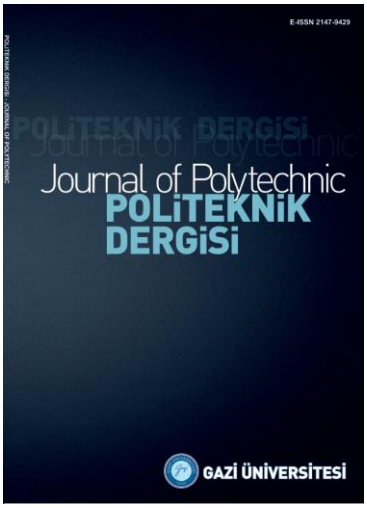

POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

A language model optimization method for Turkish automatic speech recognition system

Yazar(lar) (Author(s)): *Saadin OYUCU[1],Hüseyin POLAT[2]*

ORCID[1]: 0000-0003-3880-3039

ORCID[2]: 0000-0003-4128-2625

To cite to this article: Oyucu S., Polat H., "A language model optimization method for Turkish automatic speech recognition system", ***Journal of Polytechnic***, 26(3): 1167-1178, (2023).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Oyucu S., Polat H., "A language model optimization method for Turkish automatic speech recognition system", ***Politeknik Dergisi***, 26(3): 1167-1178, (2023).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: **10.2339/politeknik.1085512**

A Language Model Optimization Method for Turkish Automatic Speech Recognition System

Önemli noktalar (Highlights)

- *Automatic Speech Recognition*
- *Language Model*
- *Language Model Score Optimization*
- *Turkish Speech Corpus*
- *Tuskish Language Model*

Graphical Abstract

It was observed that when the corpus capacity increased, the proposed approach gave more precise results. The results obtained by improving the LM are visualized in Figure.

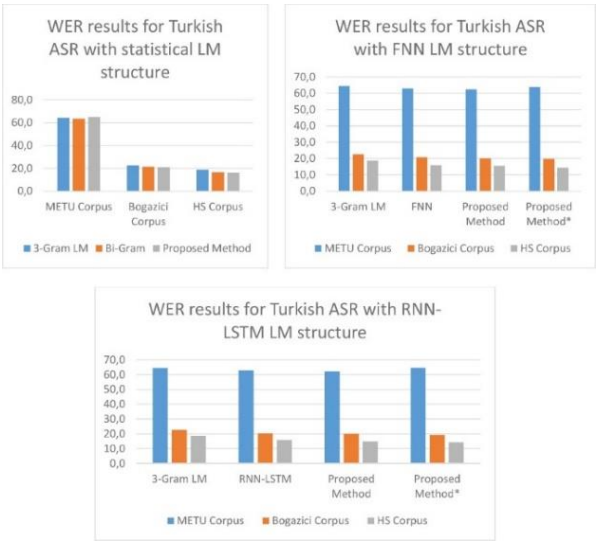

Figure. LM optimization results.

Aim

In this study an LM optimization has been performed which can model long dependencies and provide better results for AM output.

Design & Methodology

In the proposed method, instead of a fixed word sequence obtained from the Markov assumptions, the probability of the word sequence forming a sentence was calculated.

Originality

The proposed method has been tested on both statistical and Artificial Neural Network (ANN) based LMs.

Findings

According to the experimental results obtained from statistical-based LM, 0.5% WER increases for the METU corpus, 1.6% WER decreases for the Bogazici corpus, and a 2.5% WER decrease for the HS corpus were observed. In the Feedforward Neural Networks (FNN) based LM, WER decreases were observed 0.2% for the METU corpus, 0.8% for the Bogazici corpus, and 1.6% for the HS corpus.

Conclusion

As a result, when the proposed method was applied to the LMs required for ASR, WER decreased, and the total performance of ASR increased.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission

A Language Model Optimization Method for Turkish Automatic Speech Recognition System

Research Article

Saadin OYUCU[*], Hüseyin POLAT[2]

[1]Adıyaman University, Faculty of Engineering, Department of Computer Engineering, Adıayaman / Turkey
[2]Gazi University, Faculty of Technology, Department of Computer Engineering, Ankara / Turkey

(Geliş/Received : 10.03 2022 ; Kabul/Accepted : 06.07.2022 ; Erken Görünüm/Early View : 24.08.2022)

ABSTRACT

The current Automatic Speech Recognition (ASR) modeling strategy still suffers from huge performance degradation when faced with languages with limited resources such as Turkish. Especially when the Language Model (LM) does not support the Acoustic Model (AM) sufficiently, the Word Error Rate (WER) increases. Therefore, a robust LM makes a strong contribution to improving ASR performance by generating word relations from the existing corpus. However, developing a robust language model is a challenging task due to the agglutinative nature of Turkish. Therefore, within the scope of the study, a sentence-level LM optimization method is proposed to improve the WER performance of Turkish ASR. In the proposed method, instead of a fixed word sequence obtained from the Markov assumptions, the probability of the word sequence forming a sentence was calculated. A method with n-gram and skip-gram properties is presented to obtain the word sequence probability. The proposed method has been tested on both statistical and Artificial Neural Network (ANN) based LMs. In the experiments carried out using, not only words but also sub-word level, two Turkish corpora (METU and Bogazici) shared via Linguistic Data Consortium (LDC) and a separate corpus, which we separate corpus that we specially created as HS was used. According to the experimental results obtained from statistical-based LM, 0.5% WER increases for the METU corpus, 1.6% WER decreases for the Bogazici corpus, and a 2.5% WER decrease for the HS corpus were observed. In the Feedforward Neural Networks (FNN) based LM, WER decreases were observed 0.2% for the METU corpus, 0.8% for the Bogazici corpus, and 1.6% for the HS corpus. Also, in the Recurrent Neural Network (RNN)-Long Short Term Memory (LSTM) based LM, WER decreases were observed 0.6% for METU corpus, 1.1% for the Bogazici corpus and 1.5% for the HS corpus. As a result, when the proposed method was applied to the LMs required for ASR, WER decreased, and the total performance of ASR increased.

Keywords: Turkish Automatic speech recognition, Turkish language model, Turkish language model score optimization, Turkish corpus.

1. INTRODUCTION

People want to communicate with machinery or electronic devices by using their mother tongues. To fulfill this request, speech must be recognized by machinery or electronic devices. Automatic Speech Recognition (ASR) systems are being developed for this purpose, and their areas of usage are expanding every day. ASR is a technology that converts words spoken by people into computer-readable text [1]. ASR systems are frequently used to manage devices in automobiles [2], to control frequently used applications such as media players [3], and various applications in embedded systems [4] with speech. In a classical ASR architecture, there are essential components, such as Speech Processing, Decoder, Acoustic Model (AM), and Language Model (LM) (Figure 1). Although the combined use of these components increases the complexity of ASR systems, it has a significant impact on Word Error Rate (WER) [5].

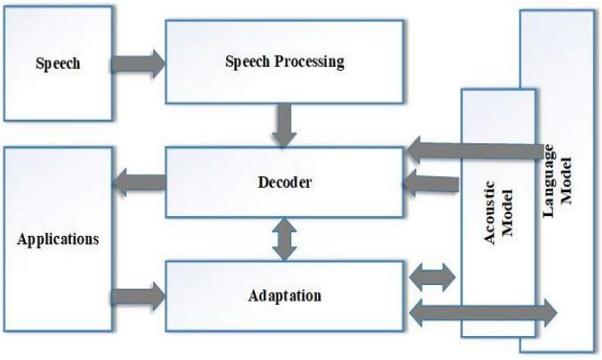

Fig. 1. The general architecture of the ASR system

During the Speech Processing phase, which is the first entry point of the ASR system, feature extraction is performed from the audio signal [6]. Feature extraction is performed to distinguish speech from other speeches and speakers. Due to the nature of the act of speech, each speech has distinctive individual characteristics integrated into the speech information [7]. Different feature extraction techniques help to obtain these unique

Sorumlu yazar/Corresponding Author
e-posta : saadinoyucu @adiyaman.edu.tr

features. For example, Mel Frequency Cepstral Coefficient (MFCC) is a feature extraction technique commonly used in speech recognition systems [8]. The frequency bands are logarithmically located in the MFCC. The method of MFCC calculation is based on short-term analysis. To extract the cepstral coefficients, the speech sample is taken as input and a Hamming window is applied to minimize the discontinuity of the signal. Another method used as an alternative to MFCC is Linear Predictive Coding (LPC). LPC is used to estimate the basic parameters of speech [9]. The main idea behind LPC is that a speech feature can be predicted as a linear combination of past speech examples [10].

Decoder, one of the other components of ASR, converts the feature vectors obtained by using AM and LM into phoneme sequences. The AM and LM components are each models produced with different training data. The AM component provides detailed information on the acoustic environment, phonetics, microphone and environmental variability, gender, and dialectical differences among speakers [11]. In acoustic modeling, firstly, the posterior probability of the phoneme within a given time signal is calculated. The phoneme order can be matched with the Hidden Markov Model (HMM) Triphone structure one-to-one. In general, the alignment of the phonemes is obtained by using the standard Gaussian distribution [12].

The speech signal given as input to the ASR system generates an output using AM. AM can also be developed with an Artificial Neural Network (ANN) instead of HMM [13]. In the ANN-based AM, the posterior probability of phonemes is independent for each window. This independence means that the phonemes in a word are separate from each other. The probability of observation of each HMM state is calculated by using Gaussian mixtures. The information produced at the AM output is phoneme sequences. The resulting phonemes are converted into word form by a dictionary. The resulting words are sent to an LM for evaluation [14].

The LM provides information on which words in a phoneme sequence form a possible word sequence, and which words will co-usability [15]. In the process of evaluating the AM output with the LM, a search is performed on the LM [16]. The complexity of the search algorithm to be used in ASR systems is directly related to the constraints imposed by the LM and the specified search area. The effect of different language models, including finite status information and n-grams, directly affects the deciphering result.

The success of HMM-based speech recognition systems is more dependent on LM than AM. In ANN-based ASR systems, some of the load on the LM is transferred to AM. However, a robust LM still affects ASR performance directly. Therefore, to obtain an ASR system with a low WER, the development of a robust language model is necessary.

When the literature is examined, it has been seen that many studies have been done on ASR for different languages. However, studies related to Turkish could not be carried forward due to low resources. Most of the studies on Turkish are focused on low resource problems and the overall success of the system was aimed to be increased [17–20]. However, most studies on LM have focused on the effect of the LM dimension on ASR [21–25]. The results show the necessity of large-scale LMs specifically for agglutinative languages such as Turkish. However, detecting different probabilities through LMs presents numerous challenges. Corpus deficiency, agglutinative structure, and irregular placement of words are some of these challenges for Turkish.

It has been seen in the literature that different techniques are used to determine the probabilities in LM. For LM, the n-gram based modeling method is generally has been used [26]. Markov assumptions are used in n-gram based LMs. In general, the probability is calculated by taking the past 2 to 4 word orders into consideration. However, the n-gram based LM cannot model clearly the word order in long sentences. This is because the limited past is reviewed through n-grams. ANN-based LMs have been developed to address this problem [26]. Since ANN-based language models do not use Markov assumptions, long dependencies in words can be modeled [27–29]. ANN-based LMs generally perform better than n-gram based LMs [30]. The major advantage of ANN LMs is that they predict word sequence probabilities in a continuous space. To improve the performance of ANN-based LMs, specific improvement techniques such as Dropout [31], Bayesian performance improvement [32], and bidirectional neural network [33] have been proposed. All these techniques have provided limited performance improvements in Turkish ASR applications [33]. The main reason for this is the agglutinative language structure of the Turkish Language [34]. With an agglutinative language structure, many long sentences can be formed. Analyzing that many words in retrospect lead to computational complexity. Thus, the offered methods or approaches must necessarily pay attention to two issues. The first is to benefit from a lot more words used in the past. The second is to reduce the computational complexity.

In this study, a method has been proposed to improve the required for Turkish ASR systems LM's N-best lists. In the proposed method, skip-gram and n-gram properties were used together. Statistical and ANN-based LMs have been created to test the success of the proposed method. Also, both wordlevel and subword-level LMs were used in the experiments. The experiments on created LMs were done on two Turkish corpora (METU and Bogazici) shared via Linguistic Data Consortium (LDC) as well as a separate corpus (the HS corpus) specifically prepared for this study. The METU and Bogazici corpora are inadequate for such a rich and agglutinative language, such as Turkish. Therefore, a unique corpus was prepared that could be sufficient to optimize the recognition errors at the Turkish ASR output. The effect of corpus size has been clearly demonstrated using different corpora. LM was developed with statistical and ANN-based

approaches and thereby the effects of different methods were demonstrated. As a result, when the proposed method was applied to LMs, WER decreased, and the total performance of ASR increased.

In Section 2 of this study, the working method of the LM for ASR structure is explained. The structure of the proposed model and the working process is described in detail in Section 3. The development of the Turkish ASR system and the experiments performed to observe the contribution of the proposed method are explained in detail in Section 4. The evaluation of the results, problems encountered, and the studies that can be carried out to solve these issues are presented in Section 5.

2. LANGUAGE MODELLING FOR ASR

LM is one of the most critical components of an ASR system. There is a deep relationship between AM and LM. The task of AM and LM used in ASR systems is described in detail in Equation 1 [35].

Equation 1 gives the corresponding word sequence in the speech recognition system for the acoustic phoneme or feature vector sequence $X = X_1, X_2, \dots, X_n$. The maximum posterior probability expressed in Equation 1 has the value $P(W|X).P(W)$ and $P(X|W)$ components are comprised of the probabilistic quantities calculated by language modeling and acoustic modeling. Since there is a wide range of words in ASR systems with an extensive vocabulary, it is necessary to break one word into a word sequence [36]. Their LMs use a statistical-based approach to predict the formation of a word sequence.

$$\mathrm{W} = \arg\max \mathrm{P}(\mathrm{W} \mid \mathrm{X}) = \arg\max \mathrm{P}(\mathrm{W})\mathrm{P}(\mathrm{X}|\mathrm{W}) \tag{1}$$

2.1. Statistical-Based Language Modeling

The task of LMs is to find the probabilistic sequence of the words by means of the statistical information obtained from the training corpus ($w_1^n = w_1 \dots w_n$). The conditional probability of each word is statistically calculated as in Equation 2 [37].

$$\mathrm{P}(w_1^n) = \mathrm{P}(w_1).\mathrm{P}(w_2|w_1) \dots \mathrm{P}(w_n|w_1^{n-1}) = \prod_{i=1}^{n} P(w_i|w_1^{i-1}) \tag{2}$$

Conditional probabilities do not produce reliable results in cases requiring long dependencies. Therefore, the real probability of a w_1^{i-1} sequence of w_1^{i-1} words w_{i+n-1}^{i-1} is approached depending on the definition of its expression instead of its sequence. In the statistical language model, words are dependent on the past word group. The number of words in the past that are taken into consideration is expressed by the n-gram language model. N-gram probabilities calculate a certain n-gram formation in the text corpus. The resulting value is $n-1$ estimated by dividing that value by the number of all n-grams starting with the same sequence of words. This operation is shown in Equation 3.

$$P(w_n|w_{n-1}, w_{n-2}, \dots w_{n-N+1}) = \frac{C(w_{n-N+1}, \dots w_{n-1}, w_n)}{\sum_{w_i} C(w_{n-N+1}, \dots w_{n-1}, w_i)} \tag{3}$$

The main problem in n-gram language modeling is that ASR performance decreases when the corpus size is small. If the training corpus is small, minimal probabilities can be assigned to a lot of the word sequences. In the decoding stage, minimal probabilities are often artificially increased to increase success.

Each language has a nature and structure of its own. Also, each language has a productivity structure different from others. Therefore, separate LMs should be created for each different language. However, on LM required for ASR, studies have been mostly conducted on English LMs. It is possible in theory and practice to form a dictionary of the morphological forms of the words in English. However, this process is almost impossible for agglutinative languages such as Turkish [38,39]. The increase in the number of words resulting from the morphological productivity of the language directly affects the language model. In this case, the success rate of ASR will decrease when statistical LM is developed with a low resource corpus.

In the literature, the use of skip-gram has been suggested to overcome the disadvantage of corpus deficiency [40]. The skip-gram model presented by Mikolov et al. aims to find the best word representations in predicting the words surrounding a target word. The process used to achieve this goal is demonstrated as in Equation 4.

$$\frac{1}{T}\sum_{t=1}^{T}\Big(\sum_{-c\le j\le c, j\neq 0} \log P\left(w_t + j|w_t\right)\Big) \tag{4}$$

$w_1, w_2, \ldots, w_t$ in Equation 4 refers to the words in the training corpus, and c refers to the size of the frame around the target word. With the representation of w_t, the set of content words to be estimated [40] is calculated. However, n-gram and skip-gram-based LMs cannot model the order of words in long sentences. Because by n-grams, a limited history is observed whereas by skip-grams, the word coming after a certain skip value is observed. Therefore, ANN-based LMs have been developed to model the sequence of words in long sentences [26].

2.2. Neural Network-Based Language Modeling

Since ANN-based language models use continuous-space representation, long dependencies in words can be modeled. ANN general structure is composed of 3 layers. These layers are the input layer, the hidden layer(s), and the output layer. The hidden layer lies between the input layer and the output layer. An ANN may have one or several hidden layers of neurons. Except for input nodes, neurons in each layer use a nonlinear activation function. Multiple layers and nonlinear activation distinguish multilayer ANN from a single layer ANN. Multilayer ANN can identify data that is not linearly separable.

The proposed method was applied to different ANN models. It was first used to Feedforward Neural Networks (FNN) architecture. In the FNN, all layers are connected, and the output of the artificial neuron in one layer becomes the input unit of a neuron in the next layer. The input layer communicates with the external environment, giving the sample to the neural network. The data is sent to hidden layers from the input layer, and after applying operations on the data in the hidden layer, they are sent to the output layer. FNN may not only have one output neuron but more than one, too. The number of neurons in the output layer should be directly related to the type of neural network [41]. Figure 2 shows an FNN structure.

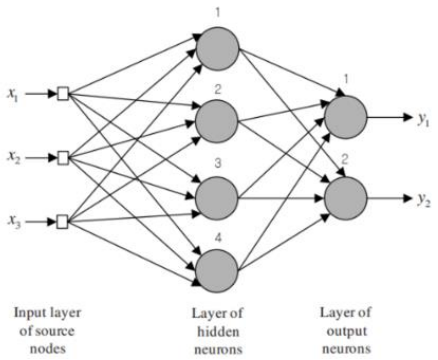

Fig. 2 General FNN structure [42]

To use the FNN for prediction purposes, the system must be trained. FNN generally utilizes a supervised learning technique called backpropagation for training. The training of the FNN is to determine the values of the connection weights between the layers to make the best prediction. Initially, the weight values are assigned randomly. The FNN then changes the weight values according to the learning algorithm used. Learning is a two-step process. At the first stage, the output produced by the neural network is determined for the sample shown. In the second stage, this output value is changed according to the weight of the connections. When the network reaches the correct weight values, it allows the samples to generalize about the event they represent.

The proposed method has also been tested on a different ANN structure, the Recurrent Neural Network (RNN). Recently, RNN has produced successful results in many fields of application, such as speech recognition, language modeling, and image analysis [43]. The main idea in RNN is that previous information can be used at the next steps of the network. This structure is generally applied to problems that can be solved dependent on the past. Sometimes we might need the information of only a few steps back in an input series. For example, for a language model to predict the missing word in the sentence “Bugün kar yağışı var ve - çok soğuk,” it is necessarily necessary to look at a few words in the past. In this case, a statistical language model construct can be used. However, in the sentence, “ben Türkiye’de yaşıyorum ve yıllardır ... Türkçe’yi de iyi konuşabiliyorum,” for the model to predict the word “Türkçe’yi,” the word “Türkiye’de,” which comes 10-15 words before it, must be remembered by the model. In such uses, the model must have information on a long history and not forget this information. In this case, as RNNs can model longer past word dependencies can be preferred. However, as the distance between the first input and the last input increases, RNNs begin to forget the information they have seen in the past. Theoretically, RNNs should remember such long dependencies. In actual use, however, this is not always true, and the model tends to become forgetful at long inputs. Long Short Term Memory (LSTM) units have been proposed to solve this problem [41]. LSTM is used to remember long-term dependencies and thus obtain context-aware neural networks (Figure 3).

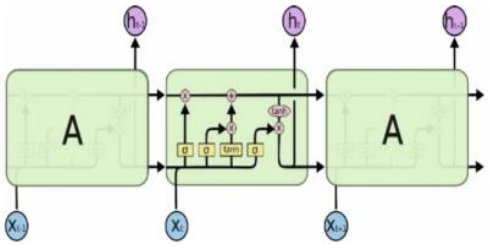

Fig. 3 LSTM structure [44]

Figure 3 shows the structure of the LSTM unit, where the repeated modules represent the hidden layer in each iteration. Each hidden layer contains a number of neurons. Each neuron performs a linear matrix calculation on the input vector and then outputs the corresponding results after the nonlinear action of the activation function. In each iteration, the output of the previous iteration interacts with the next word vector of

the text, determining the preservation or abandonment of the information and the update of the current state. X_t is the input of the hidden layer in this iteration. According to the current state information, the predicted output value of the hidden layer is obtained, and the output vector ht is provided for the next hidden layer at the same time. Whenever there is a new word vector input in the network, the output of the hidden layer at the next moment is calculated in conjunction with the output of the hidden layer at the last moment. The hidden layer circulates and keeps the latest state [44].

Several studies in the literature have reported an improvement in the performance when LSTM units are used in combination with RNN [45–47]. Another structure similar to the LSTM structure is called Gated Recurrent Units (GRU). In the study conducted by Irive et al. in 2016, LSTM and GRU approaches were compared for the same task and similar results were obtained [23]. Therefore, the proposed method was developed with an RNN having an LSTM structure.

3. PROPOSED METHOD

In the proposed method, the skip-gram feature was used to calculate the number of distant context words. The skip-gram can be looked back further by skipping certain words. For example, in the sentence "teknoloji öğrenmek en büyük hedeftir" ("learning technology is the greatest goal"), the word "hedef" ("goal") is defined as the end of the sentence. In this case, when the 1,2,3-skipgram procedure is applied, the words "en büyük" ("the greatest") refer to the skipped words. In order to reduce the complexity for the testing process of the proposed method, Skip-gram value was determined as 1. Basically, after a given k-skip value, the n-gram procedure was performed of the proposed method. The proposed method was applied to the Turkish ASR system. Besides, the effect of corpus change on the proposed method was demonstrated. In the experiments performed, interleaving was experienced in low-grade models and corpus inadequacy was encountered in high-grade models.

When modeling the proposed method, a marker was added both at the end and the beginning of the sentence. These markers represent the beginning and the end of the sentence. For this reason, the number of words was processed as $n + 2$. Fundamentally, the model was produced by two different approaches. In the first approach, a probability obtained from n consecutive sequences was calculated. In the second approach, a probability value was calculated after the skipping procedure. LM was prepared by considering Equations 5 and 6 and applied to the Turkish ASR system.

$$S(x) = P_k(x).\frac{P_{k+1}(x)}{h_k^0(x)}.\frac{P_{k+2}(x)}{h_{k+1}^1(x)}\dots.\frac{P_{2k-1}(x)}{h_{2k-2}^{k-2}(x)} \tag{5}$$

$$S(x) = P_k(x).\prod_{i=0}^{k-2}\frac{P_{i+k+1}(x)}{h_{i+k}^i(x)} \tag{6}$$

Equations 5 and 6 represent a sentence consisting of n words. Since we add a marker to the beginning and the end of the sentence, the boundary in the equations was determined as ($k \geq$. In Equation 5, $S(x)$ indicates the regular LM score of x, $P_k(x)$ indicates the S probability derived from the k-th value. $i + k(x)$ in Equation 6 indicates the probability derived from the i-skip- (i+k)-gram model. Equations 5 and 6 are combined to obtain the logarithm and so Equation 7 is obtained.

$$\log S(x) = \sum_{i=0}^{k-2}\left(\log P_{i+k+1}(x) - \log h_{i+k}^i(x)\right) + \log P_k(x) \tag{7}$$

Equation 7 can be regarded as an interpolated optimization algorithm in the log domain. Unlike conventional interpolated optimization algorithms, higher grade Markov assumptions and skip-gram features were used. Besides, a sentence-level interpolation was applied to the log domain. The purpose of the proposed approach is to model long dependencies. In the Turkish Language, sentences can be produced in long and different patterns. Therefore, the skip-gram procedure can help to model long dependencies. Our underlying assumption is that the proposed method is more reliable than the n-gram model. To explain this assumption, it is necessary to examine the structure of the Turkish Language. The Turkish Language has certain characteristics by nature. These are as follows:

- Turkish is an agglutinative language. (root - $suffix_1, suffix_2, \dots, suffix_n$)

Sunum (root) – da – ki – ler [The ones at the presentation (root)]

Çekoslavakya (root) –lı –laş –tır –a –ma –dık –lar – ı – mız –dan [One of the people we could not turn into a Czechoslovakian (The root in Turkish corresponds to "Czechoslovakia")]

- Turkish is a language with free word order and sequence. (The underlined word represents the

target word and the bold word represents the past information.)

Ben öğretmene **raporu** verdim [I gave the teacher **the report**]

P(W) = P(ben)P(öğretmene | ben) …. P(verdim | **raporu**)

Öğretmene raporu **ben** verdim[I gave the report to the teacher]

P(W) = P(öğretmene)P(raporu | öğretmene) …. P(verdim | **ben**)

Ben raporu **öğretmene** verdim [I gave the report **to the teacher**]

P(W) = P(ben)P(raporu | ben) …. P(verdim | **öğretmene**)

The characteristics mentioned above make it difficult to make a precise prediction under conditional probabilities in Turkish's. Let us assume we use a method that makes the maximum probability estimation for the language model. In this case, the best n hypothesis may not produce accurate results due to the different word combinations and the stated characteristics of the Turkish Language. According to the best n hypothesis;

- Futbol maçı gitti 0.55 (1) [Football game went]
- **Futbol maçına gitti 0.53 (2) [(S/he) Went to a/the football game]**
- Futbol maçı sordu 0.16 (3) [Football game asked]

The best result is the first result with a value of 0.55; while the correct result is the second one. This conditional probability can be changed. Equations 2, 5, and 6 indicate that conditional probabilities can be changed. Here, the $P(w_2|w_1)$ conditional probability has been replaced with $P(w_2|w_1).P(w_2|\langle/x\rangle w_1)/P(w_2|\langle/x\rangle)$. Briefly, the modeling of the past in LM is based on the $\langle/\mathrm{x}\rangle$ procedure. However, in $P(X|w_1)$ conditions, the replacement of the w_2 the condition with the X condition can be explained with the fact that the X condition is more frequently observed than the w_2 condition in the education corpus. The probability of the $P(w_2|w_1)$ condition is higher than the probability of the $P(X|w_1)$ condition. By its very nature, the Turkish Language allows for long dependencies. In this case, it should be noted that the $P(w_2|\langle/x\rangle w_1)$ the operation will yield higher conditional probability results. Theoretically, longer dependencies provide more specific semantic information about the correct words. Therefore, it is crucial to model long dependencies.

The application of the proposed method to the LM in a real Turkish ASR system presents specific difficulties. One of these difficulties is that there are more calculation operations involved in this approach than in other procedures. Particularly at the training and testing stage, selecting the k value higher necessitates the $2k-1$ model to regulate a $k+tn$ model. A single model is was used to represent the proposed method, ignoring the computational complexity (Equation 8).

$$S(x) = \prod_{j=1}^{n+1} P(w_j|w_{j-k+1}^{j-1}) \tag{8}$$

In Equation 8, the X operator aims to take out a word of the position n. For example, the $X_3 w_1^4$ the expression will produce the $w_1 w_2 w_4$ output. Thus, results regarding the re-evaluation of n-best lists are obtained. The proposed method has been applied to statistical, FNN and RNN based language models.

4. DEVELOPMENT OF THE TURKISH ASR SYSTEM

In this study, the Kaldi toolkit was used in the development of the Turkish ASR system. Kaldi is an open-source toolkit for speech recognition applications written in C ++ and licensed under the "Apache License v2.0" [48]. The Kaldi toolkit is connected to two external libraries. The first one is "OpenFst," which is used for the finite-state frame, and the other is the digital algebra library. The digital algebra library is divided into two as "BLAS" and "LAPACK". The code snippets and libraries prepared in Kaldi are called by the scripting language to create and run the ASR system.

A Turkish ASR system was developed using the 5.0 version of the Kaldi toolkit. A hybrid 3-layer LSTM was trained using the cross-entropy criterion for the acoustic model. The main LM was prepared as 3-gram. SRI Language Modeling Toolkit (SRILM) was used in the development of the model [49]. SRILM is a toolkit for building and applying statistical LMs, primarily for use in speech recognition, statistical tagging and segmentation, and machine translation. ANN-based LMs were developed using Microsoft Cognitive Toolkit (CNTK) toolkit [50].

4.1. Preparation of the Corpus

The Turkish speech data set [51] prepared by Bogazici University in 2012 and presented by LDC, the Linguistic Data Consortium, and METU 1.0 sound corpus provided by METU were used for training and testing processes of the ASR system [52]. Also, a new corpus was used to exhibit the performance of the method we propose more clearly (HS Corpus) [53]. The corpus information is given in Table 1.

The Mel frequency scale was used in feature extraction operations. The Mel frequency cepstral coefficient (MFCC) is a feature extraction technique commonly used in speech recognition systems [45]. The frequency bands are logarithmically located in the MFCC. The MFCC

calculation is based on the short-term analysis. In this study, for MFCC feature extraction, speech signals were divided into overlapping segments, and each segment was windowed. Each segment's length is 25 ms, and the overlapping ratio was taken as 40%. We calculated 13 MFCCs per segment using a Mel-scaled filter bank with 23 triangular filters distributed between 20 and 16,000 Hz.

Table 1. Turkish Corpus Information

Corpus Name	Duration in Hours	Total Number of Utterances	Number of Words	Number of Unique Words
METU	6.89	8.021	37.162	10.779
Bogazici	91.44	82.331	658.709	56.536
HS	350.00	565.760	2.976.665	252.068

4.2. Improvement of the Language Model

After the ASR system was prepared, the effect of LM on ASR performance was evaluated. There are several measurement methods available to evaluate ASR performance. The most accurate way of measurement is to assess the differences between the hypothesis and the reference word. Perplexity, which is used to compare probability models, was not used in this study. The primary purpose of the study is to evaluate the ASR system as a whole. For this reason, the evaluation process was carried out on WER in this study. WER was calculated as in Equation 9.

$$WER = \frac{D + S + I}{N} X\ 100 \qquad (9)$$

In Equation 9, N represents the total number of symbols in the reference word, D represents the number of deleted symbols in the hypothesis concerning the reference word, S represents the number of changed symbols, and I [54] describes the number of additional symbols.

The main LM was developed as 3-gram to determine the WER ratio at the ASR output. Then, the n-best lists of the main LM were revised, and WER was calculated. Firstly, Kneser-Ney smoothing algorithm, which is the most basic algorithm, was used for this process [55]. With Kneser-Ney smoothing, the n-best lists of the language model in the ASR system were updated. This process was carried out via bi-gram. There was no word restriction for LM. The number of words used is shown in Table 1. Table 2 demonstrates the obtained WER results after the main LM and n-best update.

Table 2. WER results for Turkish ASR with Statistical LM Structure

Corpus Name	3-Gram LM	Bi-Gram	Proposed Method
METU Corpus	64.5	63.5	65.0
Bogazici Corpus	22.6	21.3	21.0
HS Corpus	18.7	16.5	16.2

The 3-gram LM shown in Table 2 represents the n-gram based LM used in the ASR system. Bi-gram is the value obtained as a result of the Kneser-Ney smoothing process. The Proposed Method is explained in detail in Chapter 3. Briefly, 1-skip-2-gram represents the LM process. When the effects were examined, it was seen that the application of the proposed method in the ASR system developed using the METU corpus increased WER. The reason for this is that there are a lot of sentences composed of two or three words in the corpus. The purpose of the proposed method is to make more use of past information. When the other corpora were examined, it was found that the HS corpus gave better results in terms of both capacity and the fact that there are long sentences in its content.

The effect of the proposed method on different LMs was also investigated. Therefore, a 3-gram and FNN-based LM have been developed. The proposed method was developed as 1-skip-3-gram and 2-skip-4-gram. As in the statistical language model, there was no word limit. The corpora described in detail in Table 1 were used. The class-based output layer was used to reduce the training time of the FNN based model. The FNN model was designed to have two hidden layers. FNN LM structure was interpolated with the 3-gram LM developed as a basis. Table 3 demonstrates the effect of the proposed method on the LM structure prepared with FNN.

Table 3. WER results for Turkish ASR with FNN LM Structure

Corpus Name	3-Gram LM	FNN	Proposed Method	Proposed Method*
METU Corpus	64.5	63.0	62.4	63.9
Bogazici Corpus	22.6	20.8	20.0	19.7
HS Corpus	18.7	16.0	15.6	14.4

In Table 3, the Proposed Method* represents the 2-skip-4-gram. When the performance rates are reviewed, it is seen that the best WER result was obtained by the Proposed Method*. As shown in Table 3, the METU corpus gives better results than the proposed method in the FNN modeling. The best WER result was obtained with the HS corpus. This result can be explained essentially with two conditions. The first is that the main LM contains values that are closely matched with the LM developed by the proposed method. The other is that the training process is more successful with the HS corpus data.

The proposed method for LM is modeled with n-gram and FNN, which are frequently used in the literature. However, it has been shown in the research that the RNN structure used together with LSTM gives successful results for the LM [20, 56, 57]. Therefore, the proposed method was finally tested on the RNN-LSTM structure. Similar to FNN, there was no limit in word length. A class-based approach was used to speed up the training process. On the encoder side of the RNN-LSTM structure, a 7-layer RNN-LSTM was found, and the number of units of the LSTMs was determined to be 512. On the decoder side, there is an RNN-LSTM network with 7 layers and 512 dimensions. The developed RNN-LSTM LM used the corpus in Table 1. The proposed method was applied to the RNN-LSTM structure, and the results obtained are given in Table 4.

Table 4. WER results for Turkish ASR with RNN-LSTM LM Structure

Corpus Name	3-Gram LM	RNN-LSTM	Proposed Method	Proposed Method*
METU Corpus	64.5	62.8	62.2	64.6
Bogazici Corpus	22.6	20.3	20.1	19.2
HS Corpus	18.7	15.8	14.9	14.3

As shown in Table 4, the proposed method provided successful results also in the RNN-LSTM structure. It was observed that when the corpus capacity increased, the proposed approach gave more precise results. The results obtained by improving the LM are visualized in Figure 4.

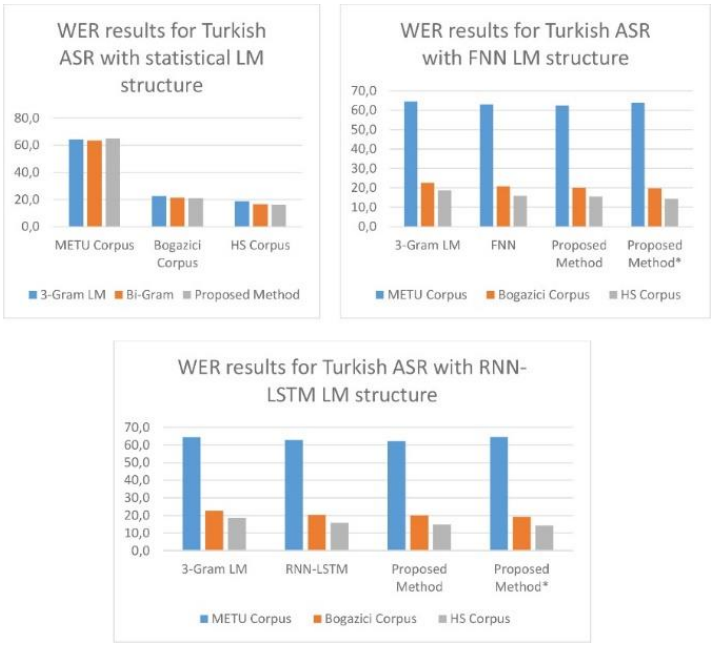

Fig. 5. LM optimization results.

As shown in Figure 4, the proposed method provided successful results on the LMs. Not only word-based LMs were used in the experiments. LMs that use sub-word levels have been developed to more clearly demonstrate the success of the proposed method. Experimental results of LMs using RNN-LSTM based sub-word level were also added to our study. The results of the experiments using the proposed method, word, and sub-word level are given in Table 5.

As seen in Table 5, the use of the sub-word level improves LM. This is true for the three corpora used in the experiments. The proposed method was not tested at the sub-word level. The reason for this is that the skip

operation gives negative results in the units that make up the words. Briefly, if a unit that creates the word is skipped or replaced, the meaning of the word is broken. For this reason, the proposed method was not used for sub-word level tests. Nevertheless, the proposed method* gave the best results in a relatively large corpus.

Table 5. WER results for Turkish ASR with RNN-LSTM LM structure developed using word and sub-word level

Corpus Name	RNN-LSTM (word-level)	RNN-LSTM (sub-word level)	Proposed Method (word level)	Proposed Method* (word level)
METU Corpus	62.8	61.9	62.2	64.6
Bogazici Corpus	20.3	19.7	20.1	19.2
HS Corpus	15.8	14.8	14.9	14.3

The proposed method has WER performance is high, but the ability to do transactions quickly is low. The proposed method requires a high number of calculations. Reduction in operation times may be possible through optimization techniques or faster computer infrastructure. In particular, an extra amount of time is spent on the revision of the N-best lists.

In this study, the N-best lists of LMs used in Turkish ASR systems have been revised. We present a new method to model long dependencies in the Turkish Language. In addition to standard approaches, tests were done by using the RNN-LSTM structure. The proposed method contributed to the standard LM creation and improvement of N-best lists. In the literature, multiple studies have been conducted with the corpora used in the study and published by LDC. Turkish ASR aims to develop systems with a low WER and to be able to calculate long dependencies. Many studies using HMM, Gaussian Mixture Model (GMM), and Support Vector Machine (SVM) have been conducted from past to present [58–61]. The current study was compared with the studies in the literature and table 5 was obtained. In Table 6, the systems developed using the METU corpus published by LDC are given.

Table 6. Results of the different ASR approach developed using the METU Corpus

Name of Authors	Recognition Approach	Name of Corpus	WER
Çiloğlu et al. [62]	HMM-N-gram	METU	35.91
Keser and Edizkan [63]	Common Vector Approach	METU	70
Salor et al. [52]	GMM-HMM	METU	29.2

When Table 6 is examined, it is seen that the same corpus has different WER results in different approaches. Studies developed using LSTM [64] gave more successful results than studies given in table 6. However, more data were added to the METU corpus in different studies. Adding data has increased the overall success of ASR systems. The system proposed within the scope of the study can model longer dependencies as it uses N-Gram and Skip-Gram methods together. More data is needed to model long dependencies. Therefore, HS corpus reveals the success of the proposed method more clearly.

The results obtained in these studies were successful compared to the METU corpus and Bogazici corpus. Optimization or improvement studies on the LM used in Turkish ASR systems have reduced the WER. Sub-word level LM was used in many of the improvement works [65, 65]. In a recent study, a different method was applied for LM and correction of ASR output was examined [64]. In this method, a template database was used. However, the sentence-level LM optimization we obtained in our study yielded much better results than the above-mentioned study when applied to RNN-LSTM based LMs.

The importance of a corpus in the performance of ASR systems should not be overlooked. The results obtained with the HS corpus [53], which is a lot larger in capacity, are more remarkable. Turkish is among the low-source languages. Therefore, there is a need for further research on the Turkish Language. The HS corpus relatively solved the problem of low resources, and long dependencies in Turkish could be modeled.

5. CONCLUSION AND RECOMMENDATIONS

In this study an LM optimization has been performed which can model long dependencies and provide better results for AM output. However, the agglutinative structure of the language and the long dependencies that the language contains make it difficult to model the Turkish Language. Therefore, an LM optimization method at the sentence level has been proposed. The proposed method has been applied to statistical and ANN-based LMs currently used in language modeling. Experimental results indicated that the proposed approach optimized the WER. In addition, it has been found that it provides better results than the ASR systems using optimization algorithms such as the Kneser-Ney smoothing. As a result, it has been presented that the performance of LMs that have a fundamental order can be optimized by rearranging the N-best lists. In the experiments performed, the proposed method has been shown to provide a consistent performance improvement when applied to statistical, FNN, and LSTM-RNN based

LMs. Experiments were conducted in different corpora for the Turkish Language. Increased corpus capacity directly affected the result, and the proposed method gave better results in large corpora. The absence of very short sentences in the corpus affected WER. However, the most substantial problem of the system is the delay in training time. The proposed method has a more complex structure than the standard models. Therefore, it requires a high number of calculations. Reducing this time to tolerable levels will improve the applicability of the proposed method in real-time Turkish ASR systems.

DECLARATION OF ETHICAL STANDARDS

The authors of this article declare that the materials and methods they use in their studies do not require ethics committee permission and / or legal-specific permission.

AUTHORS' CONTRIBUTIONS

Saadin OYUCU: Designed the ASR model and the computational framework. Carried out the implementation and performed the calculations.

Hüseyin POLAT: Performed the experiments and analyse the results.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] Hamdan P., Ridi F., Rudy H., "Indonesian automatic speech recognition system using CMUSphinx toolkit and limited dataset***", International Symposium on Electronics and Smart Devices***, 283-286 (2017).

[2] Kelebekler E., İnal M., "Otomobil içindeki cihazların sesle kontrolüne yönelik konuşma tanıma sisteminin gerçek zamanlı laboratuar uygulaması", ***Politeknik Dergisi***, 2: 109-114, (2008).

[3] Avuçlu E., Özçiftçi A., Elen A., "An application to control media player with voice commands", ***Politeknik Dergisi***, 23(4): 1311-1315, (2020).

[4] Burunkaya M. ve Dijle M., "Yerleşik ve gömülü uygulamalarda kontrol işlemleri ve pc'de yazı yazmak için kullanabilen düşük maliyetli genel amaçlı bir konuşma tanılama sistemi", ***Politeknik Dergisi***, 21(2): 477-488, (2018).

[5] Yajie, M., "Kaldi+PDNN: building DNN-based ASR systems with kaldi and PDNN", ***arXiv***:1401.6984 (2014).

[6] Davis, S., Mermelstein, P., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", ***IEEE Transactions on Acoustics, Speech, and Signal Processing***, 28, 357–366 (1980).

[7] Shreya, N., Divya, G., "International journal of computer science and mobile computing speech feature extraction techniques: a review", ***International Journal Computer Science Mobil Computer***, 4, 107-114 (2015).

[8] Tombaloğlu, B., Erdem, H., "Development of a MFCC-SVM based Turkish speech recognition system", ***24th Signal Processing Communication Applied Conference,*** 1-4 (2016).

[9] Dave, N., "Feature extraction methods LPC, PLP and MFCC. ***International Journal for Advance Research in Engineering and Technology,*** 1, 1-5 (2013).

[10] Harshita, G., Divya, G., "LPC and LPCC method of feature extraction in speech recognition system", ***International Conference Cloud System Big Data Engineering Confluence***, 498-502 (2016).

[11] Geoffrey, H., Li, D., Dong, Y., George, E. D., Abdel-rahman, M., Navdeep, J., Andrew, S., Vincent, V., Patrick, N., Tara, N. S., Brian, K., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", ***IEEE Signal Processing Magazine,*** 29, 82-97 (2012).

[12] Ussen, A. K., Osman, B., "Turkish speech recognition based on deep neural networks", ***Suleyman Demirel University Journal of Natural and Applied Sciences***, 22, 319-329 (2018).

[13] Longfei, L., Yong, Z., Dongmei, J., Yanning, Z., Fengna, W., Isabel, G., Enescu, V., Hichem, S., "Hybrid deep neural network - hidden markov model (DNN-HMM) based speech emotion recognition", ***Humaine Association Conference on Affective Computing and Intelligent Interaction***. 312-317 (2013).

[14] Xavier, L. A., "An overview of decoding techniques for large vocabulary continuous speech recognition", ***Computer Speech Language***, 16, 89-114 (2002).

[15] Stolcke, A., "Entropy-based pruning of backoff language models. ***arXiv***:cs/0006025 (2000).

[16] Biljana, P., Sinisa, I., "Recognition of vowels in continuous speech by using formants", ***Facta Universitatis - Series: Electronics and Energetics***, 379-393 (2010).

[17] Haşim, S., Murat, S., Tunga, G., "Morpholexical and discriminative language models for Turkish automatic speech recognition", ***IEEE Transaction Audio, Speech-Language Processing,*** 20, 2341-2351 (2012).

[18] Berlin, C., Jia-Wen, L., "Discriminative language modeling for speech recognition with relevance information", ***IEEE International Conference on Multimedia and Expo***, 1-4 (2001).

[19] Ahmet, A. A., Cemil, D., Mehmet, U. D., "Improving sub-word language modeling for Turkish speech recognition", ***Signal Processing and Communications Applications Conference***, 1-4 (2012).

[20] Asefisaray, B., "***End-to-end speech recognition model: experiments in Turkish***", Ph. D. Dissertation, University of Hacettepe, Ankara, Turkey (2018).

[21] Anusuya, M., Katti, S., "Speech recognition by machine: a review", ***International journal of Computer Science and Information Security,*** 6, 181-205 (2009).

[22] Dikici, E., Saraçlar, M., "Semi-supervised and unsupervised discriminative language model training for automatic speech recognition", ***Speech Communication,*** 83, 54-63 (2016).

[23] Kazuki, I., Zoltan, T., Tamer, A., Ralf, S., Hermann. N., "LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition", ***Annual Conference of the International Speech Communication Association***, 3519-3523 (2016).

[24] Siddharth, D., Xinjian, L., Florian, M., Alan, W. B., "Domain robust feature extraction for rapid low resource ASR development", ***ArXiv***: 1807.10984v2 (2018).

[25] Hirofumi, I., Jaejin, C., Murali, K. B., Tatsuya, K., Shinji, W., "Transfer learning of language-independent end-to-end ASR with language model fusion", ***IEEE International Conference on Acoustics, Speech and Signal Processing***, 6096-6100 (2019).

[26] Peter, F. B., Vincent, J., Della, P., Peter, V., Jenifer, C., Robert, L. M., "Class-Based N-gram models of natural language", ***Computer Linguistic***, 14-18 (1990).

[27] Martin, S., Hermann, N., Ralf, S., "From feedforward to recurrent LSTM neural networks for language modeling", ***IEEE Trans Audio, Speech Lang Processing,*** 23, 517-529 (2015).

[28] Tomas, M., Martin, K., Lukás, B., Jan, Č., Sanjeev, K., "Recurrent neural network based language model", ***Annual Conference of the International Speech Communication Association***, 1045-1048 (2010).

[29] Han, Z., Zhengdong, L., Pascal, P., "Self-adaptive hierarchical sentence model", ***arXiv***:1504.05070 (2015).

[30] WimDe, M., Steven, B., Marie-Francine, M., "A survey on the application of recurrent neural networks to statistical language modeling", ***Computer Speech Language***, 30, 61-98 (2015).

[31] Popova, I., Stepanova, E., "Estimation of inorganic phosphate in presence of phosphocarbohydrates (Russian)", ***Vopr Meditsinskoj Khimii***, 2, 135-139 (1977).

[32] Jen-Tzung, C., Yuan-Chu, K., "Bayesian recurrent neural network language model", ***IEEE Spoken Language Technology Workshop***, 206-211 (2014).

[33] Ebru, A., Abhinav, S., Bhuvana, R., Stanley, C., "Bidirectional recurrent neural network language models for automatic speech recognition", ***International Conference on Acoustics, Speech and Signal Processing***, 5421-5425 (2015).

[34] Ahmet, A. A., Mehmet, D., "Zemberek, an open source NLP framework for Turkic Languages", ***Structure,*** 1, 1-5 (2007).

[35] Xuedong, H., Li, D., "***An overview of modern speech recognition***." Handbook natural language process. Microsoft Corporation. 339-367 (2010).

[36] Chao, H., Eric, C., Jianlai, Z., Kai-Fu, L., "Accent modeling based on pronunciation dictionary adaptation for large vocabulary Mandarin speech recognition", ***International Conference on Spoken Language Processing***, 818-821 (2000).

[37] Erdoğan, H., Büyük, O., Oflazer, K., "Incorporating language constraints in sub-word based speech recognition", ***IEEE Workshop on Automatic Speech Recognition and Understanding***, 281-286 (2005).

[38] Daniel, J., James, H.M., "Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition", ***Journal of Perspectives in Public Health***, 1, 639-641 (2010).

[39] Arisoy, E., Dutağaci, H., Arslan, M., L., "A unified language model for large vocabulary continuous speech recognition of Turkish", ***Signal Processing***, 86, 2844-2862 (2006).

[40] David, G., Wei, L., Louise, G., Yorick, W., "A closer look at skip-gram modelling", ***International Conference on Language Resources and Evaluation,*** 148-150 (2006).

[41] Shiliang, Z., Hui, J., Mingbin, X., Junfeng, H., Lirong, D., "The fixed-size ordinally-forgetting encoding method for neural network language models", ***Annual Meeting of the Association for Computational Linguistics***, 495-500 (2015).

[42] Liu, D., Fei, S., Hou, Z. G., Zhang, H., Sun, C., "***Advances in neural networks***. Springer-Verlag Berlin Heidelberg. (2007).

[43] Eric, B., Jitong, C., Rewon, C,. Adam, C., Yashesh, G., Yi, L., Hairong, L., Sanjeev, S., David, S., Anuroop, S., Zhenyao, Z., "Exploring neural transducers for end-to-end speech recognition", ***arXiv***:1707.07413 (2017).

[44] Yiwen, Z., Xuanmin, L., "A speech recognition acoustic model based on LSTM-CTC. ***IEEE 18th International Conference on Communication Technology***, 1052-1055 (2018).

[45] Sepp, H., Schmidhuber, J., Long short-term memory", ***Neural Computation,*** 9, 1735-1780 (1997).

[46] Kyunghyun, C., Dzmitry, B., Fethi, B., Holger, S., Yoshua, B., "Learning phrase representations using RNN encoder-decoder for statistical machine translation", ***Conference on Empirical Methods in Natural Language Processing,*** 1724-1734 (2014).

[47] Rafal, J., Wojciech, Z., lya, S., "An empirical exploration of recurrent network architectures", ***International Conference on International Conference on Machine Learning***, 2332-2340 (2015).

[48] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., "The Kaldi speech recognition toolkit", ***Workshop on Automatic Speech Recognition and Understanding***, 1-4 (2011).

[49] Stolcke, A., "Srilm - an extensible language modeling toolkit", ***International Conference on Spoken Language Processing,*** 901–904 (2002).

[50] Frank, S., Amit, A., "CNTK: Microsoft's open-source deep-learning toolkit", ***22nd International Conference on Knowledge Discovery and Data Mining***, 2135-2135 (2016).

[51] Ebru, A., Doğan, C., Sıddıka, P., Haşim, S., Murat, S., "Turkish broadcast news transcription and retrieval", ***Transaction Audio, Speech Language Process***, 17, 874-883 (2009).

[52] Salor, Ö., Pellom, L. B., Ciloglu, T., Demirekler, M., "Turkish speech corpora and recognition tools developed by porting SONIC: Towards multilingual speech recognition", ***Computer Speech Language,*** 21, 580-593 (2007).

[53] Polat, H., Oyucu, S., "Building a speech and text corpus of Turkish: large corpus collection with initial speech recognition results", ***Symmetry***, 12, 290-304 (2020).

[54] Cem, A., Suha, O., Mutluergil, Hakan, E., "The anatomy of a Turkish speech recognition system", ***Signal Processing and Communications Applications Conference,*** 512-515 (2009).

[55]

[56] Vesa, S., Teemu, H., Sami, V., "On growing and pruning kneser–ney smoothed n-gram models", ***Transactions On Audio, Speech, And Language Processing***, 15, 1617-1624 (2007).

[57] Rongfeng, S., Lan, W., Xunying, L., "Multimodal learning using 3d audio-visual data for audio-visual speech recognition", ***International Conference on Asian Language Processing***, 40-43 (2017).

[58] Ahmed, A., Preslav, N., Peter, B., Steve, R., "WERd: using social text spelling variants for evaluating dialectal speech recognition", ***arXiv***:1709.07484 (2017).

[59] Reuhkala, E., Jalanko, M., Kohonen, T., "Redundant hash addressing method adapted for the post processing and error-correction of computer-recognized speech", ***International Conference Acoustic Speech Signal Processing***, 591-594 (1979).

[60] Büyük, O., Erdoğan, H., Oflazer, K., "Konuşma tanımada karma dil birimleri kullanımı ve dil kısıtlarının gerçeklenmesi", ***Signal Processing and Communications Applications Conference***, 111-114 (2005).

[61] Steve, R., Nelson, M., Hervé, B., Michael, A. C., Horacio F., "Connectionist probability estimation in HMM speech recognition", ***Transaction Speech Audio Processing,*** 2, 161-174 (1994).

[62] Yadava, G., Thimmaraja, S., Jayanna, H., "Creating language and acoustic models using Kaldi to build an automatic speech recognition system for Kannada language", ***International Conference Recent Trends Electronic Information Communication Technology,*** 161-165 (2017).

[63] Çiloğlu, T., Çömez, M., Sahin, S., "Language modelling for Turkish as a agglutinative languages", ***IEEE Signal Processing and Communications Applications Conference,*** 1-2 (2004).

[64] Keser, S., Edizkan, R., "Phonem-based isolated Turkish word recognition with subspace classifier. ***IEEE Signal Processing and Communications Applications Conference,*** 93-96 (2009).

[65] Arslan, R., S., Barışçı, N., "Development of output correction methodology for long short-term memory-based speech recognition", ***Sustainability,*** 11, 1-16 (2019).

[66] Eşref, Y., Can, B., "Using Morpheme-Level Attention Mechanism for Turkish Sequence Labelling", ***Signal Processing and Communications Applications Conference,*** 1-4 (2019).

[67] Liu, C., Zhang, Y., Zhang, P., Wang, Y., "Evaluating Modeling Units and Sub-word Features in Language Models for Turkish ASR", ***International Symposium on Chinese Spoken Language Processing***, 414-418 (2019)