PREDICTION OF WORDS IN TURKISH SENTENCES BY LSTM-BASED LANGUAGE MODELING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ABDULLAH CAN ALGAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COGNITIVE SCIENCE

FEBRUARY 2021

PREDICTION OF WORDS IN TURKISH SENTENCES BY LSTM-BASED LANGUAGE MODELING

submitted by **ABDULLAH CAN ALGAN** in partial fulfillment of the requirements for the degree of **Master of Science in Cognitive Science Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics** _____

Dr. Ceyhan Temürcü
Head of Department, **Cognitive Science** _____

Assoc. Prof. Dr. Cengiz Acartürk
Supervisor, **Cognitive Science, METU** _____

Assist. Prof. Dr. Çağrı Çöltekin
Co-supervisor, **Seminar für Sprachwissenschaft, Universität Tübingen** _____

Examining Committee Members:

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Abdullah Can Algan

Signature :

ABSTRACT

PREDICTION OF WORDS IN TURKISH SENTENCES BY LSTM-BASED LANGUAGE MODELING

Algan, Abdullah Can
MSc., Department of Cognitive Science
Supervisor: Assoc. Prof. Dr. Cengiz Acartürk
Co-Supervisor : Assist. Prof. Dr. Çağrı Çöltekin

February 2021, 45 pages

Language comprehension is affected by predictions because it is an incremental process. Predictability has been an important aspect of studying language processing and acquisition in cognitive science. In parallel, Natural Language Processing field takes advantage of advanced technology to teach computers how to understand natural language. Our study investigates if there is an alignment between human predictability and artificial language model predictability results. This thesis solely focuses on the Turkish language. Therefore, we have built a word-level Turkish language model. Our model is based on Long Short-Term Memory (LSTM), which is a recently trending method in NLP. Alternative models are trained and evaluated with their prediction accuracy on test data. Finally, the best performing model is compared to human predictability scores gathered from the cloze-test experiment. We have shown a promising correlation and analyze the cases where the correlation is high or low.

Keywords: language modeling, predictability, NLP

ÖZ

TÜRKÇE CÜMLELERDEKİ KELİMELERİN LSTM TABANLI DİL MODELLEMESİYLE TAHMİNİ

Algan, Abdullah Can
Yüksek Lisans, Bilişsel Bilimler Bölümü
Tez Yöneticisi: Doç. Dr. Cengiz Acartürk
Ortak Tez Yöneticisi : Dr. Öğr. Üyesi. Çağrı Çöltekin

Şubat 2021 , 45 sayfa

Dil anlama yeteneği tahminlerden etkilenir çünkü dil anlama artımlı bir süreçtir. Tahmin edilebilirlik, bilişsel bilimler alanında dil işleme ve edinimi çalışmalarının önemli bir yönüdür. Paralelde, Doğal Dil İşleme alanı gelişen teknolojinin avantajlarını kullanarak bilgisayarlara doğal dili öğretmeye çalışmaktadır. Bu çalışmanın amacı insan tahmin edebilme sonuçları ile yapay dil modelinin sonuçları arasındaki ilişkiyi incelemektir. Bu tez sadece Türkçe diline odaklanmıştır. Bu nedenle, kelime seviyesinde Türkçe dil modeli inşa ettik. Modelimiz son zamanlarda Doğal Dil İşleme alanının popüler bir yöntemi olan Uzun Kısa Süreli Bellek (Long Short-Term Memory) yapısını baz almaktadır. Alternatif modeller eğitildi ve modellerin tahmin sonuçları değerlendirildi. Son olarak, en iyi performansı gösteren model insan tahmin sonuçları ile karşılaştırıldı. Çalışmanın sonunda umut vadeden korelasyon sonuçları elde ettik ve korelasyonun hangi durumlarda az hangi durumlarda çok olduğunu analiz ettik.

Anahtar Kelimeler: dil modelleme, tahmin edebilme, NLP

To my family

ACKNOWLEDGMENTS

First of all, I would like to thank to my supervisors Cengiz Acartürk and Çağrı Çöltekin for their guidance. They expanded my horizon with their valuable research questions.

I also want to thank to my family for their invaluable support all my life. Especially during this journey, their encouragement helped me lot.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

DEDICATION vi

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES x

LIST OF FIGURES xi

LIST OF ABBREVIATIONS xii

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Research Questions 3

1.3 Thesis Outline 4

2 LITERATURE REVIEW AND BACKGROUND 5

2.1 Language Comprehension and Predictability 5

2.2 Statistical Language Models 8

2.3 Neural Language Models 9

2.3.1 Recurrent Neural Networks 10
2.3.1.1 Long Short-Term Memory (LSTM) 11
2.3.1.2 Gated Recurrent Units (GRU) 11
2.4 Word Representation 12
2.4.1 Word2Vec 13
2.4.2 FastText 15
3 MODEL ARCHITECTURE 17
4 IMPLEMENTATION AND EXPERIMENTS 23
4.1 Dataset and Preprocessing 23
4.2 Training 24
4.3 Complexity Comparison 26
4.4 Accuracy Results 27
4.5 GRU and LSTM 28
5 RESULTS AND DISCUSSION 29
5.1 Human vs Model 29
5.2 Generating Sentence 34
6 CONCLUSION 37
6.1 Future Work 38
Bibliography 41

LIST OF TABLES

Table 3.1 Word to ID mapping 18

Table 4.1 NLL column shows average NLL on validation set. Epoch columns indicates when the training stopped. 26

Table 4.2 accuracy of model predicting <end> correctly 27

Table 4.3 Precision @ 1 shows the accuracy of first predictions being correct. Precision @ 10 indicates actual word is predicted in top 10 predictions. . . 28

Table 4.4 Epoch column indicates where the training stopped. NLL is the average NLL result on same test set. Dropout is set to 0.3 28

Table 5.1 Word no indicates the position of a word in a sentence. Score is the human predictability results. 30

Table 5.2 Correlation results that shows the relationship between human and model predictability scores. d is the dropout rate. Parameters not shown in the table such as learning rate, optimizer, batch size are the same as shared in Chapter 3. 31

Table 5.3 Human probability scores are separated and percentage of each set is calculated. Average of model's probability is calculated for each group. 31

Table 5.4 Correlation analysis for each word with respect to their positions in a sentence. d indicates the dropout rate. 32

LIST OF FIGURES

Figure 2.1 Illustration of LSTM. (Figure Source: (Chung et al., 2014)) . . . 11

Figure 2.2 Illustration of GRU (Figure Source: (Chung et al., 2014)) 12

Figure 2.3 CBOW architecture (Mikolov, Chen, et al., 2013) 13

Figure 2.4 Skip-gram architecture (Mikolov, Chen, et al., 2013) 14

Figure 2.5 Singular to plural relationship (Mikolov, Yih, & Zweig, 2013) . . 14

Figure 3.1 RNN being unfolded over time 19

Figure 3.2 LSTM cell structure 20

Figure 3.3 a) Fully connected Neural Network with its connections b) Neural network connections after dropout is applied. Figure is taken from (Srivastava et al., 2014) 21

Figure 4.1 NLL reaches infinity when probability is 0 and it becomes 0 when probability is 1. 25

Figure 5.1 Average probability graph based on word's position 32

LIST OF ABBREVIATIONS

CBOW	Continuous Bag of Words
FFNN	Feed-Forward Neural Network
GRU	Gated Recurrent Unit
LM	Language Model
LSTM	Long Short Term Memory
NLL	Negative Log Likelihood
NLP	Natural Language Processing
OOV	Out-of-vocabulary
RNN	Recurrent Neural Network
SLM	Statistical Language Modelling

CHAPTER 1

INTRODUCTION

1.1 Motivation

Making predictions about the next time step is an essential component of almost every task that the cognitive system performs. From everyday tasks like crossing the street to critical decisions like investment plans, career choices need to be done with probabilistic predictions beforehand. The brain has a critical role in deciding the next step because it is responsible for cognitive processes underlying all these tasks. In recent studies, describing the brain as a prediction machine is increasingly popular (Clark, 2013). The nature of the predictions varies greatly (Bubic et al., 2010). Some of them are much more complex and needs long term memory, while others need focused attention at the moment. Bubic et al. (2010) stated that predictions about shorter timescales are more accurate than the long-term predictions.

To consider possible future actions, the brain models the physical environment internally. Researchers have done a notable amount of studies to investigate the prediction capability of the human brain (Bar, 2007, 2009; Bubic et al., 2010). Although the cognitive processes underlying internal models are not completely understood, it is known that new inputs from the environment update this model. Therefore, humans are adapting to the environment, and later predictions would change. This is called learning in general.

The process of learning is a continuous activity that is based on acquiring knowledge. After acquiring knowledge, the human brain can generate outputs that it has never seen before. In other words, it uses past knowledge to create new knowledge. For example, one can create a sentence after learning the grammar and necessary words. One can also understand the sentence which he/she never saw before. Using language knowledge, there is an infinite amount of possible sentences. Human has the ability to process language. Thus, human does not need to learn all the sentences.

As defined by Zhang (2019), cognitive functions of the brain are the mental process that allows humans to understand the world. These brain-based skills can be conscious or unconscious. Some of them are intuitive. Intuitive cognition provides the sensation of knowing. As Shaules (2018) give some examples, we

- know if a sentence in our native language is grammatical
- read a face and understand the emotion

- feel how much salt should be added to scrambled egg
- have a sense for how to be polite

These activities are done on a daily basis, but it is difficult to explain how we do them. Some of them are embedded in our nature. Nevertheless, it is also possible to develop intuitive knowledge for a particular domain (Shaules, 2018). One of the most important intuitive actions for a human is language processing. Essentially, language is the main tool that humans express their thoughts and feelings. Most of the time, natural language becomes highly ambiguous. However, a native speaker can understand the complex language signals (spoken, written, or signed word) and link those signals to meaning in only hundreds of milliseconds (Federmeier, 2007). Analyzing such complex structures shows that there is a complex cognitive layer at language comprehension. Federmeier (2007) stated that understanding the process underlying the language processing ability could help understand human cognition. Therefore, not only linguistics but also other fields like neuroscience and psychology have an interest in language comprehension ability.

Explaining the cognitive process of human language understanding is a popular research topic for a long time in many disciplines. With developing technology, it has attracted much attention in the computer science community too. In parallel, new methods have been proposed to teach computers to understand the human language. The field of Natural Language Processing (NLP) uses computational techniques to analyze human language. The results of the analysis are used to build artificial systems that understand the human language.

In the light of new findings, novel Natural Language Processing techniques have been introduced recently. These techniques have found their places in many practical applications. Each NLP application has a different task, but the majority of modern NLP applications build an artificial language model as a first step. This pre-trained model can then be used in a variety of tasks such as question answering, machine translation, and handwriting recognition.

There has been much less effort to build a language model for Turkish compared to other languages such as English and French. Turkish is an agglutinative language that is based on creating a complex word by concatenating a large number of suffixes. As a morphologically complex language, Turkish suffers from sparsity problems in many natural language processing tasks such as question answering, machine translation, speech recognition because each word becomes a different token when it is concatenated by any of the suffixes. Therefore, traditional methods like n-grams have performed poorly for Turkish.

The objective of this research is to train an LSTM-based language model for Turkish and compare the performance of the model to experimental data from humans performing the same task. In this thesis, we use an LSTM-based model to predict the next word in a sentence. As an evaluation score, two distinct metrics will be used. One of them is the prediction accuracy of the artificial language model that will show the model's quality, and the second one is to explore how close its predictions to human predictions. Essentially, there are two primary aims of this thesis:

1. To investigate if deep neural network architecture could model the Turkish language at word-level

2. To investigate the correlation between our LSTM-based language model and human predictability results

1.2 Research Questions

The major research question of this thesis is to investigate if there is any alignment between human predictability scores and the output of our language model. Human predictability scores are gathered from an independent reading study (Özkan et al., 2020). Our language model is based on Neural Networks that are trained on a pre-processed corpus. Like any complex machine learning method, the (deep) neural network models used in this study have to be tuned. Therefore, this thesis compares four different sized network models according to intrinsic evaluation metrics such as negative log-likelihood (NLL) and accuracy. As stated before, the main goal of this study is to build a model that shows alignment with human cloze-test answers. This thesis explores if the model that has the best score on accuracy is also correlating most with human predictability scores.

Although the architecture we design is generic, our experiments are focused solely on Turkish. Therefore, results will be discussed in the scope of Turkish language features. As stated before, the Turkish language has unique properties compared to frequently studied languages such as English, French, and Spanish. Predicting the whole word is challenging because of the vocabulary size of Turkish.

The example below would be helpful to understand the morphological productivity of Turkish. It is shown that unique words are created by adding various suffixes to stem (‘car’). Morpheme boundaries are indicated with "-".

araba	car
araba-m	my car
araba-sı	her/his car
araba-ları	their car
araba-ları-ndan	from their car
araba-ları-ndaki	at their car

In theory, it is possible to create an infinite number of words by concatenating derivational suffixes multiple times. Sak et al. (2011) give example below:

(1) ölümsüzleştiriveremeyebileceklerimizdenmişsinizcesine

(2) (behaving) as if you are among those whom we could not cause hastily to become immortal

The first one is a single word in Turkish and the second one is the equivalent in English. While being a single word, there are 11 morphemes in the first example. This suffixation makes vocabulary being too large and sparse. Another challenge is that morphemes have different forms depending on the phonology. This thesis aims to investigate how accurately a language model predicts the whole word in a sentence.

1.3 Thesis Outline

The remaining part of the thesis proceeds as follows:

Chapter 2 describes the background of our study. It focuses on language comprehension and how predictability affects this comprehension process. The chapter gives detailed information about the history of language modeling. Word representations are also discussed in this chapter.

Chapter 3 is concerned with the model design choices. Details of neural network architectures that we used in this study are presented.

Chapter 4 focus on training procedure. Details about our dataset and how it is prepared for the training are also described. This chapter presents the results of our initial experiments.

Chapter 5 discusses our model's prediction results while comparing them to human predictability results. In this chapter, the trained model is also used to generate text in Turkish.

Chapter 6 outlines the findings of our experiments and their importance. The chapter also mentions the challenges of modeling the Turkish language. Limitations of the current study and future work are also discussed in this section.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

2.1 Language Comprehension and Predictability

As the main communication tool, language is used for exchanging information about the world. Therefore, language comprehension is essential in human life, and it is extensively researched. As stated by Altmann and Kamide (1999), language comprehension is an incremental process. Meaning is built up while phrases are encountered word by word. In this process, the reader/listener has an expectation about the next word.

Prediction plays a crucial role on language comprehension (DeLong et al., 2005; Altmann & Kamide, 1999). As we read the written text, we continuously try to predict upcoming words. Predictability affects not only the speed of reading but also the movement of eyes. Therefore, predictability is one of the key variables that could explain how humans process information during reading. A considerable amount of studies (Huettig, 2015; Kuperberg & Jaeger, 2016; Willems et al., 2016) review the role of predictability in language comprehension.

Predictability is the probability of knowing the upcoming word based on the previous context. The scope of the context could change. In most cases, it is preceding words in the current sentence. However, there can be larger previous contexts like previous sentences or previous paragraphs. Sometimes, contextual information is not enough to make predictions. A reader has to use prior knowledge of the language (grammar) and the real world. For example, suppose there is a sentence with a missing last word, "the child is eating __". Using language knowledge, it can be decided that this sentence should continue with an object, but still, there are almost infinite possible answers. On the other hand, the real-life experiences narrow down the possible options. A missing word is most probably a noun denoting some kind of food.

Altmann and Kamide (1999) have investigated the relationship between verbs and their arguments. Their study shows that sentence processing is driven by predictions. The authors have recorded the participants' eye movements while they were watching a visual scene that shows a boy, a cake, and some other objects (toys, ball). Participants heard sentences with similar structure but with different verbs. One of the example setups has the following sentences.

(1) The boy will eat the cake

(2) The boy will move the cake

While they hear both sentences, their eye movements to the object were recorded. After hearing the verbs "move" and "eat", participants' gazes have moved to one of the objects in the scene. The reason is that when they are processing the sentence, they look at things that they think it will be the upcoming word. Classic Subject-Verb-Object word order in English suggests that the "object" follows the "verb". Therefore, participants focus on the object, which is the more likely argument of the verb even before hearing the word. As a result, more participants look at the "cake" for sentence (1) than they do at sentence (2). Since the cake is the only food in the scene, it shows that humans have confident predictions about the next word. In sentence (2), "move" could be applied to any of the objects in the scene, so it is not selective. Their results support that the human sentence processing procedure is affected by the predictions.

Hagoort et al. (2004) investigates the effect of truthness on predictability. They recorded the brain activity of participants as they ask them to read three different versions of a sentence. The sentences were:

(1) The Dutch trains are *yellow* and very crowded

(2) The Dutch trains are *white* and very crowded

(3) The Dutch trains are *sour* and very crowded

The first version is correct while the second one has false information according to real-world knowledge, and the third version has a semantic violation. The participants are Dutch, and it is stated that Dutch trains being yellow and crowded is a well-known fact among Dutch people. The fMRI data for the first sentence shows that participants are satisfied with their expectations of upcoming words. The second and third sentences created a fluctuation in fMRI data because some words were unexpected for them. The experiment shows that previous knowledge is used in language comprehension process. As people reading the sentence, they build an expectation for the next word. This expectation might affect not only the predictability but also the brain activity.

Predictability helps in identifying a word when there are environmental noise and speech errors. When two people talking with each other, words are spoken quickly. A person should maintain a dialogue by both understandings what the other person said and prepare an answer in the meantime. Some words may be missed because of environmental noise or speech errors. However, predictions help in those cases. Over the course of a sentence, we have a simulation about the context in our mind, which helps predict the upcoming words. Bergen (2012) gives an example sentence with one word is missing. "In my research with rabid monkeys, I've found that they're most likely to bite you when you're feeding them—you get little scars on your h...s." As Bergen (2012) stated, they assume it is a noisy environment like inside an airplane, which is easy to miss some part of the word. Even someone can fail to hear the last word, which is started with the letter "h" and ends with "s", it is possibly easy to guess that the missing word is "hands". Essentially, their claim is that the listener simulated how someone can feed monkeys and which part of the body is accessible for monkeys to bite.

These eye movement experiments show that the participant's attention is affected by predictability. However, gaze direction is not the only property of eye movements.

Oculomotor control involves many other important properties such as saccades, fixation duration, and fixation count. As Kliegl et al. (2006) stated, predictability has an influence on these measurements and is one of the "big three" factors. Other factors are word frequency and length. Many studies investigated the predictability effects on oculomotor control (Inhoff & Rayner, 1986; Rayner, 1998; Kliegl et al., 2004; Fernández et al., 2014). As a result, it is concluded that predictability has a negative correlation with fixation duration.

Together, these studies provide important insights into human language processing skills. Listeners/readers have an expectation when processing a sentence. Their expectations might be affected by different factors such as real-world knowledge, visual scene, previous context. However, a language model could only use the previous context. In this thesis, it will be investigated if that is enough to make predictions as precise as humans.

The studies cited above show the importance of predictability. However, how to accurately measure the predictability is another challenge. A large number of experiments have been done to investigate how to score predictability. The most popular method is based on a procedure called cloze test (Taylor, 1953). Cloze-test is surveying participant's reading comprehension by asking them to supply words for missing words in texts. Since the words to be removed depends on the purpose of the task, there are many variants of the cloze-test. Words can be deleted from the text randomly, every nth word, or selectively to test certain aspects of the language. Taylor (1953) designed the procedure based on how much humans tend to complete an unfinished pattern that they are familiar with. Cloze tests are also known as gap-filling questions, which are commonly used in exams at schools to evaluate the students' ability of language comprehension. For the same purpose, it is used to measure how much a computer understands the language. Word predictability is calculated as in Formula 2.1, where N is the number of participants. The total score is how many participants predict the missing word accurately. In other words, cloze probability is an indicator that reflects the expectancy of a target word in a specific context, which is computed as the percentage of individuals who supply the target word.

$$\text{Cloze Predictability} = \text{Total Score / N} \tag{2.1}$$

The number of participants should be high enough to get generalized scores on cloze-predictability. Although cloze tests have a straightforward procedure, it is hard to get a generalized result because of the limited number of participants. If the diversity of the participants is not enough, the results will be biased.

In our experiments, our language model is compared to human predictions. Human predictions are gathered from reading study (Özkan et al., 2020), which is based on the cloze-test. Details of the experiment procedure are discussed in Chapter 5.

2.2 Statistical Language Models

Language modeling aims to assign a probability to the next word in a sentence. Statistical language modeling has been a popular research field throughout the years in NLP. It is used in many language technology applications like machine translation (Brown et al., 1990), information retrieval (Ponte & Croft, 1998), document classification (Bai et al., 2004), spelling correction (Kemighan et al., 1990), handwriting recognition (Srihari & Baltus, 1992).

A language model can assign a probability to an entire sequence of words, such as a sentence. To illustrate the concept, Jurafsky (2000) gave two example sentences and stated that a language model could estimate that the sentence (1) has a higher probability to be in a text than sentence (2).

(1) *All of a sudden I notice three guys standing on the sidewalk*

(2) *On guys all I of notice sidewalk three a sudden standing the*

Such an estimation could be made by some kind of language processor. In early history, language processing systems were based on a set of rules. Rules were handwritten, which makes it impossible to cover all the grammar rules, including exceptions too. In the 1980s, computation was involved in the field of Natural Language Processing. In this era, statistical language models (SLM) are introduced. SLMs have shown notable performance.

Unlike rule-based models, Statistical Language Models (SLM) use the language training data to calculate statistical estimation. They basically rely on the frequency of occurring ***n*** consecutive words to assign a probability to the next word.

After acquiring a trained language model, a language model compute the probability of a sentence using the chain rule, as shown in Formula 2.2.

$$P(W) = P(w_1, ..., w_N) = \prod_{i=1}^{N} P(w_i | w_1 w_2 ... w_{i-1}) \tag{2.2}$$

Typically, the Formula 2.2 is used for variety length sequences. For larger ***i*** values, it is difficult to estimate it from a corpus since it requires multiple occurrences of any $w_1 ... w_i$. Therefore, limiting the previous context is necessary. Limiting preceding words to ***n*** results in Formula 2.3.

$$P(w_1, ..., w_N) \approx \prod_{i=1}^{N} P(w_i | w_{i-n} ... w_{i-1}) \tag{2.3}$$

The above technique is called N-gram, and it is one of the most widely used SLM methods. N-grams are sequences of ***n*** words that form a sentence. Their performance is highly dependent on both the quantity and quality of the training data. The last

decades have made it possible to access larger amounts of text. Therefore, state-of-art results were improved dramatically.

Simplest N-gram is the unigram. In the unigram model, the probability of a word $\boldsymbol{w_i}$ only depends on the probability of $\boldsymbol{w_i}$ being in the dataset. It does not take preceding context into calculation. In a trigram model, Language Models computes the probability of a sentence, as shown in Formula 2.4.

$$P(w_1, ..., w_N) \approx \prod_{i=1}^{N} P(w_i | w_{i-2} w_{i-1}) \tag{2.4}$$

Maximum likelihood estimation (MLE) is commonly used technique to compute N-gram probabilities. Considering trigram example, probability of word w_i can be calculated by counting how many times w_i appeared with w_{i-2}, w_{i-1} and normalizing by all occurences of w_{i-2}, w_{i-1} as shown in Formula 2.5.

$$P(w_i | w_{i-2} w_{i-1}) = \frac{count(w_{i-2}, w_{i-1}, w_i)}{count(w_{i-2}, w_{i-1})} \tag{2.5}$$

Considering how many word combinations can be made using natural language, there is a great possibility that an n-gram never occurred in the dataset (Goodman, 2001; Rosenfeld, 2000). As a result, the probability is assigned to 0, which will prevent the model to predict any words. To solve this problem, different smoothing extensions are introduced. Alternatives such as interpolation, back-off are described and evaluated in (Jurafsky, 2000; Goodman, 2001).

Rosenfeld (2000) pointed out that most popular SLM methods use almost nothing about language knowledge. The reason is that SLMs treat words as sequences of arbitrary symbols with no meaning. Major problems of N-grams:

- sparsity
- ignoring word's semantics
- ignoring long distance relationship

2.3 Neural Language Models

N-gram models maintained their dominance for a long time because of their simplicity. However, the sparsity problem prevents N-gram models from being trained on large corpora. Neural Networks (NN) are introduced to solve this problem.

Early NN architectures suffer from a large/sparse set of features. To fight against the curse of dimensionality, Bengio et al. (2003) propose a new architecture based on Feed Forward Neural Networks (FFNN). Their method is learning a distributed representation for each word. Therefore, dense word vectors are obtained. By converting

words into relatively low-dimensional vectors, FFNN language model reduces the effects of the curse of dimensionality. In this model, the conditional probability of a word "n" depends on "n-1" words. Their experiments on two different corpora show that they achieve better perplexity compared to the state-of-the-art n-gram model.

FFNN LM shows that neural networks have a promising future in the field of language modeling. However, it has some drawbacks. Feedforward networks only use fixed-size inputs. Bengio et al. (2003) also stated that introducing a priori knowledge such as semantic information is also required. Both drawbacks result in the model failing to capture some important aspects of the language.

Young et al. (2018) comprehensively compares the popular deep learning methods in the context of NLP. Last trends have shown that neural language models have outperformed pure statistical models because Neural LMs

- takes word meaning into account
- better handles OOV words
- use long term dependencies

In this section, the use of neural networks in language modeling and its advantages over traditional methods were summarized. In the following sections, another type of neural network called Recurrent Neural Network and its gated variants (LSTM, GRU) will be introduced.

2.3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are a group of neural networks that can process sequential data. The difference between FFNN and RNN is that RNNs have recurrent connections on their hidden units. Recurrent Neural Networks can also process variable-length input.

Elman (1990) introduced the idea of using RNNs in language processing tasks. There are two advantages of RNN. The first one is about the input's form. Characteristics of input data differ significantly from one problem domain to another. NLP tasks are dealing with language data, which can be both written text or speech. However, both of them are sequential. Another advantage of RNNs is that they can produce output at each time step. In word-level language models, the time step is the position of each word. In conclusion, RNNs are better in language modeling because of these two advantages.

Mikolov et al. (2010) introduced the first RNN based language model. Their experiments show that RNN LM significantly outperformed state-of-the-art n-grams and FFNN LMs.

Basic RNNs suffer from vanishing/exploding gradients (Bengio et al., 1994). To overcome these problems, RNNs are modified with a gating mechanism. These gates are supposed to control which information should be ignored or remembered by the model.

2.3.1.1 Long Short-Term Memory (LSTM)

LSTM (Hochreiter & Schmidhuber, 1997) is introduced to overcome the vanishing/-exploding gradient problem. LSTM networks have shown to be better at storing information and learning long-term dependencies than standard RNNs (Goodfellow et al., 2016). Since its introduction, Long Short-Term Memory have shown successful performance for variety of tasks such as machine translation (Sutskever et al., 2014), language modeling (Sundermeyer et al., 2012) and text classification (Zhou et al., 2015)

LSTMs make use of three gates: *input gate*, *forget gate* and the *output gate*. Using these gates, LSTMs can remove information no longer needed and also adds information that is important for the context. An illustration of these gates is shown in Figure 2.1. Further explanation can be found in (Chung et al., 2014).

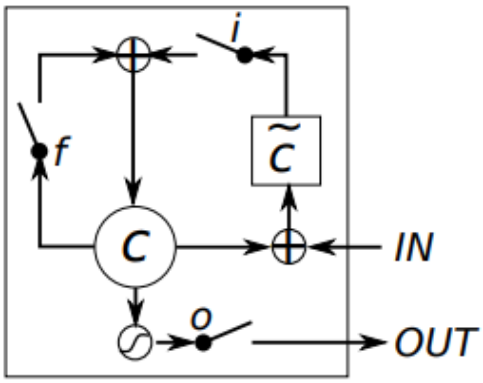

Figure 2.1: Illustration of LSTM. (Figure Source: (Chung et al., 2014))

Sundermeyer et al. (2012) introduced an LSTM-based language model. Their experiments demonstrated that they outperformed standard recurrent neural network architectures with an improved perplexity of 8%. As a recent state-of-the-art method, we have chosen LSTM as our main architecture.

2.3.1.2 Gated Recurrent Units (GRU)

GRU is another variation of gated RNNs (Cho et al., 2014). It differs from LSTM by having one less gate. Gated Recurrent Unit consists of 2 gates: *reset gate* and *update gate*. Its structure is very similar to LSTM, and GRU handles the flow of information like an LSTM. Figure 2.2 illustrate the structure of the GRU.

Empirical evaluations clearly show that gated variants, LSTM and GRU, are superior to simple RNN (Chung et al., 2014). However, they could not conclude which gated RNN is better. The authors also stated that GRU could be more efficient in terms of computation cost because its structure is simpler. In this thesis, GRU-based LM is trained to see if GRU is a better fit for our tasks. We compared their outputs using intrinsic metrics in Chapter 4.

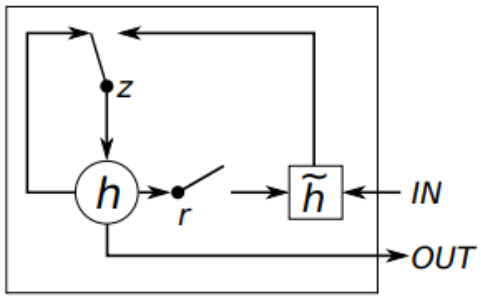

Figure 2.2: Illustration of GRU (Figure Source: (Chung et al., 2014))

Both architectures (LSTM and GRU) presented in this section are used in our experiments. As an RNN variant, they are able to produce output at each time step. Therefore, we use them to get a probability distribution for each word. In the next section, alternative word representation methods will be presented.

2.4 Word Representation

One of the biggest challenges in NLP is how to represent words accurately. In other problem domains like image processing or stock price prediction, data has formed by numbers that indicate a relative relationship with each other. However, in NLP, the input is a text which computers can not represent the same way. The text contains words that are sequences of letters. Although their meaning has a correlation with each other, it is impossible to find it just by looking at a sequence of letters. For example, synonymous words have the same meaning but are written differently. To use words in a computational model, the text has to be encoded in a numeric form. Since the words are the input to most NLP systems, word representation has become an important concern.

Traditionally, one-hot encoding was used to represent words in numerical form. One-hot encoding is vectorizing each word by creating a vector with a length equal to the number of unique words in the vocabulary set. Each word is represented by a 1 in its index, and other dimensions are set to 0. However, the semantics of the words are completely ignored by this approach. There would be no correlations between the vectors. The other problem with one-hot vectorization is the curse of dimensionality. As stated before, adding new words means adding a new dimension to the vector. Therefore, its computation cost would not be affordable at some point.

As the subject is in the core of NLP, many studies are done to come up with word representations. The fundamental purpose is to represent words as dense (mostly non-zero) and relatively short dimensional(50-500) vectors, while the representations contain information about the word's semantics. Therefore, vectors have to correlate with each other in some way or another. Linguists have had a hypothesis to define this correlation in the 1950s. It is called Distributional hypothesis (Joos, 1950; Harris, 1954). Their hypothesis has been designed on the importance of context. The idea

behind the distributional hypothesis is that similar words occur in a similar context.

Below advantages of word embeddings are the reason for them being in the standard in NLP.

- They capture word's semantic
- Word vectors are dense
- Use long term dependencies

2.4.1 Word2Vec

Word2Vec (Mikolov, Sutskever, et al., 2013) implementation has achieved to produce continuous word vector for every word in the vocabulary. As stated before, there have been many studies with the same purpose, but none of them was trainable on large data sets. However, as a revolutionary method, word2Vec introduced an efficient implementation that obtains the meaning behind the word instead of using it as a random sequence of symbols. The method has a set of two models, Continuous Bag of Words (CBOW) and Skip-gram models.

CBOW model predicts the target word depending on the surrounding words. In this model, word order does not have any effect on prediction. Model architecture is shown at Figure 2.3 (Mikolov, Chen, et al., 2013)

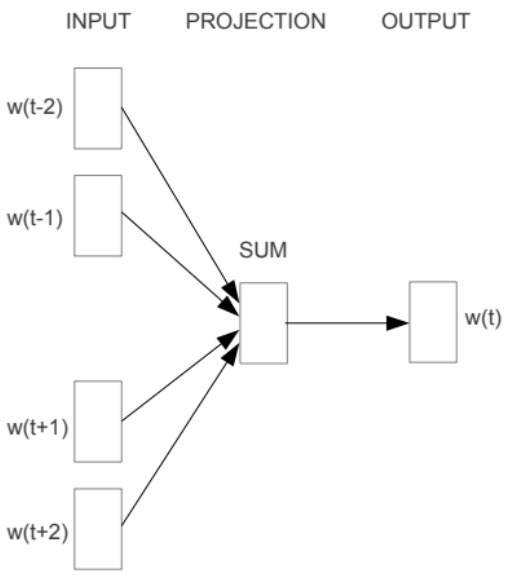

Figure 2.3: CBOW architecture (Mikolov, Chen, et al., 2013)

Skip-gram Model predicts surrounding words based on a target. Figure 2.4 shows the projection. In this model, the given the word will be used to predict its neighboring

words. Neighboring words will be determined by a window size, which is a hyperparameter to be optimized. Windows size affects the quality of the word vector directly, so increasing the size improves the quality. However, it also increases the training time by increasing computational complexity.

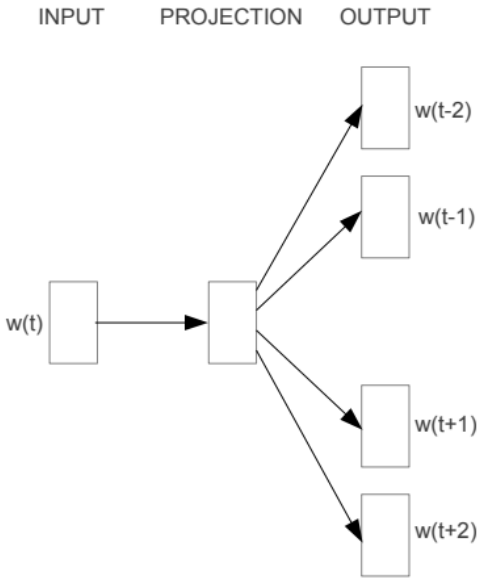

Figure 2.4: Skip-gram architecture (Mikolov, Chen, et al., 2013)

It was not the first study to use continuous vectors as a word representation. However, it is the first method to demonstrate that it is possible to create analogies using vector arithmetic. For example; Mikolov, Yih, and Zweig (2013) shows that it is possible to get male/female relationship.

$$vector('King') - vector('Man') + vector('Woman')$$

Above formula results in a vector that is closest to the word vector of 'Queen'. Another example given in Figure 2.5 display concept of singularity plurality.

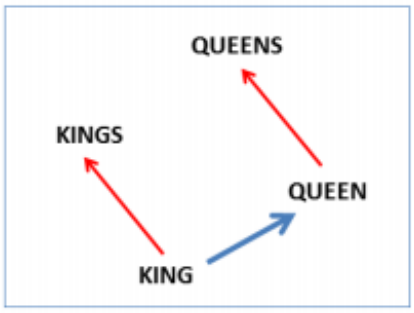

Figure 2.5: Singular to plural relationship (Mikolov, Yih, & Zweig, 2013)

Nevertheless, Word2vec has a major disadvantage. It lacks handling out-of-vocabulary words. Therefore, word2vec is unable to produce a word vector for OOV words.

2.4.2 FastText

Previous embedding methods are able to capture semantic information of the word. However, their learning process is based on only the words of their vocabulary. To cover OOV words, new methods are invented. One of the most popular methods, called FastText, is introduced by (Bojanowski et al., 2017). As the name suggests, FastText is fast at the training phase in terms of speed while outperforms previous methods in effectiveness and handling OOV.

While word2vec treats the word as the smallest unit, FastText treats each word as character n-grams. After learning representation for character n-grams, the whole word is represented as a sum of the n-gram vectors. With this approach, sub-word information is captured. Another advantage of fastText is that it can produce word vectors for OOV words.

In this thesis, fastText word embeddings are used. Turkish has a high OOV word rate because of morphological productivity. Therefore, handling OOV words is crucial for our experiments. Turkish word vectors are provided in (Grave et al., 2018). The vector dimension is 300, and they are trained using character n-gram of length 5.

In this chapter, human language comprehension was discussed. As humans have expectations based on real-life knowledge and the current context, predictions play an important role in human language processing mechanisms. In this thesis, human predictions will be compared to the artificial language model. Therefore, the current chapter also focused on language modeling. Not only the current state-of-the-art LMs but also previous methods were presented. Since gated RNNs are becoming mainstream in language modeling, we have selected LSTM and GRU architectures to perform our experiments. Design details such as hidden unit size and regularization techniques will be shared in Chapter 3.

CHAPTER 3

MODEL ARCHITECTURE

Designing an architecture requires an understanding of the task. NLP tasks have different characteristics than many other deep learning tasks. This thesis aims to investigate the correlation between human and artificial language model predictions. First, we build a language model based on Long-Short Term Memory (LSTM). Details of our architecture will be shared in this chapter.

LSTMs have some advantages over other neural network architectures. Firstly, it has the ability to process the input of any length. The second advantage is that historical information is taken into account without suffering vanishing/exploding gradient problem. As with any machine learning method, the LSTM structure has some parameters to be decided before training.

Besides tunable hyperparameters, there are other design choices to make. In accordance with common practices at word-level language modeling (Melis et al., 2017), all models use the following hyperparameters.

- has a batch size of 64.
- have trained for a maximum of 15 epochs
- have a learning rate of 0.0005
- have used Adam Optimisation (Kingma & Ba, 2014)

Input interpretation is primary challenge in NLP. In this thesis, dense word vectors are used instead of one-hot vectors. Since words are not understood by computers, words need to be converted into numeric form. First, every word in the vocabulary is mapped to a unique integer ID. For example, assume our training data consists of two sentences ["<start> hello how are you <end>", "<start> what are you doing today <end>"]. After tokenization, each unique words are mapped to word ID resulting in Table 3.1

Table 3.1: Word to ID mapping

Words	ID
<start>	1
<end>	2
hello	3
how	4
are	5
you	6
what	7
doing	8
today	9

To equalize the length of the sentences, post padding is used. Therefore, ID 0 is not used in Table 3.1 and it is reserved for padding. Sentences "<start> hello how are you <end>" and "<start> what are you doing today <end>" would be represented as [1, 3, 4, 5, 6, 2, 0] and [1, 7, 5, 6, 8, 9, 2], respectively. Target sequence is equal to input sequence shifted one time step. Sample input-target sequence is shown in the following example:

Input 1: [1, 3, 4, 5, 6, 2, 0] Target 1: [3, 4, 5, 6, 2, 0, 0]

Input 2: [1, 7, 5, 6, 8, 9, 2] Target 2: [7, 5, 6, 8, 9, 2, 0]

Each word is mapped to its corresponding pre-trained embeddings. All the models in this experiment take advantage of neural word embeddings. We have chosen FastText embeddings because of three reasons. The first reason is that FastText word embeddings handle OOV words. The second reason is that it uses subword information. Lastly, there are published pre-trained embeddings for Turkish.

Figure 3.1 illustrate the flow in basic RNN model. Our words are converted into an embedding matrix before feeding the input layer. The figure shows only one layer of RNN. However, neural networks are trainable models that extract high-level features from raw input and use them to learn hierarchical representations. Therefore, multiple layers could be stacked on top of each other to increase complexity.

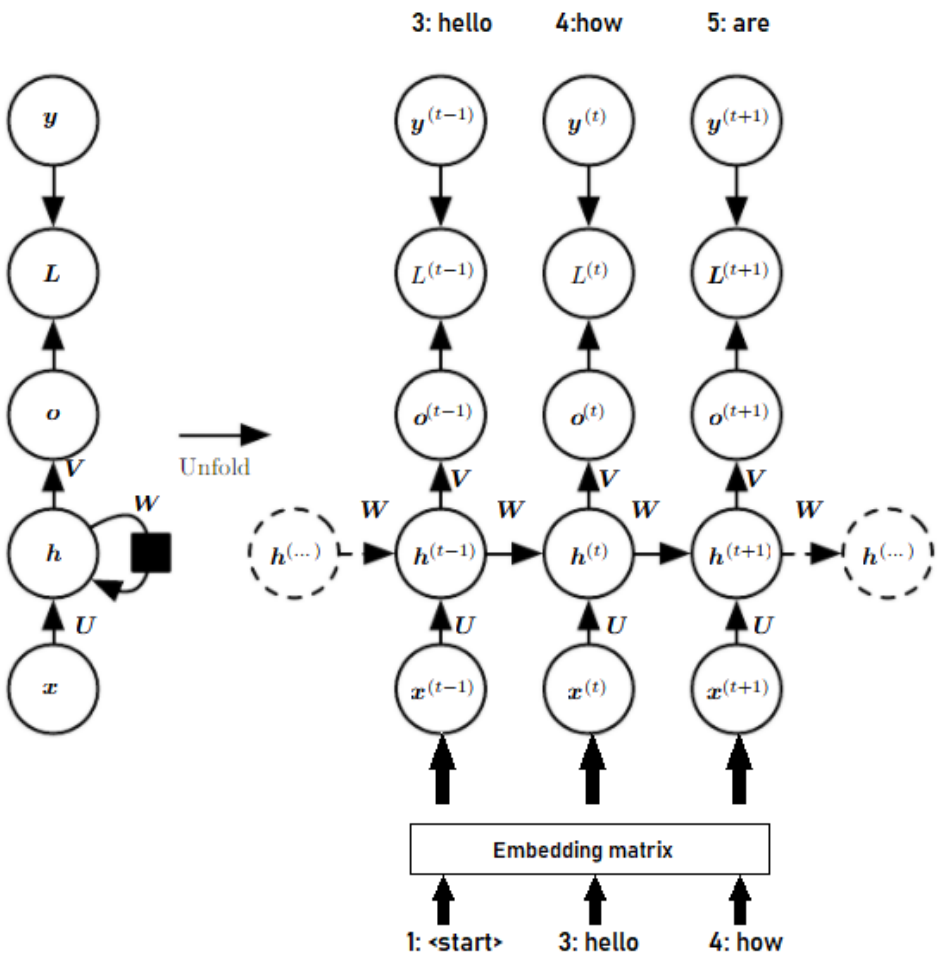

Figure 3.1: RNN being unfolded over time (Goodfellow et al., 2016). **x** is input sequence which is mapped to output **o** values. The loss **L** computes the difference between o and corresponding target **y**. Left side shows the network with recurrent connections. Right side is the time-unfolded version. In our case, one time-step means one word in a sentence.

Fig 3.1 displays the repeating modules in basic RNN structure. In our study, we use LSTM cells, which have gates in their structure. Fig 3.2 illustrate the details of LSTM. In the figure, current input is indicated with x_t and h_t is the output. There are three gates to control the current state (C_t). These gates contain a sigmoid activation function which gives values between 0 and 1. f_t is the forget gate, and it is responsible for choosing which information to be forgotten by manipulating the current state C_t. i_t gate is called input gate. It is controlling if the current input should be preserved or not. Output gate, o_t, decides the final output by updating hidden state h_t.

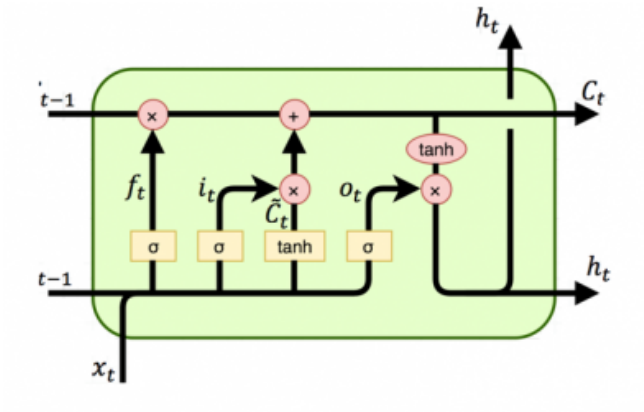

Figure 3.2: LSTM cell structure [1]

The purpose of training a model is to obtain optimal generalized performance. It is one of the challenging problems with deep neural networks because their architectures are very complex. Therefore, neural networks tend to overfit to training data. Overfitting occurs if the model achieves great accuracy on the training data while it does not on the test (unseen) data. It implies that the neural network model has not "learn" but "memorize" training data instead.

To prevent overfitting, some precautions should be taken. One alternative is early stopping. Early stopping is controlling the training procedure by monitoring evaluation metrics in regular intervals, typically at the end of every epoch. In the present study, we have reserved some portion of our data for computing validation accuracy at every epoch. We monitor the validation accuracy to decide when to stop training.

Another advantage of neural networks is that they have different regularization techniques to improve performance. One of the most popular ones is called dropout, introduced by Srivastava et al. (2014). The authors stated that dropout, as a term, refers to dropping out units in a neural network. Their method is based on removing the unit from a fully connected network, including its connections, both incoming and outgoing ones. Which units to be removed are chosen randomly based on a given probability. Figure 3.3 shows the effects of using dropout. We have applied different dropout rates in our implementation.

Recently, novel architecture called Transformers is introduced (Vaswani et al., 2017). Transformers aims to solve sequence-to-sequence problems while handling long-term dependencies. Last trends suggest bi-directional methods such as BERT (Devlin et al., 2018), showing improved NLP tasks performance. Their novelty is that bi-directional methods look at the input from both directions. However, our research question is to find alignment with human predictions, so bi-directional methods are not suitable for our study. This thesis aims to approximate a cognitively plausible model.

Single Layer Architecture: In this thesis, three different sizes of LSTM models will be used. To investigate the effect of complexity, 400, 800, and 1200 hidden units

[1] Figure is taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs

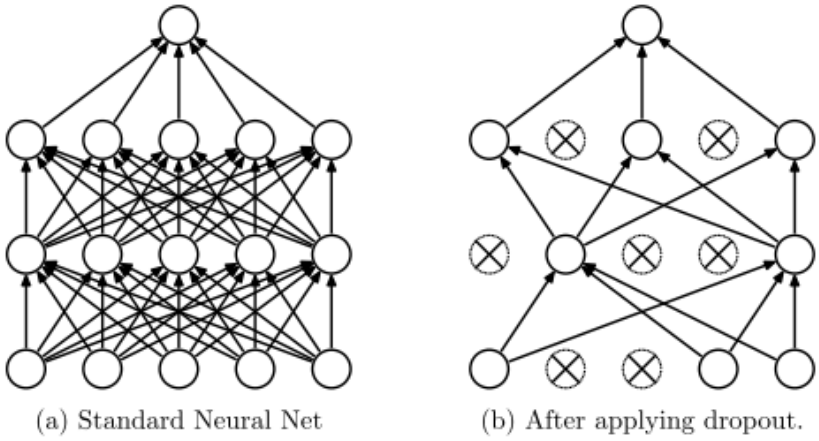

Figure 3.3: a) Fully connected Neural Network with its connections b) Neural network connections after dropout is applied. Figure is taken from (Srivastava et al., 2014)

are chosen. The maximum size is chosen as 1200 because increasing complexity means more training time/computation resources and a higher chance of overfitting. Therefore, there needs to be a limit.

Stacked Layers: Deep Learning models have layers. The first and last layers are called input and output layers, respectively. Between the input and output layers, there is a layer called, hidden layer. Stacking hidden layers on top of each other is possible. Additional hidden layers increase the levels of abstraction, which makes the model deeper. Therefore, some model structures use stacked layers. In our study, 2 stacked LSTMs are used in addition to single LSTMs. The output of the first LSTM layer is the input of the subsequent LSTM layer.

In this chapter, we shared the details of our neural network architectures. We have implemented all of our alternative models using TensorFlow (Abadi et al., 2016). As our first purpose, we aim to build a language model based on LSTM. LSTM structure has some parameters to be decided before training. We considered our experiment's purpose while deciding these parameters, such as hidden unit size, learning rate, dropout, and early stopping.

In the next chapter, information about our dataset and how it is preprocessed will be shared. After preprocessing, the training procedure will be discussed in detail. Trained models will be evaluated with intrinsic measurements such as negative log-likelihood and precision.

CHAPTER 4

IMPLEMENTATION AND EXPERIMENTS

4.1 Dataset and Preprocessing

All experiments in this study are performed on the corpus gathered from the most popular news websites in Turkish. Corpus is mostly formed from news about politics, economics, and sports. Since all sentences need to be of equal size at the beginning of training, every sentence will be padded to match the longest sentence length. Some of the news has very long sentences (80 words) because of two reasons. First, some sentences contain a list of proper names, such as the list of participants in a competition. They would not contribute to the model about Turkish grammar. The second reason is that sentences are not split correctly unless they use appropriate punctuation. Web crawler parses a couple of sentences at once and counts them as a single sentence. Therefore, limiting the maximum length was necessary. Our preprocessing procedure is as follows.

- Punctuations are removed
- Letter with circumflex are converted to their equal. For example; â → a, ê → e, î → i
- All characters are turned into lower case
- <start> and <end> symbols are added to beginning and end of every sentence, respectively.
- Numbers are replaced with <num> tag
- Vocabulary set is created with the most frequent 500K words
- Sentence length is restricted to be between 4 and 15 (5 and 16 with <start>, <end>)
- Words are marked <unk> if it is not in the vocabulary set
- Sentences are removed if they have more than two <unk> markers
- We have removed non-Turkish sentences using python package named Turkishnlp. [1]

[1] https://pypi.org/project/turkishnlp/

- Common spelling errors about Turkish characters are corrected. For example; antreman → antrenman, şarz → şarj

As any data-driven approach, training data is critical in our study. This is why preprocessing was the first step before training. With the above preprocessing steps, we intended to improve data quality. Next, we started to train our model, as presented below.

4.2 Training

As stated before, Turkish has a huge vocabulary size because of its morphological richness. Therefore, we limit the vocabulary size. Dataset was restricted to the most frequent 500K words, but after <UNK> and <NUM> tokenization, 479294 words are left. After splitting randomly, training data has 5024750 sentences, and the validation set has 25250 sentences. The test set has 120 sentences that are nine words long.

Although there are numerous studies about language modeling with LSTMs, researches to date has not yet determined the ideal parameter values. Therefore, our training procedure also aims to find optimized values for hyperparameters. Hyper parameter values;

- Hidden units: {400, 800, 1200}
- Dropout rate: {0.1, 0.2, 0.3, 0.4}

Early stopping is a commonly used technique in the neural network training process. A neural network structure has a lot of hyperparameters. These parameters should be optimized to get a generalized model. All the parameters are directly related to network architecture except the epoch number. The epoch number is how many cycles the data is passed forward and backward through the neural network. Therefore, it has a huge impact on the end result. Model updates its neurons' weight in every epoch. A high epoch number could result in a model that memorizes the training data instead of learning the concept. This can only be understood at the testing phase. This is called overfitting. Overfitted models would perform poorly on test data, which is seen the first time.

To prevent overfitting, our training procedure has been designed to end with custom early stopping mechanisms if it does not improve for three consecutive epochs. Custom early stopping mechanism is based on a metric called negative log-likelihood (NLL). At the end of every epoch, the validation set is used to calculate the average NLL of the sentence. NLL is calculated using log base two, as shown in Formula 4.1.

$$\text{Word NLL} = -\log_2 P(w) \tag{4.1}$$

P(w) comes from our model's output layer. At the output layer, the softmax activation function is placed. The output of softmax describes the probability of a sample

belongs to a certain class. In our study, it implies how likely a particular word could be the next word. Softmax assigns probabilities in the range of 0 and 1 while the sum of all the probabilities equals 1. In other words, it returns a probability distribution. Therefore, using softmax output improves the interpretability of the network. All these advantages of softmax make it possible to understand the behavior of the model.

From word NLL, average NLL of a single sentence and then validation set NLL is calculated as shown in Formula 4.2 and Formula 4.3, respectively.

$$\text{Sentence NLL} = \frac{1}{N}\sum_{n=1}^{N} WordNLL \tag{4.2}$$

$$\text{Validation set NLL} = \frac{1}{N}\sum_{n=1}^{N} SentenceNLL \tag{4.3}$$

As shown in Formula 4.3, our accuracy metric is averaging NLLs for all validation data. Approximately 25K sentence were used only as validation set. The relation between NLL and probability score for a binary random variable is shown at Figure 4.1.

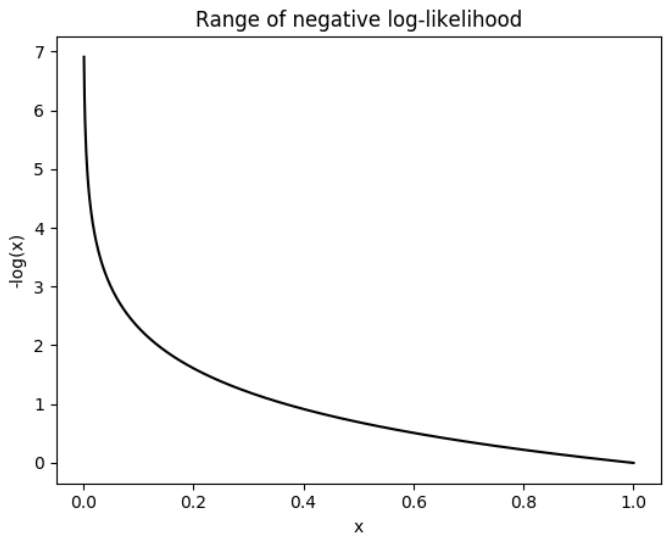

Figure 4.1: NLL reaches infinity when probability is 0 and it becomes 0 when probability is 1.

When calculating sentence probability, word probabilities are multiplied. Since probabilities are less than or equal to 1, it results in numerical underflow. Therefore, computing probabilities in log format is a common method to get relatively bigger numbers. It is possible to take a logarithm in any base, but typically, base two is used.

As shown in Figure 4.1 there is an inverse relationship. Therefore, in the NLL metric, lower means better performance.

Empirical evaluation of language models is challenging, especially if it is impossible to use the same training and test data for all candidates. For a word, NLL is the surprisal of the word that has proved to be a powerful metric. As the surprisal of the word gets higher, the word becomes less expected. In other words, surprisal is correlated with difficulty in predictability (Levy, 2008). The cognitive workload required to process a word is also proportional to its surprisal (Hale, 2001).

We have trained our alternative models, as presented above. As stated before, some of the hyperparameters need to be tuned, such as hidden unit size. To finalize the hidden unit size, we will compare four different sized LSTM models. The first measurement was NLL results, and then their prediction accuracy will be evaluated in the next two sections.

4.3 Complexity Comparison

In this section, four different models are compared to investigate the effects of model complexity. As stated before, LSTMs have hidden units, and increasing the hidden unit size also increases the complexity of models. To find an optimal hidden unit, three different sizes (400, 800, 1200) are used. The learning rate and the dropout rate is set to 0.0005 and 0.3 for all models, respectively.

Table 4.1: NLL column shows average NLL on validation set. Epoch columns indicates when the training stopped.

Model Architecture	***NLL***	***Epoch***
LSTM 400	9.20	7
LSTM 800	9.04	7
LSTM 1200	8.94	7
LSTM 1200 - LSTM 1200	**8.69**	8

Two stacked LSTMs with 1200 hidden units have shown the best performance. It was expected for more complex models to show better performance, but increasing complexity means more training time and computational resources. Considering that each epoch lasts approximately 10 hours, there needs to be a limit. Also, increasing complexity means increasing the risk of overfitting. As it can be seen from Table 4.1, all models early stopped instead of continuing training for maximum epoch number. Nevertheless, since it is only evaluated with intrinsic metric, it does not imply this will be the most successful model for any NLP task.

4.4 Accuracy Results

This thesis primarily focuses on investigating the correlation between human and computer predictability scores. However, we further analyzed the quality of the models to choose between the alternatives. In this section, the prediction results of each model will be discussed.

Two different tasks are performed. The first one is the end of sentence analysis. This task focus solely on the last word in a sentence. Model is given the first nine words. Then, it predicts the last word based on these words. The second task is called precision at 1 and precision at 10. In this task, a model is used to get predictions for each word in a sentence. Precision at 1 considers only the word with the highest probability, while precision at 10 considers top 10 predictions.

Same test sentences are used for all tasks. The test set contains 120 sentences. There are 1200 words, including the <end> marker.

End of sentence analysis: The task is based on how many sentence endings are correctly predicted by the model. Models are given a set of sentences with an ending tag of <end>. If a final softmax probability of <end> is the highest, it received a score of 1.

Table 4.2: accuracy of model predicting <end> correctly

Model Architecture	***Accuracy***
LSTM 400	97.5%
LSTM 800	97.5%
LSTM 1200	98.33%
LSTM 1200 - LSTM 1200	**99.16%**

As shown in Table 4.2 two-layered LSTM model has achieved the best performance. It predicted the <end> in its first prediction for 119 out of 120 sentences.

End of sentence analysis is essential because in Turkish, word order suggests that verb is the last word most of the time. Therefore, the results would give an idea of how much the model learn about verbs.

Precision at 1 and 10 is analyzed by calculating how many words are predicted by the model. In this task, the first words are excluded because there would be no previous context to predict. Since <end> marker analyzed previously, the model is not expected to predict <end> marker in this task. 2 stacked LSTM model, same as the previous task, outperforms other alternatives with slightly better precision. Table 4.3 shows the prediction accuracies.

Table 4.3: Precision @ 1 shows the accuracy of first predictions being correct. Precision @ 10 indicates actual word is predicted in top 10 predictions.

Model Architecture	***Precision @ 1***	***Precision @ 10***
LSTM 400	15.73%	35.83%
LSTM 800	16.88%	38.85%
LSTM 1200	18.13%	38.23%
LSTM 1200 - LSTM 1200	**18.23%**	**40.00%**

4.5 GRU and LSTM

There is another well-known variation of gated RNNs, which is called GRU. Both methods are reported to perform almost the same (Fu et al., 2016; Irie et al., 2016; Greff et al., 2016; Shewalkar et al., 2019). LSTM seems to have slightly better performance, especially for larger hidden units. However, GRU has a lower training time, and it is simpler in terms of computation. To decide which architecture is better for our purpose, we have compared LSTM and GRU architectures. We have already observed that 2 stacked LSTM with 1200 hidden units outperformed our other alternatives. Therefore, GRU with the same configuration is trained, and their comparison is shown at Table 4.4

Table 4.4: Epoch column indicates where the training stopped. NLL is the average NLL result on same test set. Dropout is set to 0.3

Models	***Epoch***	***NLL***	***<end>***	***Precision @ 1***	***Precision @ 10***
2 x LSTM 1200	**8**	**8.59**	**99.16%**	**18.23%**	**40.00%**
2 x GRU 1200	9	8.87	98.33%	17.29%	39.17%

In our case, both better performance and lower training time are achieved by LSTM. Therefore, LSTM is chosen for our final experiment.

In this chapter, dataset preprocessing and training procedure were described. Both single-layer LSTM and 2 stacked LSTM structures were trained and evaluated with intrinsic measurements. Among single-layer architectures, the model with 1200 hidden units displayed better performance than smaller alternatives (400 and 800 hidden units). As the results show, 2 layered LSTM architecture outperforms other alternatives. However, it is not our purpose to build a model that is best at intrinsic evaluation. Our primary objective is to approximate human behavior. Therefore, the final experiment will be analyzing the correlation between humans and our model. In the following chapter, different dropout rates will be applied.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Human vs Model

In this section, the output of the models is compared to human predictions. LM predictions are usually evaluated by comparing them to the original word in a test sentence. It is very rare to correlate these predictions to human predictions (Bianchi et al., 2014). Bianchi et al. (2014) have obtained their computer-predictability scores using computational algorithms such as N-grams, Latent Semantic Analysis, Word2Vec, and fastText. In the present work, the LSTM-based language model is used. To our knowledge, there are no previous studies in Turkish.

Human predictability scores have been gathered from an independent study (Özkan et al., 2020). Their experiment is based on a cloze test, which is used to measure the sentential predictability of the target word. The experiment was done with a total of 80 people (40 males, 40 females, the average age of 24). Test sentences are independent, and the order is random. The first words of the sentences were excluded, as there is no sentential context to make predictions. Therefore, eight words are left to be predicted by readers. The participants were asked to supply a word for the upcoming position. After each prediction, participants are told the correct word so that the next target would be predicted based on the intended sentential context. After each word predicted by 10 participants, whole-word predictions are compared with the original words. Correctly predicted word received a score of 1, while others are scored as 0. The probability of predicting a word was calculated over 10 participants, which is the attempt number.

Since we want to investigate the correlation between human and LM predictability scores, our model is set to output probabilities for each word. Both probabilities are between 0 and 1.

The test set contains 120 sentences, and all of them have nine words. Same test sentences are used for both human and computer experiments. Test sentences are preprocessed with the same steps, which are described in Chapter 4. A sample sentence and predictability scores can be seen from Table 5.1.

Test sentence: *Bu karışık dünyada senin yaşadığını bilmek beni mutlu ediyor.*

('It makes me happy to know that you live in this messy world.')

Table 5.1: Word no indicates the position of a word in a sentence. Score is the human predictability results.

word	***word_no***	***Score***
bu	1	-
karışık	2	0.0
dünyada	3	0.1
senin	4	0.0
yaşadığını	5	0.0
bilmek	6	0.3
beni	7	0.1
mutlu	8	0.4
ediyor	9	0.8

The score is calculated using Formula 5.1. As expected, human predictability gets higher towards the end of the sentence. The reason may be that the human brain is building up the current context and narrows down the possible words.

$$\text{Score} = \frac{\text{number of correct predictions}}{\text{total attempts}} \tag{5.1}$$

Our intrinsic evaluation pointed out that two stacked LSTM with 1200 hidden units outperformed other alternatives in previous experiments. However, the dropout was set to 0.3. As one of the most important regularization methods, dropout has a great effect on the end result. Therefore, we trained three more alternatives to our main LSTM model. We keep the same model design except for dropout. In addition to 0.3, dropout rates of 0.1, 0.2 and 0.4 are added. In total, seven different neural language model architectures were presented in this comparison task. We follow the same procedure as the cloze test to get a prediction from our trained language models.

As our primary objective, we investigated if our language model predictions are aligned with the human cloze test predictability scores. We have used correlation analysis as a measurement. Correlation analysis is a statistical technique used to evaluate the strength of a relationship between two quantitative variables. Correlation results are between -1 and 1. 1 is the highest correlation, while the weakest relationship is indicated with 0. -1 is a strong inverse correlation result.

Table 5.2: Correlation results that shows the relationship between human and model predictability scores. d is the dropout rate. Parameters not shown in the table such as learning rate, optimizer, batch size are the same as shared in Chapter 3.

	Correlation
LSTM 400 d:0.3	0.51
LSTM 800 d:0.3	0.55
LSTM 1200 d:0.3	0.56
2 x LSTM 1200 d:0.4	0.57
2 x LSTM 1200 d:0.3	0.58
2 x LSTM 1200 d:0.2	**0.60**
2 x LSTM 1200 d:0.1	0.58

Softmax outputs are used as probability scores for the LSTM. As described in Section 4.2, softmax probabilities are assigned between 0 and 1, similarly as human scores, shown in Table 5.1.

As Table 5.2 shows, all of our models have a correlation higher than 0.5, which is notable. However, the best score is achieved with the 2 stacked LSTM and when the dropout rate is set to 0.2. At first glance, a 0.6 correlation seems promising. We explore the correlation results in detail. For this purpose, we have separated the human probability results into different bins and calculate the model's average probability for the respective bin. Table 5.3 shows that model's average probability is lower than the human scores in each set.

Table 5.3: Human probability scores are separated and percentage of each set is calculated. Average of model's probability is calculated for each group.

Human's Probability	**Percentage**	**Model's Probability**
0.0 - 0.1	75.10%	0.03
0.2 - 0.3	11.46%	0.12
0.4 - 0.5	6.25%	0.28
0.6 - 0.7	3.85%	0.34
0.8 - 0.9	2.50%	0.38
1.0	0.83%	0.54

To investigate further, we have analyzed correlations for each word with respect to

their position in a sentence. The result Table 5.4 is consistent with our previous findings. 2 x LSTM 1200 d:0.2 shines out once again with its high correlation results.

Table 5.4: Correlation analysis for each word with respect to their positions in a sentence. d indicates the dropout rate.

	2	3	4	5	6	7	8	9
LSTM 400 d:0.3	0.73	0.82	0.68	0.50	0.34	0.40	0.47	0.39
LSTM 800 d:0.3	**0.80**	0.80	0.70	0.57	0.44	0.53	0.50	0.38
LSTM 1200 d:0.3	0.70	0.83	**0.72**	0.57	0.40	0.49	0.53	**0.47**
2 x LSTM 1200 d:0.4	**0.80**	0.81	0.71	0.56	0.57	0.50	0.52	0.43
2 x LSTM 1200 d:0.3	0.77	0.80	0.71	0.56	0.57	**0.60**	0.52	0.43
2 x LSTM 1200 d:0.2	0.75	0.83	0.71	**0.63**	**0.63**	0.52	**0.57**	**0.47**
2 x LSTM 1200 d:0.1	0.77	**0.84**	0.68	0.56	0.62	0.57	0.49	0.45

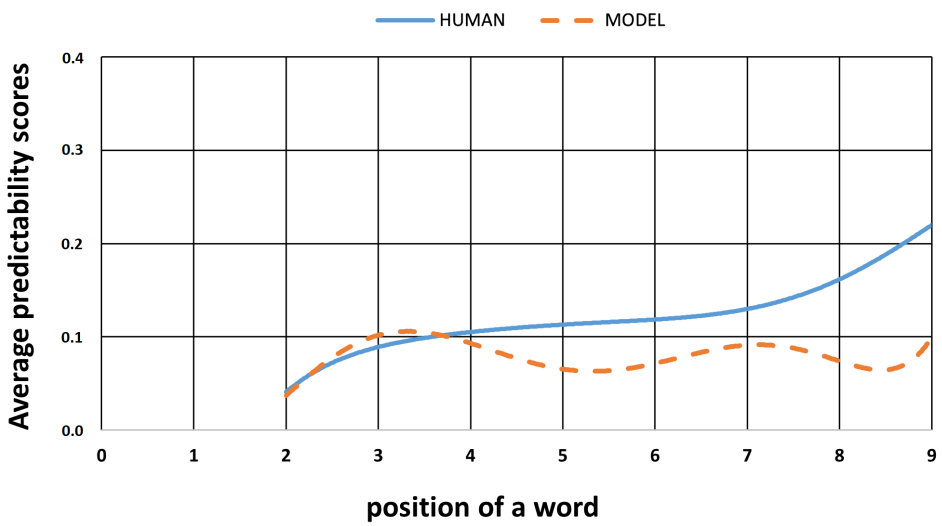

Figure 5.1: Average probability graph based on word's position

Figure 5.1 shows average probability scores on each word. As stated before, the first word is excluded since there would be no previous context to make a prediction. The correlation is very high in the first words. However, the difference between predictability scores is high in eighth and ninth words. Human predictabilities are getting higher to the end, as expected. However, our model could not show the same performance, which results in lowering the overall correlation.

When we further investigated the results, predictabilities seem mostly aligned. How-

ever, some sentences have very low correlation results. There are two reasons for the low correlation. Human predictability scores might be much higher than the model or vice versa. Both cases can be seen in our experiments. Therefore, a low correlation does not mean that our model has shown poor predictability performance. There are some sentences that our model displays higher predictability scores than humans. However, our research motivation is to find the correlation between two results, not the one that is better at predicting the upcoming words. For this purpose, we have taken some precautions to prevent overfitting, which would increase predictability scores on certain test sentences. We have investigated the correlation results in detail to decide future directions of the present study. In the below cases, human predictability scores are higher.[1]

- *Rare phrases:* Humans have higher predictability scores on phrases that are not present or rarely present in our training dataset. Some examples: "taze ve çıtır simit ('fresh and crispy bagel')", "renkten renge girmek (to go all shades of red/purple')", "çil yavrusu gibi dağılmak ('run off in all directions')", "hiç mi hiç ('not in the least')". Some phrases are idioms, a group of words that have a particular meaning different from the meanings of each word on its own. FastText word embeddings handle some phrases because it assigns similar word vectors to synonyms, but rare phrases mostly decrease the correlation. Humans take advantage of their real-world knowledge.

- *Last words:* Correlation is decreasing to the end of the sentences. There might be several reasons for that. One reason could be that our model cannot handle long context and needs to be modified. Another reason might be that human predictability is getting much higher than normal because all the test sentences are nine words long. People's prediction choices might be affected by this information.

- *Suffixes:* Model has shown poor performance on suffixes such as tense, plurals. As shown below examples, the model is aware of the context and can correctly predict the root. However, words with suffixes decrease the whole-word predictability.

Actual word	**Model prediction**
lastikleri ('its tires')	*lastiği* ('tires')
sokaklar ('streets')	*sokaklarda* ('at streets')
düşünmüşümdür ('i have thought')	*düşünüyorum* ('I'm thinking')
niyetleri ('their intentions')	*niyeti* ('his/her intentions')

Cases where LM predictability scores are higher than humans.

- *Formal sentences:* Our model has shown better prediction results on formal

[1] Turkish to English translation is done by the author.

sentences. It was expected because our training data consist of sentences taken from the news. Some example sentences are presented below.

- "Ana cadde ve sokaklar araç ve yaya trafiğine kapatıldı." ('Main avenue and streets are closed to vehicle and pedestrian traffic')
- "Dev uçağın kaptan pilotunun kadın olması da dikkat çekti." ('It also draws attention that the captain pilot of the giant plane is a woman')
- "Tıp tarihi açısından bu son derece önemli bir olaydır." ('This is an extremely important event in the history of medicine')

- *Frequent words and clitics:* As any data-driven approach, our language model is affected by training corpus. Since the model largely lacks the other sources of information (world knowledge, better representation of semantics), it relies more on word frequency in comparison to people. Model's prediction is biased towards most frequently used words such as "ve", "bir", "de", "da", "ki".

We have analyzed seven different LM outputs by comparing their softmax probability outputs to human predictability scores. The findings listed above suggest that predictability is affected by various properties of Turkish. In the following chapter, a model with the highest correlation to human behavior will be used to generate a complete sentence. Generated sentences will be analyzed in terms of word choices.

5.2 Generating Sentence

In this section, the pre-trained language model is used to complete a sentence. 2 layered LSTM with 1200 hidden units is selected for this task because it outperformed other models in our previous experiments. Dropout is set to 0.2.

The model is given a word sequence to perform text generation. Model is expected to predict the next word based on given words. Model output the softmax probabilities for each word in the vocabulary, and the highest probability is selected. Once the model has a prediction for the current position, the predicted word is appended to the initial sequence. A new sequence is used as the subsequent input and get prediction again. This prediction procedure is repeated until <end> marker is predicted or maximum word length is reached. The maximum word length is 16.

Some example sentences will be shown in (1) - (8) below. First (1-4), the model is given phrases that present in training data. The aim was to see if the model is memorized sentences from training data. In the second part (5-8), the model is given phrases that never occurred in training data.

Given words: *kabul etmek gerekir ki* (‘must admit’)
Output: kabul etmek gerekir ki bu konuda bir şey söylemek mümkün değil
(‘must admit that it is impossible to say something about it’) (1)

Given words: *her kitabın* (‘Every book’)
Output: her kitabın bir hikayesi var
(‘every book has a story’) (2)

Given words: *aklından geçen* (‘in your mind’)
Output: aklından geçen bir şey var mı
(‘is there anything in your mind’) (3)

Given words: *fotoğraf yarışması* (‘photography contest’)
Output: fotoğraf yarışması için de <num> bin lira harcandı
(‘<num> thousand liras is spent for the photography contest as well’) (4)

Given words: *müthiş soğuk bir gündü* (‘it was an extremely cold day’)
Output: Müthiş soğuk bir gündü ve çok güzel bir atmosfer vardı
(‘it was an extremely cold day and there was a nice atmosphere’) (5)

Given words: *sıkıntı oluşturacak davranışlardan* (‘from distressing behavior’)
Output: sıkıntı oluşturacak davranışlardan uzak durmak lazım
(‘It is needed to stay away from distressing behavior’) (6)

Given words: *işverenler elbette ki* (employers of course’)
Output: işverenler elbette ki bu konuda çok önemli bir adım attı
(employers, of course, have taken a very important step in this regard’) (7)

Given words: *işimi yaptıktan sonra paramı* (‘my money after I did my job’)
Output: işimi yaptıktan sonra paramı verdim
(‘I gave my money after I did my job’) (8)

As can be seen from the outputs, the model is affected by the subjects of corpora. Generated sentences look like a sentence taken from a newspaper instead of a sentence that would be used in daily life. However, generated sentences show that even the model is given the same phrases that are present in the training data, it does not complete it to the same sentences but a slightly different version. This concludes that

the model does not memorize but learn. The further investigation remains as future work.

CHAPTER 6

CONCLUSION

In this thesis, the Turkish language is modeled using LSTM, which is a trending method in Deep Learning. Although it can be possible to build language models for any linguistic units such as character-based, word-based, sentence-based, this study is focused on word-based language modeling. We have implemented both single-layer LSTM and stacked LSTM to predict the next word in a sentence in written text. Therefore, tasks are designed to calculate accuracy on a word level. We have followed the common practices and then tune the hyperparameters for our preprocessed Turkish corpus. Since our data have been gathered from the websites, there were spelling mistakes. Therefore, our preprocessing procedure includes spelling correction as well.

Newer deep learning alternatives are considered as well. A newer gated RNN method called GRU is implemented. We compared GRU and LSTM side by side, and LSTM has shown slightly better performance. Other alternatives would be bi-directional LSTM and transformers. However, they are not applicable to our research questions because both methods do not process input in order. This study was motivated to approximate cognitively plausible word predictions in a sentence. Therefore, our model has to process sentence word-by-word and in the same order as humans would read.

All of our experiments are done in Turkish. Turkish is an agglutinative language, and as an agglutinative language, it is morphologically rich. Because of this productivity, vocabulary size is huge, which increases the OOV rate. Suffixes have different forms due to Turkish grammar rules, including tenses, phonology, and subject-verb agreement. Because of these rules, the language model should both predict root and its appropriate suffix to get a score. All these constraints make the Turkish language modeling challenging.

In human vs. model task, human language processing ability is investigated. Our motivation was to see if a language model could behave like a human at predicting the next word. Therefore, we have trained seven different alternative models to perform this task. Our models' output has shown a notable correlation with human results. Our detailed analysis reveals that correlation is high for frequently used words/clitics. On the other hand, correlation is very low on phrases, which is never or rarely present in our training data. The reason is that humans have real-life knowledge to make predictions, but the model can only use what it learns from the training data.

In conclusion, we were able to train a generalized LSTM-based language model for

Turkish. In the previous chapter, our LM generate sentences after given 2-3 words. The output shows, it is capable of completing a basic sentence. The correlation analysis shows that there is a substantial alignment between human and computer predictability scores. With the future works planned, our findings might be inspiring. A model with a high correlation to human behavior might be useful in modeling human language processing/acquisition. Alternative models that test predictability under different conditions may be useful for testing some hypotheses on human language processing. Also, it could be used for selecting words/sentences with different predictability scores for psycholinguistics experiments.

6.1 Future Work

Although the results are promising, there remains some future work to improve accuracy. The end results could be improved by increasing training data quality. While our training data consists of formal sentences taken from a news website, test sentences are taken from a literary book. Therefore, their text type is different. Since we want to build a generic language model, the training corpus should be more homogeneous in terms of text types. Another improvement on the training data might be using additional special markers for dates and percentages as we use <num> marker for numbers.

Since our primary object is trying to investigate the correlation between human and artificial LM predictability results, there could be an improvement in the experiment with humans as well. Human cloze-test predictability experiments could be done with more participants and more test sentences to get generalized results. Then, correlation analysis could be calculated again with our pre-trained LM.

The correlation results could be extended with consideration of other factors. In this study, only prediction accuracy is considered. However, reading a printed text is a very complex process that involves eye-movement. Associating eye-movements with predictability scores could give further insights into human language processing.

Another future work is to explore the effect of word characteristics. In the scope of this thesis, the only position of the word is studied. However, there are some other features such as word length, frequency of a word. By further investigating word features, the results could be improved.

Our findings about the idiomatic expressions show that our model is not performing as good as humans. However, further investigation on new evaluation metrics is needed to obtain quantitative results. Also, evaluations could be extended to consider synonym words.

This thesis has counted if a full word is predicted by the model, but partial scoring can be considered to adapt features of the Turkish language. For example, if a model predicts the root word but could not predict the correct suffix, it does not get any score in this study. However, considering the huge amount of vocabulary, predicting a root is still an achievement. We have observed that our model has a poor performance on suffixes. The reason might be that humans do not predict the whole word in the same phase instead, root and suffixes are predicted in different phases. To investigate these

mechanisms, two different models could be built. Then, root and suffix predictions could be made with these two models instead of one whole-word language model.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... others (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283).

Altmann, G. T., & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, *73*(3), 247–264.

Bai, J., Nie, J.-Y., & Paradis, F. (2004). Using language models for text classification. In *Proceedings of the Asia Information Retrieval Symposium, Beijing, China.*

Bar, M. (2007). The proactive brain: using analogies and associations to generate predictions. *Trends in cognitive sciences*, *11*(7), 280–289.

Bar, M. (2009). The proactive brain: memory for predictions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1521), 1235–1243.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166.

Bergen, B. K. (2012). *Louder than words: The new science of how the mind makes meaning*. Basic Books (AZ).

Bianchi, B., Carrillo, F., Fernandez Slezak, D., Kamienkowski, J. E., & Shalom, D. E. (2014). Human and computer estimations of predictability of words on written language. In *XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO)-XV Argentine Symposium on Artificial Intelligence (ASAI)(Buenos Aires, 2014).*

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., ... Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, *16*(2), 79–85.

Bubic, A., Von Cramon, D. Y., & Schubotz, R. I. (2010). Prediction, cognition and the brain. *Frontiers in human neuroscience*, *4*, 25.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, *36*(3), 181–204.

DeLong, K. A., Urbach, T. P., & Kutas, M. (2005). Probabilistic word pre-activation during language comprehension inferred from electrical brain activity. *Nature neuroscience*, *8*(8), 1117–1121.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Federmeier, K. D. (2007). Thinking ahead: The role and roots of prediction in language comprehension. *Psychophysiology*, *44*(4), 491–505.

Fernández, G., Shalom, D. E., Kliegl, R., & Sigman, M. (2014). Eye movements during reading proverbs and regular sentences: The incoming word predictability effect. *Language, Cognition and Neuroscience*, *29*(3), 260–273.

Fu, R., Zhang, Z., & Li, L. (2016). Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st youth academic annual conference of chinese association of automation (yac)* (pp. 324–328).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (`http://www.deeplearningbook.org`)

Goodman, J. (2001). A bit of progress in language modeling. *arXiv preprint cs/0108005*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222–2232.

Hagoort, P., Hald, L., Bastiaansen, M., & Petersson, K. M. (2004). Integration of word meaning and world knowledge in language comprehension. *science*, *304*(5669), 438–441.

Hale, J. (2001). A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the north american chapter of the association for computational linguistics.*

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Huettig, F. (2015). Four central questions about prediction in language processing. *Brain research*, *1626*, 118–135.

Inhoff, A. W., & Rayner, K. (1986). Parafoveal word processing during eye fixations in reading: Effects of word frequency. *Perception & psychophysics*, *40*(6), 431–439.

Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., & Ney, H. (2016). Lstm, gru, highway and a bit of attention: An empirical overview for language modeling in speech recognition. In *Interspeech* (pp. 3519–3523).

Joos, M. (1950). Description of language design. *The Journal of the Acoustical Society of America*, *22*(6), 701–707.

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kemighan, M. D., Church, K., & Gale, W. A. (1990). A spelling correction program based on a noisy channel model. In *COLNG 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics.*

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kliegl, R., Grabner, E., Rolfs, M., & Engbert, R. (2004). Length, frequency, and predictability effects of words on eye movements in reading. *European journal of cognitive psychology*, *16*(1-2), 262–284.

Kliegl, R., Nuthmann, A., & Engbert, R. (2006). Tracking the mind during reading: The influence of past, present, and future words on fixation durations. *Journal of Experimental Psychology: General*, *135*(1), 12.

Kuperberg, G. R., & Jaeger, T. F. (2016). What do we mean by prediction in language comprehension? *Language, cognition and neuroscience*, *31*(1), 32–59.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, *106*(3), 1126–1177.

Melis, G., Dyer, C., & Blunsom, P. (2017). On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In T. Kobayashi, K. Hirose, & S. Nakamura (Eds.), *Interspeech* (p. 1045-1048). ISCA. Retrieved from `http://dblp.uni-trier.de/db/conf/interspeech/interspeech2010.html#MikolovKBCK10`

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mikolov, T., Yih, W.-t., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 746–751). Atlanta, Georgia: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/N13-1090`

Özkan, A., Beken Fikri, F., Kırkıcı, B., Kliegl, R., & Acartürk, C. (2020). Eye movement control in Turkish sentence reading. *Quarterly Journal of Experimental Psychology*, 1747021820963310.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275–281).

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, *124*(3), 372.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, *88*(8), 1270–1278.

Sak, H., Güngör, T., & Saraçlar, M. (2011). Resources for Turkish morphological processing. *Language resources and evaluation*, *45*(2), 249–261.

Shaules, J. (2018). Deep learning: Unconscious cognition, the intuitive mind, and transformative language learning. *Juntendo Journal of Global Studies*, *3*, 1–16.

Shewalkar, A., Nyavanandi, D., & Ludwig, S. A. (2019). Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, *9*(4), 235–245.

Srihari, R., & Baltus, C. (1992). Combining statistical and syntactic methods in recognizing handwritten sentences. In *AAAI Symposium: Probabilistic Approaches to Natural Language* (pp. 121–127).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association.*

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Taylor, W. L. (1953). “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, *30*(4), 415–433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Willems, R. M., Frank, S. L., Nijhof, A. D., Hagoort, P., & Van den Bosch, A. (2016). Prediction during natural language comprehension. *Cerebral Cortex*, *26*(6), 2506–2516.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, *13*(3), 55–75.

Zhang, J. (2019). Cognitive functions of the brain: Perception, attention and memory. *arXiv preprint arXiv:1907.02863*.

Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.