proceedings of the Sixth International Joint Conference on Natural Language Processing* 2013; 1230–1236.
- [15] Yucebas S, Tintin R. GovdeTurk: A Novel Turkish Natural Language Processing Tool for Stemming, Morphological Labelling and Verb Negation 2021, *methods*: 10.15: 25.
- [16] Luong MT, Manning CD. Achieving open vocabulary neural machine translation with hybrid word-character models. arXiv preprint [arXiv:1604.00788](https://arxiv.org/abs/1604.00788) 2016.
- [17] Oflazer K. Error-tolerant finite state recognition with applications to morphological analysis and spelling correction. arXiv preprint [cmp-lg/9504031](https://arxiv.org/abs/cmp-lg/9504031) 1995.
- [18] Aydoğan M, Karci A. Spelling Correction with the Dictionary Method for the Turkish Language Using Word Embeddings. *Avrupa Bilim ve Teknoloji Dergisi* 2020; 57–63.
- [19] Hämäläinen, M, Wiecheteck L. Morphological Disambiguation of South Sámi with FSTs and Neural Networks. arXiv preprint [arXiv:2004.14062](https://arxiv.org/abs/2004.14062) 2020.
- [20] Büyük O, Arslan LM. Learning from mistakes: Improving spelling correction performance with automatic generation of realistic misspellings. *Expert Systems: Wiley Online Library* 2021; e12692.
- [21] Drobac S, Lindén K. Optical character recognition with neural networks and post-correction with finite state methods. *International Journal on Document Analysis and Recognition (IJDAR)*: Springer 2020; 23, 4, 279–295.
- [22] Karttunen L. Finite-state constraints. In: Goldsmith J (editor). *The Last Phonological Rule 6*. Chicago and London: The University of Chicago Press, 1993, pp. 173-194.
- [23] Schmid H, Fitschen A, Heid U. SMOR: A German Computational Morphology Covering Derivation, Composition and Inflection. In: *4th International Conference on Language Resources and Evaluation*; 26-27-28 May 2004; Lisbon, Portugal. 2004. pp. 1263-1266.

- [24] Ul-Hasan A, Breuel Thomas M. Can we build language-independent OCR using LSTM networks? Proceedings of the 4th International Workshop on Multilingual OCR 2013; 1–5.
- [25] D'hondt E, Grouin C, Grau B. Generating a training corpus for OCR post-correction using encoder-decoder model. Proceedings of the Eighth International Joint Conference on Natural Language Processing 2017; 1006–1014.
- [26] Mokhtar K, Bukhari Syed S, Dengel A. OCR Error Correction: State-of-the-Art vs an NMT-based Approach. 13th IAPR International Workshop on Document Analysis Systems (DAS) 2018; 429–434.
- [27] Karpathy A. The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog 2015; 23.